

Настройка хостинга под apache + PHP + MySQL

Автор: lissyara.

Оригинал: http://www.lissyara.su/articles/freebsd/www/apache1.3+_php+_mysql/

Был у конторы хостинг, для клиентов, бесплатный - как дополнение к тарифным планам с большим траффиком. Десятка полтора сайтов на FreeBSD4.11, apache1.3, php4, mysql4.0 - ничё особенного. Достался он мне по наследству, от старых админов, всё настроено, работает, что называется - не трожь. Но однажды переполнился раздел /var - почтовый спул занял всё свободное место. Недолгое разбирательство выяснило - все письма отправлены с какого-то из сайтов, и в них спам. С какого - выяснить не представлялось возможным - письма шли от пользователя www. Спул почистил, залочил в php.ini функцию mail(), сел думать - что можно сделать подручными средствами. У всех знакомых, работавших у хостеров, были самописные фишки на эту тему - они меняли идентификатор пользователя процесса apache. Апач меня не интересовал - у него своих средств для защиты хватало, а вот php, крутившийся модулем, - работал от того же пользователя, что и апач, - а это не есть гуд. Из других решений - php как CGI. Почитав, понял - подойдёт. Работать будет медленней, чем модулем, т.к. будут дополнительные затраты на порождение дополнительного процесса, но во FreeBSD это достаточно шустрый процесс (хотя и накладный, не отрицаю), поэтому особого замедления быть не должно.

Итак, начинаем с апача - нам нужен suexec - он позволяет выполнять пользовательские программы от его имени, независимо от того, под каким пользователем работает сам http-сервер. Таким образом, для апача нам надо будет лишь предоставить доступ на чтение, включив его в группу пользователя, а всем - запретить вообще доступ. Ставим апач (если стоял - сносим и ставим заново), для начала добавив в файл /etc/make.conf такие строки: # Директория где лежат порты
PORTSDIR?= /usr/ports

```
# APACHE
```

```
.if ${CURDIR} == ${PORTSDIR}/www/apache13
```

```
# Для запуска CGI-скриптов от gid и uid пользователя, а не WEB-сервера
```

```
WITH_APACHE_SUEXEC=yes
```

```
# Где будет работать suexec (ещё будет в userdir)
```

```
APACHE_SUEXEC_DOCROOT=/usr/local/hosting
```

```
# Пользовательские директории, в которых будет работать suexec
```

```
APACHE_SUEXEC_USERDIR="/usr/local/hosting/*/cgi-bin"
```

```
# Настройки производительности (без комментариев - объяснений толковых не нашёл,
```

```
# а может торопился и плохо искал...)
```

```
APACHE_SUEXEC_LOG=/usr/local/var/log/httpd-suexec.log
```

```
APACHE_HARD_SERVER_LIMIT=yes
```

```
WITH_APACHE_PERF_TUNING=yes
```

```
APACHE_HARD_SERVER_LIMIT=1024
```

```
.endif
```

После чего, собственно, пересобираем: `/usr/home/lissyara/>cd /usr/ports/www/apache13`
`/usr/ports/www/apache13/>make deinstall && make clean && make`
`? && make install && make clean`

После этого, создаём новый класс пользователей, которых будем ограничивать, у меня он называется ``webuser". Для этого, в файл `/etc/login.conf` добавляем такую секцию: # конфиг для клиентов. Сразу же - пояснения:

```
# copyright - файл содержащий информацию о копирайтах
# welcome - приветственное сообщение, выводится при входе по ssh. Я пока не
# планирую давать доступ по ssh - но на будущее возможно пригодится.
# setenv - устанавливаем переменные пользователя
# path - пути к исполняемым программам
# manpath - пути к манам
# nologin - местоположение программы, выводящей сообщение о блокировке аккаунта
# cputime - предел использования CPU
# datasize - максимальный размер данных пользователя (в памяти)
# stacksize - максимальный размер стека
# memorylocked - максимальный размер зарезервированной памяти ядра
# memoryuse - ограничение размера используемой памяти ядра
# filesize - максимальный размер файла
# coredumpsize - максимальный размер создаваемого дампа (при ошибках)
# openfiles - максимальное число открытых файлов, на процесс
# maxproc - максимальное число процессов
# sbsize - максимальный разрешённый размер буфера сокета
# priority - приоритет исполняемых процессов
# requirehome - для логина пользователя требуется реально существующая
# домашняя директория
# umask - начальная umask для создаваемых файлов
# tc - остальные переменные берём из указанного класса
```

