Оглавление

Предисловие

Доброго времени суток, уважаемый читатель. Я конечно понимаю, что я далеко не известный и уж тем более не писатель. Я самый обычный блоггер, и еще более давний системный администратор. Для чего же мне тогда надо это все? Для чего мне сидеть ночью и набивать этот текст, компоновать свои же блоггерские посты в эту PDF «книгу»? Все довольно просто. В блоге я уже писал для чего я это делаю и первой же заметкой поставлю как раз ее.

Если у кого-либо возникнет мысль написать комментарий к одной из «статей», то я всегда готов их обсудить в комментариях к блогу или же связавшись со мной одним из следующих способов:

- естественно это джаббер — silver@itoutsourcing.org.ua (прошу учесть, что это не е-мейл, писать сюда бесполезно);
- второй способ — это классический е-мейл: silver@silverghost.org.ua.

Честно говоря, это мой первый опыт публикации в таком виде, так что уж не обессудьте. Лучше подскажите как улучшить, а я за это на Вас сошлюсь в следующей версии выпуска.

Если же Вы хотите написать гостевой пост в моем блоге, то милости прошу, если он будет соответствовать около компьютерной тематике.

Благодарности:

Первый и пока единственный, кто откликнулся на мою PDF-версию — Владимир Бредников[1], за что ему огромное спасибо. Он посоветовал, как более правильно оформить все эти записки в что-то удобоваримое.

С уважением, Silver Ghost.
http://silverghost.org.ua

1 http://bappoy.pp.ru/toc/

Зачем и почему я веду этот блог

На написание данной заметки меня навела статья на сайте BlogTips. В общем, вот мои 5 причин ведения блога и создания PDF версии своих записок:

1. Мозг уже не может вмещать в себя весь поток информации.

В эру нынешних технологий поток информации, который проходит через наш мозг, весьма и весьма не маленький. Запомнить все параметры, настройки, “галочки” и “птички” становится просто невозможно. Потому я в свой блог записываю все то, с чем сталкиваюсь по работе, дабы потом не искать в джунглях интернета то, что уже один раз было настроено и работало.

2. Помощь новичкам.

Все мы когда-то начинали с нуля. И каждый, кто чему-либо учился сам, скажет на сколько тяжело разбираться в чем-то с нуля. Именно поэтому я описываю то, с чем разбираюсь. Может многое из записей и банально, но новичкам будет проще разобраться, т.к. стараюсь писать все таки “для себя”.

3. Личный авторитет.

Нет, не в том плане, что я расту в своих глаза и заболеваю “звездочкой”. Авторитет в глазах моих друзей, как профессионала, человека, который разбирается и хочет чего-либо добиться. Опять же доверие ко мне, как к человеку и специалисту.

4. Профессиональное общение.

Мой блог привнес в мою личную и деловую жизнь многих интересных людей, с которыми приятно общаться и всегда приятно иметь дело.

5. Естественно деньги.

Как многим известно, мой блог хостится не на профессиональном хостинге, а у меня дома, на домашнем сервере. Т.е. фактически он находится в сети бесплатно и денег не требует. Но тем не менее блог приносит мне деньги за счет рекламы. И чтоб денег было больше, надо больше писать. Их ведь никогда много не бывает. :)

6. Раскрутка блога (PDF дополнение)

Естественно за счет PDF-версии я хочу привлечь посетителей к своему блогу.

Пользователю Линукс

Хороший скрипт для wine

После установки wine шрифты как минимум корявые в виндовых приложениях. Лечится так:

1. Дергаем скрипт winetricks[2].
2. Устанавливаем chmod 755.
3. Запускаем и устанавливаем corefonts.

Все. Теперь у нас нормальные шрифты. :)

URL заметки: http://silverghost.org.ua/2008/02/09/xoroshij-skript-dlya-wine/

DirectX в Wine - скажи Cedega “пока”

Ну вот наконец-то игроманы могут порадоваться. Под wine теперь можно инсталлировать DirectX 9.0c.

Скажу сразу, что сам не пробовал, т.к. не играюсь, но статью переведу для тех, кто не хочет сам этого делать. Перевод не дословный и не литературный, всего лишь список необходимых действий. Переведу лишь ключевые моменты.

Версия DirectX - 9.0c, Wine: 0.9.58, эмуляция Windows 2000.

