// // // //

umask великая вещь =) или расшариваем содержимое ftp через smb

понадобилось расшарить содержимое фтп по локальной сети.
и вот незадача — прав вечно недостаточно было nobody, от которого по умолчанию на шару попадали юзеры через smb.

в общем надоело мне обновлять права на новые файлы, так что я взялся за конфиг vsftpd =)
нужные параметры в этом случае: **anon_umask & local_umask**

именно они отвечают за маску свежесозданных файлов и папок, загруженных через фтп.
прочесть про umask можно в википедии =)

мега фраза оттуда:

Права доступа файлов вычисляются при помощи следующих побитовых операций: *побит овое И между унарным дополнением аргумента (используя побитовое НЕ) и режимом полного доступа.Режим полного доступа для файлов — 666, для директорий — 777*

Настройка vsftpd с исползованием вирт.пользователей

Итак, можно продолжить, в первой части я рассказал о том что есть, что нужно и как установить vsftpd, как будто бы это может вызвать сложность =)

Пришла пора рассказать о конфигурационном файле и его настройке.
я привожу текст своего конфига, напомню что мне нужны были вирт.пользователи, чрут для каждого, кроме одного из них и разные права доступа.

```
/etc/vsftpd> cat vsftpd.conf | grep --invert-match ^#
#фтп работает в режиме стандалон, слушает дефолтный порт
listen=YES
listen_port=21
#отключаем доступ анонимусу



anonymous_enable=NO
#разрешаем локальным пользователям логинится
local_enable=YES
#разрешаем операции записи
write_enable=YES
#разрешаем видеть не только ворлд ридабл
anon_world_readable_only=NO
#запрещаем остальные операции записи, такие как изменение имени, удаление и тд.
anon_other_write_enable=NO
#маска для загруженных файлов



local_umask=022
#разрешаем аплоуды
anon_upload_enable=YES
#разрешаем создавание каталогов
anon_mkdir_write_enable=YES
#отключаем сообщение при входе в директории
dirmessage_enable=NO
#ведем лог
xferlog_enable=YES
dual_log_enable=NO
#записываем команды и ответы в лог
log_ftp_protocol=YES



xferlog_file=/var/log/vsftpd.log
```

```
 #юзер без баша и привелегий. в генту по умолчанию создается и используется ftp
 nopriv_user=ftp
 #запрещаем ascii аплоуды и доунлоды. для большей секурности. читай коментарий в
man vsftpd.conf
 ascii_upload_enable=YES
 ascii_download_enable=YES
 #показывает в списке процессов чем процесс vsftpd занят. включил на время дебагинга
 setproctitle_enable=YES
 use_localtime=YES



#запрещаем ls -R
 ls_recurse_enable=NO



#ВАЖНО! для работы вирт.юзеров это необходимо!
 guest_enable=YES
 #пользователь, которым представится вирт.пользователь для системы
 guest_username=ftp



#настройка работы в пассивном режиме. не забудьте пропустить их на файерволле
 pasv_min_port=2222
 pasv_max_port=2299
 pasv_enable=YES



#я отрубил ссл. зря кнчно наверное. но не хотел заморачиваться с ним
 ssl_enable=NO



#ВАЖНО!! для работы индивидуальной настройки пользователь необходимо указать
директорию
 #где будут хранится файлы с именем = именю пользователя, где будут описаны
правила
 #для конкретно этого пользователя
 user_config_dir=/etc/vsftpd/userconf



#памы. для аутентификации пользователей
```

```
 pam_service_name=vsftpd



#задаем в байтах в секунду ширину исп. канала
 local_max_rate=61440



#количество клиентов и количество неверных логинов в сессию
 max_clients=12
 max_login_fails=1



#ВАЖНО! автоматически создаем  и помещаем пользователя в его личную папку



user_sub_token=$USER
 #ВАЖНО! чрутим пользователя в его папке



local_root=$USER
```

 вот вроде и все что касается конфига. теперь создаем папку для индивидуальных настроек.

 /etc/vsftpd> mkdir userconf

 и создаем там файлы с именем = имени пользователя с индивидуальными настройками.

 поскольку моя задача была сделать 25 пользователей, 24 из которых идентичны (за исключением имени, пароля и папки для чрута) и имеют право аплодить файлы, читать их, создавать директории и все =) и один пользователь для просмотра всего и вся, чрутить его не нужно, права - только чтение. поскольку мне было лень создавать 24 идентичных файлов я решил занести разрешение на аплоуд и создание директорий в конфиг vsftpd.conf, а для пользователя с правами просмотра создать отдельный файл в директории /etc/vsftpd/userconf со следующим содержанием:

```
 user_sub_token=
 write_enable=NO
 anon_world_readable_only=NO
 anon_mkdir_write_enable=NO
```

anon_other_write_enable=NO
anon_upload_enable=NO
local_root=

тоесть я для одного пользователя отказался от определенных настроек в конфиге, таким образом он может гулять по всем папкам, но не может что либо писать =)

далее, в /etc/pam.d нужно положить файл vsftpd со след. содержимым:

/etc/vsftpd> cat /etc/pam.d/vsftpd
auth required /lib/security/pam_userdb.so db=/etc/vsftpd/vsftpd_login crypt=hash
account required /lib/security/pam_userdb.so db=/etc/vsftpd/vsftpd_login crypt=hash

не забудьте указать crypt=hash!! в официальной документации об этом не слова, но без указани механизма криптования ничего не выйдет

далее, заносим в файл logins.txt логины и пароли построчно и создаем файл в формате беркли дб с хешами:

db_load -T -t hash -f logins.txt /etc/vsftpd/vsftpd_login.db
chmod 600 /etc/vsftpd/vsftpd_login.db

вот теперь все готово =)
поскольку вирт.пользователи мапяться на пользователя ftp то в качестве рута фтп сервера используется хоум директория пользователя ftp: /home/ftp
соответвеннно в ней и создаются папки для остальных пользователей для чрута.