webuser:

```
:copyright=/etc/COPYRIGHT:
:welcome=/etc/motd:
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K,EDITOR=ee:
:path=/bin /usr/bin /usr/local/bin:
:manpath=/usr/share/man /usr/local/man:
:nologin=/sbin/nologin:
:cputime=5m:
:datasize=512M:
:stacksize=4M:
:memorylocked=64M:
:memoryuse=256M:
:filesize=512M:
:coredumpsize=0:
:openfiles=1024:
```

```
:maxproc=64:  
:sbsize=unlimited:  
:priority=20:  
:requirehome:  
:umask=026:  
:tc=default:
```

Пересобираем БД классов пользователей:/usr/home/lissyara/>cap_mkdb -v /etc/login.conf
cap_mkdb: 10 capability records
/usr/home/lissyara/>

Теперь надо поставить квоты на файловую систему, где у нас хостинг (надеюсь, вы не пускаете юзеров пастись в /usr и прочих системных разделах?). Вначале, я попробовал воткнуть её слёту:/usr/home/lissyara/>echo 'enable_quotas="YES"' >> /etc/rc.conf

Перезагрузился, и попробовал поставить квоту:/usr/home/lissyara/for_hosting/>
/usr/home/lissyara/for_hosting/>edquota -u user_name2
edquota: warning: quotas are not compiled into this kernel
edquota: creating quota file /usr/local/quota.user

Непрокатило :). Для этого в ядре должна быть её поддержка, для чего вносим в конфиг такую опцию:options QUOTA

Надо заметить ещё такой момент, во FreeBSD есть ограничение, пользователь не может входить более чем в 16 групп, это очень старое ограничение, связанное каким-то боком с NFS. В данном случае, в процессе экспериментов, я в него очень быстро упёрся - согласно создаваемой системе, у каждого пользователя будет своя персональная группа, в которую кроме пользователя входит пользователь www - от него работает apache. Если его не включить в неё, то он не сможет читать файлы в директориях пользователя. При других раскладках - типа, все пользователи в одной группе - косяк с безопасностью - есть возможность, что прочтут чужие файлы. Поэтому пришлось найти где это ограничение забито и подправить. Это файл /usr/src/sys/syslimits.h, в нем редактируем такую строку:#define NGROUPS_MAX 64 /* max supplemental group id's */

Из минусов - надо помнить про эту опцию, при обновлениях, ну и при передаче дел неплохо бы упомянуть. Вроде как с 7.0 должны сделать это ограничение "мягким" - можно будет его менять через sysctl. На данный момент, во всех версиях, его менять нельзя - переменная есть, но не меняется:/usr/home/lissyara/>sysctl -a | grep ngroup
kern.ngroups: 16

```
/usr/home/lissyara/>sysctl kern.ngroups=64
sysctl: oid 'kern.ngroups' is read only
/usr/home/lissyara/>
```

Короче - ставим сколько надо (всё не тестил, но вроде как всё рабтает нормально, правда, для верности, пересобирал вообще всё - мир, ядро, и все установленные приложения. Каюсь - метод, всёже косячный, но умней ничё в башке не родилось, да и время поджимало). Пересобираем ядро. Это описано у меня, и в хандбуке. Пока ядро пересобирается, готовимся, надо подправить /etc/fstab. Правим строку, относящуюся к монтированию раздела где живут сайты пользователей. У меня, в итоге, оно выглядит так:

| # устройство | Куда моунтим | ФС | опции | дамп |
|--------------|--------------------|-----|---------------------|------|
| /dev/ad2s1e | /usr/local/hosting | ufs | rw,userquota,nosuid | 2 2 |

Также, добавляем такую строку в /etc/pw.conf, чтобы при создании пользователя, генерились рандомные пароли: defaultpasswd=random