Запускаем winecfg и выставляем аудио драйвер для своей системы.

Далее нужно выставить режим “родной (Windows)” для файлов mscoree.dll и streamci.dll и скопировать их из Windows в system32.

Теперь нужно установить кучу dll в “родной” режим для корректной работы установки:

```
[Software\\Wine\\DllOverrides] 1206264929

“d3d8″ =”builtin”
“d3d9″ =”builtin”
“d3dim”=”native”
“d3drm”=”native”
“d3dx8″ =”native”
“d3dx9_24″ =”native”
“d3dx9_25″ =”native”
“d3dx9_26″ =”native”
“d3dx9_27″ =”native”
“d3dx9_28″ =”native”
“d3dx9_29″ =”native”
“d3dx9_30″ =”native”
“d3dx9_31″ =”native”
“d3dx9_32″ =”native”
“d3dx9_33″ =”native”
“d3dx9_34″ =”native”
“d3dx9_35″ =”native”
“d3dx9_36″ =”native”
“d3dxof”=”native”
“dciman32″ =”native”
“ddrawex”=”native”
```

2 http://kegel.com/wine/winetricks

```
“devenum”=”native”
“dinput”=”builtin”
“dinput8″ =”builtin”
“dmband”=”native”
“dmcompos”=”native”
“dmime”=”native”
“dmloader”=”native”
“dmscript”=”native”
“dmstyle”=”native”
“dmsynth”=”native”
“dmusic”=”native”
“dmusic32″ =”native”
“dnsapi”=”native”
“dplay”=”native”
“dplayx”=”native”
“dpnaddr”=”native”
“dpnet”=”native”
“dpnhpast”=”native”
“dpnlobby”=”native”
“dsound”=”builtin”
“dswave”=”native”
“dxdiagn”=”native”
“mscoree”=”native”
“msdmo”=”native”
“qcap”=”native”
“quartz”=”native”
“streamci”=”native”
```

Это можно сделать проще. Во вкладке “Библиотеки” установить первую библиотеку “d3d8″ =”Встроенный”. Потом в каталоге ./wine найти user.reg и скопировать остаток туда в раздел [Software\\Wine\\DllOverrides].

Дальше скачиваем DirectX 9.0c March 2008 release[3].

Запускаем и распаковываем инсталлятор.

Идем в папку, где лежат распакованные файлы и запускаем “wine ./dxsetup.exe”

Скачиваем в ~/.wine/drive_c/windows/system32/drivers драйвер[4] для тестирования звука Direct Music.

Запускаем dxdiag и проверяем.

По идее все.

URL заметки: http://silverghost.org.ua/2008/03/23/directx-v-wine-skazhi-cedega-poka/

Ubuntu 7.10 vs PhotoShop CS2

Сегодня поставил под wine 0.9.46 Adobe PhotoShop CS2. При установке особых проблем не было, кроме того, что активация ругнулась на нехватку места и отвалилась. Установка завершилась удачно, а дальше начались танцы с бубном вокруг активации.

Проблема с ошибкой о нехватке места осталась и после установки. Долгие поиски в интернете привели к методу вычитывания из реестра wine ветки и ее конвертации с последующим возвратом назад. Этот метод окончился провалом и очередной ошибкой уже в:

3 http://filehippo.com/download_directx/
4 http://www.kirupa.templarian.com/gm.dls

```
$ recode ucs-2..ascii ./adobe.reg

recode: ./adobe.reg failed: Untranslatable input in step
`ISO-10646-UCS-2..ANSI_X3.4-1968'
```

Побороть это дело каким-либо способом не получилось, а тащить этот файл на виндовую машину совсем не хотелось. Пришлось рыть англоязычные форумы и в результате в обсуждении этой ошибки на сайте wine нашел более простой метод. Вот он мне и помог. Итак:

1. Создаем симлинк: **ln -s /dev/sda ~/.wine/dosdevices/c::**
2. Далее временно меняем права на /dev/sda: **chmod 666 /dev/sda**
3. Запускаем PhotoShop, активируем его и меняем назад права на 660 на раздел диска.

Собственно говоря вот и весь метод решения проблемы. У меня сейчас стоит CS2 и работает идеально.