если что - еще добавлю. но вроде все =)

vsftpd и авторизация

половину сегодняшнего дня я потратил на выяснение почему **vsftpd** не пускает виртуального юзера.

пары логинпароль в оф.документации предлагают хранить в беркли дб.
цитата :
Step 1) Create the virtual users database.
We are going to use pam_userdb to authenticate the virtual users. This needs
a username / password file in "db" format - a common database format.
To create a "db" format file, first create a plain text files with the
usernames and password on alternating lines.
See example file "logins.txt" - this specifies "tom" with password "foo" and
"fred" with password "bar".
Whilst logged in as root, create the actual database file like this:

db_load -T -t hash -f logins.txt /etc/vsftpd_login.db
(Requires the Berkeley db program installed).
NOTE: Many systems have multiple versions of "db" installed, so you may
need to use e.g. db3_load for correct operation. This is known to affect
some Debian systems. The core issue is that pam_userdb expects its login
database to be a specific db version (often db3, whereas db4 may be installed
on your system).

This will create /etc/vsftpd_login.db. Obviously, you may want to make sure
the permissions are restricted:

chmod 600 /etc/vsftpd_login.db

настроив все как было сказано, за исключением того, что версия беркли дб у меня 4.3, я пол дня пытался заставить работать авторизацию, получая разные сообщения, начиная с **login failure** заканчивая **500 OOPS: priv_sock_get_result**.
получив последнее я наконец то выяснил в чем косяк. оказывается в паме нужно указать какой алгоритм применять =)
забавно млин. то решение, что я нашел:
необходимо добавить **crypt=hash** для каждой строки в паме. отсюда
вариант решение так же, найденный на гентушном форуме: emerge -u world broke VSFTPd (Urgent!)

vsftpd на замену serv-u ftp

Больше года назад по производственной необходимости был установлен **Serv-U FTP** server для одной простой цели - хранение фотографий экспертиз. Для каждого филиала была выделена отдельная тачка, для каждого филиала - один пользователь. Пользователи имеют право только читать и писать. Есть так же один юзер с правом смотреть все папки.
Соответсвенно задачи перед фтп-сервером следующие:
* чрутить юзеров в их папки
* работа с виртуальными юзерами
* нормально понимать кирилицу в именах папок/файлов
это из необходимого. отказался от использования мускула в качестве хранилища пользователей - с учетом низкой активности, вернее никакой активности добавления пользователей или смены паролей не вижу смысла что то ковырять в этом направлении.
Мигрирую с серв-у по одной простой причине: лицензию на него приобретать не собираемся, да и в любом случае хотел реализовать фтп на линухе.

Как всегда вопрос, что поднять на линухе с подобным функционалом? Из вариантов: **vs ftpd**
и
proftpd
. первый мне внушил больше доверия. так что начал его установку.

Юзаем мега eix для поиска пакета и изучения юз флагов:

```
gate ~ # eix net-ftp/vsftpd
* net-ftp/vsftpd
Available versions:  2.0.4-r1 2.0.5 2.0.5-r1 2.0.5-r2 2.0.5-r3 {caps logrotate pam selinux ssl tcpd xinetd}
Homepage:            http://vsftpd.beasts.org/
Description:         Very Secure FTP Daemon written with speed, size and security in mind
```

читаем для каждого флага *euse -i <флаг>* как учил сансэй =) выясняем что нам понадобится. делаем соответсвующую запись в /etc/portage/package.use
смотрим что потянется вместе с vsftpd:

```
gate ~ # emerge -p net-ftp/vsftpd
```

These are the packages that would be merged, in order:

Calculating dependencies... done!
 [ebuild N   ] app-admin/logrotate-3.7.2 USE="(-selinux)"
 [ebuild N   ] net-ftp/ftpbase-0.01 USE="pam"
 [ebuild N   ] net-ftp/vsftpd-2.0.5-r3 USE="logrotate pam tcpd -caps (-selinux) -ssl -xinetd"

 как видно, у меня до сих пор не стоит в системе logrotate. думаю пора бы его поставить. его настройка - в след. заметке.

 Инстолим этот комплект.

 Идем далее. и начинаем курить руководство =) видимо докуривать буду завтра =) когда будет готов точно рабочий конфиг =)

оригинал: http://butch.blog.ru/tag/vsftpd

запись файлов через FTP в правам все всем 777

write_enable=YES
local_umask=0000

мой конфиг vsftpd

{jcomments on}