По окончании сборки ядра перезагружаемся, и снова пробуем запустить edquota - если пройдёт без ошибок - значит всё нормально. (не забываем добавить вышеописанную строчку про 'enable_quotas="YES"' в /etc/rc.conf!). Далее, я настрогал небольшой скриптик, для администрирования всего этого добра: #!/bin/sh -

```
# Скрипт для создания клиентской шары, и заведения нужного пользователя
# в системе. Написан lissyara 2006-10-02 - 2006-10-09 www.lisssyara.su
```

```
# Ограничения заводимого пользователя - сколько может занимать места,
# и сколько инод ему выделять (фактически, это - максимальное число файлов
# и директоий которые он сможет завести. Но не точно.
```

```
# Реально - может быть меньше). 0 - анлим.
```

```
user_datasize="500000" # В килобайтах в 4.11 и в Mb в 6.1, почему так -
```

```
# я не понял. будьте внимательны.
```

```
user_inode="100000" # Иноды не резиновые. Оцените число командой
```

```
# `df -i` и решайте какие у вас должны быть лимиты
```

```
# Файловая система, на которую вносим ограничения
```

```
fs="/usr/local/hosting"
```

```
# Данные для MySQL
```

```
mysql_user="root"
```

```
mysql_passwd=""
```

```
# Переменные:
```

```
domain_name="$1"
```

```
user_name="$2"
```

```
# программы
```

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
cat="/bin/cat" # Перенаправление потоков ввода-вывода
awk="/usr/bin/awk" # Язык обработки шаблонов
echo="/bin/echo" # Вывод строк на экран
grep="/usr/bin/grep" # Выбор строк по шаблону
pw="/usr/sbin/pw" # Управление учётками, группами пользователей
rm="/bin/rm" # Удаление файлов и директорий
mkdir="/bin/mkdir" # Создание директорий
chown="/usr/sbin/chown" # Выставление владельца файлов и директорий
php="/usr/local/bin/php-cgi" # Бинарник PHP для использования в виде CGI
cp="/bin/cp" # Копирование файлов и директорий
mysql="/usr/local/bin/mysql" # MySQL - клиент
wc="/usr/bin/wc" # подсчёт числа строк
cut="/usr/bin/cut" # Обрезка строк
md5="/sbin/md5" # подсчёт md5 файлов, строк
jot="/usr/bin/jot" # генерация случайного числа
tail="/usr/bin/tail" # Показывает последнюю часть файла
edquota="/usr/sbin/edquota" # редактор квот
ls="/bin/ls" # Кбснбуу диреткорий, файлов
chflags="/usr/bin/chflags" # Утановка флагов на файл
tr="/usr/bin/tr" # Замена, удаление символов в строке
mysqldump="/usr/local/bin/mysqldump" # Содание дампов MySQL
gzip="/usr/bin/gzip" # Архиватор
rm="/bin/rm" # Удаление файлов и директорий
mv="/bin/mv" # Перемещение файлов и директорий
basename="/usr/bin/basename" # Вывод базового имени файла (из полного пути)
uname="/usr/bin/uname" # Информация о локальной машине
```

Файл с настройками php

```
php_ini="/usr/local/etc/php.ini-for-hosting"
```

Временный файл - для технических нужд

```
tmp_file="/tmp/$.file.tmp"
```

Файл с системными альясами (для почты - отлупы на письма с сайтов будут

возвращаться на этот сервер, надо их пересылать на пользователя, у меня

забито как info@домен_пользователя)

```
aliases_file="/etc/aliases" # На самом деле, во фпре, это симлинк на
```

```
# файл /etc/mail/aliases.
```

Директория хостинга

```
hosting_prefix="/usr/local/hosting/sites"
```

Директория для бэкапов удалённых сайтов