URL записки: http://silverghost.org.ua/2008/01/28/ubuntu-vs-photoshop-cs2/

Ubuntu 8.04 и Adobe PhotoShop

Обновился до Ubuntu 8.04 Hardy Heron и тут же возникла проблема с Фотошопом. После загрузки стал ругаться на то, что не может найти какие-то нужные ему библиотеки.

Путем запуска его из консоли выяснил, что wine не может использовать первый мегабайт адресного пространства ДОС.

```
preloader: Warning: failed to reserve range 00000000-60000000
err:dosmem:setup_dos_mem Cannot use first megabyte for DOS
address space, please report
```

Лечится это просто:

```
$ sudo sysctl -w vm.mmap_min_addr=0
$ sudo gedit /etc/sysctl.conf
```

Ищем *vm.mmap_min_addr = 65536* и меняем 65536 на 0.

Все. :) Теперь наслаждаемся работой в ФШ.

URL записки: http://silverghost.org.ua/2008/04/30/ubuntu-804-i-adobe-photoshop/

Переключение раскладки в Ubuntu 8.04

Многие, кто обновил Ubuntu до Hardy Heron, заметили проблему с переключением раскладки после перезагрузки системы. Решается проблема на самом деле просто. Идем в xorg.conf и правим секцию клавиатуры до такого состояния:

```
Section "InputDevice"
     Identifier "Generic Keyboard"
     Driver "kbd"
     Option "CoreKeyboard"
     Option "XkbRules" "xorg"
     Option "XkbModel" "pc105"
     Option "XkbLayout" "us,ru"
     Option "XkbVariant" ",winkeys"
```

```
    Option "XkbOptions" "grp:ctrl_shift_toggle"
EndSection
```

Рестартуем Иксы и пробуем. Если не помогло, то это значит, что Вы что-то перемудрили в конфигах Гнома, который начал управлять клавиатурой. Просто удалите каталог ~/.gconf/desktop/gnome/peripherals/keyboard и сделайте logout/login.

URL записки: http://silverghost.org.ua/2008/05/01/pereklyuchenie-raskladki-v-ubuntu-804/

Перезагрузка повисшей системы

Нашел вот совет[5] по перезагрузке зависшей Линукс системы.

Бывает такое, что система зависает. Или иксы не выходят из свопа, или некий процесс отъел всю память и обработчик клавиатуры не может получить управление, или придётся признать, что у броузера медленно подтекает крышак, ну в общем, ничего не сделать, кроме ребута.

А вот как его сделать, чтобы не повредить причёску файловую систему. Прибегнем к так называемой магической ядерной кнопке, а именно сочетанию **Alt-SysRq** (он же PrintScreen) с разными буквами. Понадобятся нам такие, в этом порядке, с учётом, что у нас клавиатура qwerty:
Alt-Sysrq-**R** переключить клавиатуру в режим XLATE (перехватить управление у иксов)
Alt-Sysrq-**E** послать всем процессам, кроме инита, решительный привет, то есть SIGTERM
Alt-Sysrq-**I** послать всем процессам, кроме инита, окончательный привет, то есть SIGKILL
Alt-Sysrq-**S** sync. Для последователей старой школы: нажать дважды :).
Alt-Sysrq-**U** перемонтировать все файловые системы в read-only
Alt-Sysrq-**B** начать загрузку

Говорят, что эту последовательность (Alt-SysRq-**REISUB**) можно запомнить как слово **BUSIER** наоборот. От себя можно добавить, что если хочется освежить в голове эту краткую мнемонику, то можно на консоли нажать Alt-SysRq-H, и будет выдана краткая справка. Для тех, кто не уверен, где у него консоль: надо нажать Ctrl-Alt-F1.

URL записки: http://silverghost.org.ua/2007/11/20/perezagruzka-povisshej-sistemy/

Linux в качестве клиента для Abills

Позавчера мне сказали, что один из клиентов вызвал "специалиста" по Линуксу и они строили ВПН к нашему биллингу. В качестве биллинга у нас используется Abills. Ничего не получилось и подозревают, что проблема у нас. Честно говоря, я тоже начал сомневаться, т.к. это не проверял. Сегодня поковырял pptp у себя на ноутбуке. Все работает как часы. Конфиги прилагаю:

```
# cat /etc/ppp/peers/Spektr
name USER
remotename USER
debug
lock
deflate 0
defaultroute
file /etc/ppp/options.pptp
pty "/usr/sbin/pptp SERVER —nolaunchpppd"

# cat /etc/ppp/options
lock
```