```
backup_dir="/usr/local/hosting/backup"
```

Строим префикс для работы с БД

```
query_prefix="${mysql} --user=${mysql_user} --password=${mysql_passwd}
```

```
--database=mysql"
```

Проверяем наличие приложений - в разных версиях FreeBSD разные пути

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
for application in ${cat} ${awk} ${echo} ${grep} ${pw} ${rm} ${mkdir}
${chown} ${php} ${cp} ${mysql} ${wc} ${cut} ${md5} ${jot} ${tail}
${edquota} ${ls} ${chflags} ${tr} ${mysqldump} ${gzip} ${rm} ${mv}
${basename} ${uname} ${php_ini}
do
if [ ! -s ${application} ]; then
echo "Приложение ``${application}`` не найдено! Проверьте путь!"
exit
fi
done
```

```
# Поехали :)))
case "$3" in
create)
```

```
# Проверяем, не начинается ли имя пользователя с цифры (у `pw`, в этом случае
# будут проблемы с созданием группы пользователя. Либо руками, либо эта
# проверка. Я решил что лучше с цифры не начинать.)
```

```
${echo} "1111111111" >> ${tmp_file}
${echo} ${user_name} | ${grep} -v "^[0-9]" > ${tmp_file}
# проверяем размер файла - не равен ли нулю
if [ -s ${tmp_file} ]
then
# Ничё не делаем
${rm} ${tmp_file}
else
# Пытаются пользователя с первой цифрой всандалить.
${echo} "Имя пользователя не может начинаться с цифры!!"
${rm} ${tmp_file}
exit
fi
```

```
# Проверяем, не содержит ли имя пользователя точки:
${echo} ${user_name} | ${tr} -d "[:alnum:]" | ${grep} "." > ${tmp_file}
if [ -s ${tmp_file} ]
then
# Пытаются пользователя с точками в имени всандалить.
${echo} "Имя пользователя не может содержать точки!!"
${rm} ${tmp_file}
exit
else
# Ничё не делаем
${rm} ${tmp_file}
fi
```

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
# Проверяем уникальность имени пользователя
${cat} /etc/passwd | ${awk} -F ":" '{print $1}' | ${grep} "^[a-zA-Z0-9_]" |
{
while read existing_user_name
do
if [ ${existing_user_name} = ${user_name} ]
then
# Есть такой юзер. Ругаемся, выходим
${echo} "Пользователь с именем "${user_name}" существует!"
# пишем временный файл, чтоб передать инфу из цикла
${echo} 1 > ${tmp_file}
exit
fi
done
}
# Проверяем наличие и содержимое временного файла.
if [ -s ${tmp_file} ]
then
if [ `cat ${tmp_file}` -eq 1 ]
then
${rm} ${tmp_file} # удаляем временный файл
exit # выходим
fi
fi

# проверяем, нету ли такого виртуалхоста в apache
if [ -s ${hosting_prefix}/../httpd_configs/${domain_name}.conf ]
then
# Уже есть такой виртуалхост
${echo} "Хост с именем "${domain_name}" уже есть в конфиге!"
exit
fi

# такого пользователя нет - заводим в системе
user_passwd="" ${pw} useradd ${user_name} -c "UserDomain =
${domain_name}" -L webuser -s /bin/csh -d ${hosting_prefix}/${user_name}`"
# Добавляем апаче в группу пользователя
${pw} groupmod ${user_name} -m www

# Создаём структуру директорий для пользователя
${mkdir} -p ${hosting_prefix}/${user_name}/log
${mkdir} -p ${hosting_prefix}/${user_name}/www
${mkdir} -p ${hosting_prefix}/${user_name}/cgi-bin
${mkdir} -p ${hosting_prefix}/${user_name}/tmp

# копируем файлы в CGI
```

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
{cp} {php} {hosting_prefix}/{user_name}/cgi-bin/
{cp} {php_ini} {hosting_prefix}/{user_name}/cgi-bin/

# Даём права на эти директории
{chown} -R {user_name}:{user_name} {hosting_prefix}/{user_name}

# Ставим флаги на php.ini
{chflags} sunlnk {hosting_prefix}/{user_name}/cgi-bin/php.ini
{chflags} schg {hosting_prefix}/{user_name}/cgi-bin/php.ini

# Добавляем пользователя в список тех, кто chroot`ится при
# заходе по FTP
{echo} {user_name} >> /etc/ftpchroot

# Стругаем квоты для пользователя
{edquota} -e
{fs}:{user_datasize}:{user_datasize}:{user_inode}:{user_inode}
{user_name}

# Стругаем пароль для MySQL
mysql_user_passwd="{jot} -r 1 0 9000000 | {md5} -p | {tail} -1
| {cut} -c 1-8"

# Заводим пользователя в MySQL
{query_prefix} --execute="CREATE DATABASE IF NOT EXISTS `{user_name}_db`"
{query_prefix} --execute="GRANT ALL PRIVILEGES on `{user_name}_db`.`* to
'{user_name}'@'localhost' IDENTIFIED BY '{mysql_user_passwd}'"

# Создаём виртуалхост в апаче
{echo} "# Этот файл создан автоматически, скриптом `basename $0`

# Virtual host added by `$_USER`, `date +%Y-%m-%d` in `date +%H:%M:%S`
# from host = `$_REMOTEHOST`, host IP = ``echo $SSH_CLIENT | awk '{print $1}'``

<VirtualHost *:80>
ScriptAlias /cgi-bin/ {hosting_prefix}/{user_name}/cgi-bin/
ServerAdmin support@{domain_name}
User {user_name}
Group {user_name}
DocumentRoot {hosting_prefix}/{user_name}/www
ServerName {domain_name}
ServerAlias www.{domain_name}
# Директория пользователя
<Directory "{hosting_prefix}/{user_name}/www">
```


Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order deny,allow
Allow from all
</Directory>
DirectoryIndex index.php index.html index.htm index.shtml index.php3
AddType application/x-httpd-php .php .php3
Action application/x-httpd-php /cgi-bin/php-cgi
ErrorLog ${hosting_prefix}/${user_name}/log/error.log
CustomLog ${hosting_prefix}/${user_name}/log/access.log common
</VirtualHost>
```

```
" >> ${hosting_prefix}/../httpd_configs/${domain_name}.conf
```

```
# Создаём почтовый альяс для юзера (иначе его почта будет захламлять
# раздел /var). У меня стоит exim в простейшей (дефолтовой) конфигурации,
# т.к. мыльница для клиентов не на хостинге, а на отдельном сервере,
# и он работает с системными пользователями. Если будете делать по другому,
# перепишите этот кусок под себя.
```

```
$echo "
```

```
# Alias for user = ${user_name}, domain = ${domain_name},
# added by `$_USER`, `date +%Y-%m-%d` in `date +%H:%M:%S`
# from host = `$_REMOTEHOST`, host IP = ``echo $SSH_CLIENT | awk '{print $1}'``
${user_name}: info@${domain_name}
```

```
" >> ${aliases_file}
```

```
${echo} "Имя домена: ${domain_name}"
${echo} "Имя пользователя: ${user_name}"
${echo} "Пароль пользователя(FTP, SSH): ${user_passwd}"
${echo} "Имя базы данных: ${user_name}_db"
${echo} "Пароль пользователя (MySQL): ${mysql_user_passwd}"
${echo} "Квота пользователя: ${user_datasize}"
${echo} -n "Альяс почты (${aliases_file}) ${user_name}@`${uname} -n`"
${echo} " --> info@${domain_name}"
;;
delete)
${echo} "Вы действительно хотите удалить пользователя и домен?!"
${echo} "Наберите YES для подтверждения, или Ctrl+C для выхода..."
read confirmation
if [ $confirmation = YES ] > /dev/null 2>&1
then
# Подтверждено.
# Стругаем директорию
${mkdir} -p ${backup_dir}/${domain_name}
```