5 http://linux.xlibs.net/2007/11/19/howto-reboot-a-frozen-system-with-the-magic-sysrq-keys/

```
+chap

# cat /etc/ppp/options.pptp
lock
+chap
require-mschap-v2
nobsdcomp
nodeflate
noaccomp
nopcomp
defaultroute
noipdefault
mtu 1492
mru 1492
ipcp-accept-local
ipcp-accept-remote
noauth

# cat /etc/ppp/chap-secrets
USER * PASSWORD *
```

Запуск:

```
$ sudo pppd call Spektr
```

или

```
# pppd call Spektr
```

Вот с такими конфигами у меня все соединилось и работает.

URL записки: http://silverghost.org.ua/2007/08/06/linux-v-kachestve-klienta-dlya-abills/

Ubuntu, WiFi & фича FireFox

Поставил себе на ноутбук Acer TravelMate 2492 систему Ubuntu Linux 7.10 Gutsy.

Немного помучался с драйверами под WiFi, т.к. системные драйвера в упор не видели сеть. Чип от Broadcom вообще загадочная вещь. Под Федорой у меня с большим трудом получилось его завести и то глючило что-то. В принципе проблема решилась установкой ndiswrapper + драйвера от винды.

Все достаточно просто:

1. Устанавливаем ndiswrapper:

```
$ sudo apt-get install ndiswrapper-common ndiswrapper-utils-1.9
```

2. Распаковываем куда нибудь драйвера и пишем из под пользователя root такие команды:

```
$ sudo ndiswrapper -i bcmwl5.inf
$ sudo modprobe ndiswrapper
$ sudo su
# echo «blacklist bcm43xx» >> /etc/modprobe.d/blacklist
# echo «ndiswrapper» >> /etc/modules
```

3. Перегружаемся и проверяем. Все должно работать.

Если же Вы обновили Ubuntu 7.10 до 8.04 и ndiswrapper отказался работать, то загляните в /var/log/syslog. Скорее всего там есть ссылка на firmware и инструкцию по установке для Вашего чипа. Во всяком случае у меня было именно так.

Еще одно неудобство вызвал FireFox, который по Backspace не хотел переходить на страницу назад по истории, а перелистывал на один экран вверх текущую страницу. Тоже не все так сложно. Открываем страницу «about:config» и в ней находим параметр browser.backspace_action и выставляем его в 0.

Теперь в FireFox все как положено.

Системному администратору

Apache

5 пунктов обезопасить Апач

1. Отключить сигнатуры:

```
ServerSignature Off
ServerTokens Prod
```

2. Запуск апача под собственным аккаунтом:

```
User apache
Group apache
```

3. Запретить доступ Апачу в другие директории:

```
Order Deny,Allow
Deny from all
Options None
AllowOverride None
```

4. Запретить просмотр листинга каталога:

```
Options -ExecCGI -FollowSymLinks -Indexes
```

5. Выключить лишние модули:

```
cat ./httpd.conf | grep LoadModule
```

URL записки: http://silverghost.org.ua/2007/03/04/5-punktov-obezopasit-apach/

.htaccess

1. Закрыть директорию паролем:

```
require valid-user
Authname “DirectoryName”
Authtype Basic
AuthUserFile “/path/to/.htpasswd”
```

2. Закрыть файл паролем:

```
require valid-user
Authname “DirectoryName”
Authtype Basic
AuthUserFile “/path/to/.htpasswd”
```

Создание пачки поддоменов через .htaccess

Иногда надо создавать автоматически поддомен в какой-либо зоне. Например, для хостинга домашних страничек пользователей. Вот как это делается:

1. Нам надо создать запись для всех поддоменов в домене.

В зону мы вносим новый домен “*”, ссылающийся на необходимый IP адрес сервера.

2. Настраиваем Apache.

Создаем вирутальный хост с примерно такими параметрами:

```
<VirtualHost *:80>
DocumentRoot /var/www/example.com
ServerName example.com
…

ServerAlias *.example.com
RewriteEngine On
RewriteCond %{HTTP_HOST} ^((.*)\.)example.com$
RewriteRule ^/(.*) /%2/$1

</VirtualHost>
```

Теперь, чтоб у нас работал сайт, создаем каталог /var/www/example.com/www и туда заливаем содержимое сайта www.example.com. По аналогии делаем для остальных поддоменов.