```
# снимаем флаги с директории юзера
${chflags} -R nosunlnk ${hosting_prefix}/${user_name}/*
${chflags} -R noschg ${hosting_prefix}/${user_name}/*
# Копируем содержимое
${echo} "Копируются пользовательские данные..."
${cp} -R ${hosting_prefix}/${user_name}
${backup_dir}/${domain_name}
# Дампим БД пользователя
${echo} "Делается дамп БД пользователя...."
${mysqldump} --user=${mysql_user} --password=${mysql_passwd}
--database ${user_name}_db | ${gzip} >
${backup_dir}/${domain_name}/${user_name}_db.sql.gz
# Сносим всё нахрен:
# Перемещаем конфиг апача для этого домена
${mv} ${hosting_prefix}/../httpd_configs/${domain_name}.conf
${backup_dir}/${domain_name}/
# Удаляем доки:
${echo} "Удаляются пользовательские данные..."
${rm} -Rf ${hosting_prefix}/${user_name}
# Удаляем БД
${echo} "Удаляется БД пользователя...."
${query_prefix} --execute="DROP DATABASE IF EXISTS
`${user_name}_db`"
# Удаляется пользователь из MySQL
${echo} "Удаляется пользователь из MySQL..."
${query_prefix} --execute="REVOKE ALL PRIVILEGES
ON *.* FROM `${user_name}`@'localhost'"
${query_prefix} --execute="REVOKE GRANT OPTION
ON *.* FROM `${user_name}`@'localhost'"
${query_prefix} --execute="DELETE FROM mysql.user WHERE
User='${user_name}' and Host='localhost'"
# Удаляется пользователь из системы
${echo} "Удаляется пользователь из системы..."
${pw} userdel ${user_name}
${echo} "Удаляется группа пользователя из системы..."
${pw} groupdel ${user_name}
else
# не захотел удалять :)
${echo} "Пользователь и его сайт были оставлены"
fi
;;
update)
# Обновление версии php у всех пользователей.
${ls} ${hosting_prefix}/|
{
# идёт листинг сайтов, но т.к. имя пользователя и директории
```

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
# одинаковые - значит это ещё и имена пользователей :)
while read site_user_name
do
${echo} "Идёт обновление версии php для пользователя ${site_user_name}"
# Копируем новый
${cp} ${php} ${hosting_prefix}/${site_user_name}/cgi-bin/
# Даём права на него
${chown} ${site_user_name}:${site_user_name}
${hosting_prefix}/${site_user_name}/cgi-bin/${basename} ${php}`
done
}
${echo} "Обновление завершено!"
;;
*)
```

```
${echo} ""
${echo} "Этот скрипт создаёт пользователя в системе, генерит ему
пароль, создаёт базу данных MySQL, генерит для неё пароль,
выставляет квоты на использование ресурсов системы - на данный
момент все новые пользователи равноправны - смотрите в login.conf
класс 'webuser'. Текущие ограничения:
Места: 500mb (правиться в этом файле)
Инод: 10000 (правиться в этом файле)
Процессов: 64 (правиться в login.conf)
Время работы процесса: 5 минут (правиться в login.conf)
Размер стека: 4mb (правиться в login.conf)
Размер блокируемой оперативки: 64mb (правиться в login.conf)
Макимальный объём оперативки: 256mb (правиться в login.conf)
Максимальный размер файла: 100mb (правиться в login.conf)
Открытых файлов на процесс: 1024 (правиться в login.conf)
Приоритет приложений: 20 (правиться в login.conf)
```

Внимание! На файл php.ini ставятся флаги `schg` и `sunlnk`,
не сняв их - файл не уалить и не изменить! Цель - не позволить юзеру
менять свои настройки php.

Удаление флагов:

```
chflags noschg php.ini
```

```
chflags nosunlnk php.ini
```

Установка:

```
chflags schg php.ini
```

```
chflags sunlnk php.ini
```

(Может понадобится на случай если надо всё-таки чё-то поменять
в дефолтовом файле. Но не забывайте вернуть флаги!)

Также есть предложение придерживаться определённого именования
юзеров для их сайтов - типа если домен называется `some.com` то

юзер должен быть `somesom` - так понятней. Либо, как вариант, завести таблицу, типа: 1- su, 2 - ru, 3 -som и т.п. и добавлять эти цифры в конце. Кому как удобней будет. На данном сервере, под FreeBSD 4.11, ограничение на длину имени пользователя - 16 символов. Если длинней - придётся обрубать с конца, или ещё как - чтобы не совпадало с существующими.

Про удаление - перед удалением всё бэкаптся, кроме учётки пользователя.

Про обновления - имя пользователя и домен - любые. Нужно при обновлении версии php на сервере. Обновляются все бинарники и перевыставляются права.

P.S. Не забывайте перезапускать апач после создания-удаления!"

```
{echo} ""  
{echo} "Использование: `${basename} $0`  
{domainname username create|delete|update}" >&2  
{echo} ""  
exit 64  
;;  
esac
```

Можете воспользоваться моим, а можете делать всё руками - решать вам. Тока к скрипту надо одно пояснение - он удалит любого системного пользователя, если по ошибке вы воткнёте его имя в качестве параметра с командой на удаление. Поэтому, если не понимаете, как оно работает - не надо пользоваться. Там есть защита от дурака, но от дурака умного. У меня он пашет, а как будет у вас - не знаю.