Apache + mod_ssl

Разобрался я как прикрутить к Апачу SSL и авторизовывать юзеров по клиентским сертификатам. Весьма удобно, должен заметить. В общем и целом схема проста:

1. Создать собственный доверенный сертификат (Certificate Authority), для того чтобы с помощью него подписывать и проверять клиентские сертификаты.
2. Создать клиентские сертификаты, подписанные доверенным сертификатом, для последующей передачи их клиентам.
3. Сконфигурировать веб-сервер для запроса и проверки клиентских сертификатов.

Начнем с первого пункта:

Собственный доверенный сертификат (Certificate Authority — далее CA) необходим для подписи клиентских сертификатов и для их проверки при авторизации клиента веб-сервером. С помощью приведенной ниже команды создается закрытый ключ и самоподписанный сертификат.

```
openssl req -new -newkey rsa:1024 -nodes -keyout ca.key -x509
-days 500 -subj /C=RU/ST=Msk/L=Msk/O=My\
Inc/OU=Sale/CN=bla/emailAddress=usr@dom.ru -out ca.crt
```

Описание аргументов:

req

Запрос на создание нового сертификата.

-new

Создание запроса на сертификат (Certificate Signing Request — далее CSR).

-newkey rsa:1024

Автоматически будет создан новый закрытый RSA ключ длиной 1024 бита. Длину ключа можете настроить по своему усмотрению.

-nodes

Не шифровать закрытый ключ (См. примечание выше).

-keyout ca.key

Закрытый ключ сохранить в файл ca.key.

-x509

Вместо создания CSR (см. опцию -new) создать самоподписанный сертификат.

-days 500

Срок действия сертификата 500 дней. Размер периода действия можете настроить по своему усмотрению. Не рекомендуется вводить маленькие значения, так как этим сертификатом вы будете подписывать клиентские сертификаты.

-subj /C=RU/ST=Msk/L=Msk/O=My\ Inc/OU=Sale/CN=bla/emailAddress=usr@dom.ru

Данные сертификата, пары параметр=значение, перечисляются через ’/’. Символы в значении параметра могут быть «подсечены» с помощью обратного слэша “\”, например «O=My\ Inc». Также можно взять значение аргумента в кавычки, например, -subj «/xx/xx/xx».

Описание параметров:

C — Двухсимвольный код страны (Country). Необязательный параметр.

ST — Название региона/области/края/республики/… (State Name). Необязательный параметр.

L — Название города/поселка/… (Locality Name). Необязательный параметр.

O — Название организации (Organization Name). Необязательный параметр.

OU — Название отдела (Organization Unit). Необязательный параметр.

CN — Имя сертификата, при создании серверных сертификатов используется доменное имя сайта, для клиентских сертификатов может быть использовано что угодно (Common Name). Обязательный параметр. Максимальная длина 64 символа.

emailAddress — почтовый адрес (E-mail address). Необязательный параметр. Максимальная длина 40 символов.

Необязательные параметры могут быть пропущены, например, /C=RU/CN=blabla/emailAddress=user@domain.ru.

-out ca.crt

Сертификат сохранить в файл ca.crt.

В результате выполнения команды появятся два файла ca.key и ca.crt.

Далее нам необходимо сгенерировать клиентские сертификаты, что тоже не сложно.

Создайте конфигурационный файл с именем ca.config следующего содержания.

```
[ ca ]
default_ca = CA_CLIENT          # При подписи сертификатов
                                # использовать секцию CA_CLIENT
```

```
dir = ./db                          # Каталог для служебных файлов
certs = $dir/certs                  # Каталог для сертификатов
new_certs_dir = $dir/newcerts       # Каталог для новых сертификатов
database = $dir/index.txt           # Файл с базой данных
                                    # подписанных сертификатов
serial = $dir/serial                # Файл содержащий серийный номер
                                    # сертификата
                                    # (в шестнадцатеричном формате)
certificate = ./ca.crt              # Файл сертификата CA
private_key = ./ca.key              # Файл закрытого ключа CA
default_days = 365                  # Срок действия подписываемого
                                    # сертификата
default_crl_days = 7                # Срок действия CRL
default_md = md5                    # Алгоритм подписи
policy = policy_anything            # Название секции с описанием
                                    # политики в отношении данных
                                    # сертификата
[policy_anything]
countryName = optional              # Код страны — не обязателен
stateOrProvinceName = optional      # ……
localityName = optional             # ……
organizationName = optional         # ……
organizationalUnitName = optional   # ……
commonName = supplied               # …… — обязателен
emailAddress = optional             # ……
```