Небольшое замечание по поводу пути к файлу php.ini. В том файле, что копируется, я подправил лишь несколько пунктов, вот их настройки:safe_mode = On

```
safe_mode_gid = On  
disable_functions = system  
# поправка - этот пункт лучше так не ставить - php некорректно  
# это обрабатывает... может даже не php но у пары клиентов  
# были проблемы с аплоадом файлов на сервер, пока  
# не поставил абсолютный путь...  
#upload_tmp_dir = ../tmp  
session.save_path = ../tmp  
session.auto_start = 1  
upload_max_filesize = 4M  
memory_limit = 64M
```

А вообще, возьмите файл /usr/local/etc/php.ini-recommended, и правьте его под свои нужды.

После проделанного, не забываем поставить парва на скрипт - в нём хранится рутовый пароль от MySQL (в принципе, можно заюзать флаг "-p", чтобы запрашивал его каждый

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
раз, или отдельную переменную, а потом подставлять её значение - смотрите сами.)/usr/local/hosting/scripts/>chmod 700 create_client.sh
/usr/local/hosting/scripts/>chown root:wheel create_client.sh
/usr/local/hosting/scripts/>ls -alh create_client.sh
-rwx----- 1 root wheel 12K 24 окт 10:47 create_client.sh
```

```
Ну и пробуем скриптик:/usr/local/hosting/scripts/>./create_client.sh testing.su testing create
Имя домена: testing.su
Имя пользователя: testing
Пароль пользователя(FTP, SSH): QBnOXa2k8DCnGY3
Имя базы данных: testing_db
Пароль пользователя (MySQL): 64cb2586
Квота пользователя: 500000
Альяс почты (/etc/aliases) testing@hosting.kontora.su --> info@testing.su
/usr/local/hosting/scripts/>
/usr/local/hosting/scripts/>
/usr/local/hosting/scripts/>
/usr/local/hosting/scripts/>
/usr/local/hosting/scripts/>./create_client.sh testing.su testing delete
Вы действительно хотите удалить пользователя и домен?!
Наберите YES для подтверждения, или Ctrl+C для выхода...
YES
Копируются пользовательские данные...
Делается дамп БД пользователя....
Удаляются пользовательские данные...
Удаляется БД пользователя....
Удаляется пользователь из MySQL...
Удаляется пользователь из системы...
Удаляется группа пользователя из системы...
/usr/local/hosting/scripts/>
```

Вот и всё. Соответственно, где надо подрихтуйте напильником под свои нужды (в 4.11 и 6.0 пару путей различаются), добавьте такие строки в конец конфига апача:NameVirtualHost *:80

```
# Это первый виртуалхост - он должен быть этого сервера, т.к.
# иначе при обращении на несуществующие сайты будет выводиться
# самый первый виртуалхост - это может оказаться чей-то из клиентов,
# поэтому создаём скриптом, и копируем конфиг сюда.
<VirtualHost *:80>
ScriptAlias /cgi-bin/ /usr/local/hosting/sites/hosting/cgi-bin/
ServerAdmin support@hosting.kontora.su
User hosting
Group hosting
```

Настройка хостинга под apache + PHP + MySQL

Автор: Administrator

07.01.2010 21:23 - Обновлено 28.05.2010 13:38

```
DocumentRoot /usr/local/hosting/sites/hosting/www
ServerName hosting.kontora.su
ServerAlias www.hosting.kontora.su
# Директория пользователя
<Directory /usr/local/hosting/sites/hosting/www>
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order deny,allow
Allow from all
</Directory>
DirectoryIndex index.php index.html index.htm index.php3
AddType application/x-httpd-php .php .php3
Action application/x-httpd-php /cgi-bin/php-cgi
ErrorLog /usr/local/hosting/sites/hosting/log/error.log
CustomLog /usr/local/hosting/sites/hosting/log/access.log common
</VirtualHost>
```

```
Include /usr/local/hosting/httpd_configs/*.conf
```

P.S.1 Настройки системы и php брались из головы, так, как я считал нужным. Может что-то и неправильно, или можно сделать лучше. Смотрите и думайте сами.

P.S.2 Насчёт ограничения в 16 групп и правки исходников - это правда плохая идея.

Хотя проблем вроде нету. И если знаете, как можно сделать грамотней - поделитесь...

P.S.3 Настройки MySQL не рассмотрены сознательно,