Создайте структуру каталогов и файлов, соответствующую описанной в конфигурационном файле

```
# mkdir db
# mkdir db/certs
# mkdir db/newcerts
# touch db/index.txt
# echo «01» > db/serial
```

Для создания подписанного клиентского сертификата предварительно необходимо создать запрос на сертификат, для его последующей подписи. Аргументы команды полностью аналогичны аргументам использовавшимся при создании самоподписанного доверенного сертификата (см. $1), но отсутствует параметр -x509.

```
# openssl req -new -newkey rsa:1024 -nodes -keyout client01.key -
subj
/C=RU/ST=Msk/L=Msk/O=Inc/OU=Web/CN=usr/emailAddress=usr@dm.ru
-out client01.csr
```

В результате выполнения команды появятся два файла client01.key и client01.csr.

При подписи запроса используются параметры заданные в файле ca.config (см. $2.1.)

```
# openssl ca -config ca.config -in client01.csr -out client01.crt
-batch
```

Описание аргументов:

ca

Подпись запроса с помощью CA.

-config

ca.config

Использовать конфигурационный файл ca.config.

-in

client01.csr

CSR находится в файле client01.csr

-out

client01.crt

Сохранить сертификат в файл client01.crt

-batch

Не спрашивать подтверждения подписи.

В результате выполнения команды появится файл клиентского сертификата client01.crt.

Для передачи полученных в результате предыдущих операций файлов клиенту, обычно используется файл в формате PKCS#12. В этот файл упаковывается и защищается паролем вся информация необходимая клиенту для инсталляции сертификата в броузер.

```
# openssl pkcs12 -export -in client01.crt -inkey client01.key
-certfile ca.crt -out client01.p12 -passout pass:q1w2e3
```

Описание аргументов:

pkcs12

Работа с файлами формата PKCS#12.

-export

Экспортирование данных в файл.

-in client01.crt

Файл клиентского сертификата.

-inkey client01.key

Файл закрытого ключа.

-certfile ca.crt

Файл доверенного сертификата.

-out client01.p12

Сохранить данные в файл client01.p12.

-passout pass:q1w2e3

Установить пароль q1w2e3 на файл (пароль может быть любым, в том числе и пустым)

На этом процесс создания клиентского сертификата завершен. Теперь вам необходимо передать клиенту файл client01.p12 и пароль к нему любым удобным безопасным способом, а также проинструктировать его о процедуре инсталяции сертификата в броузер.

Для реализации процесса **авторизации по клиентским сертификатам** необходимо сконфигурировать веб-сервер для решения следующих задач:

1. Запрет доступа к защищаемой области по протоколу HTTP.
2. Запрос и проверка клиентских сертификатов.

Найдите в конфигурационном файле веб-сервера httpd.conf секцию <VirtualHost>, соответствующую вашему сайту и добавьте в неё следующие директивы

```
<Directory /path/to/secure/area/>
SSLRequire
</Directory>
```

Описание директив:

/path/to/secure/area/

Абсолютный путь до директории защищаемой области.

SSLRequire

Запрещает доступ клиенту, если при соединении не используется протокол HTTPS (HTTP через SSL).

Найдите в конфигурационном файле веб-сервера httpd.conf секцию , соответствующую вашему сайту и добавьте в неё следующие директивы:

```
SSLCACertificateFile /path/to/ca.crt
<Directory /path/to/secure/area/>
SSLVerifyClient require
</Directory>
```

Описание директив:

SSLCACertificateFile /path/to/ca.crt

Абсолютный путь до доверенного сертификата. Также в качестве значения директивы SSLCACertificateFile может быть указан файл, содержащий несколько доверенных сертификатов (формируется путем обычной конкатенации файлов сертификатов), тогда все они будут считаться доверенными сертификатами.

SSLVerifyClient require

При наличии этой директивы веб-сервер будет запрашивать сертификат у клиента в обязательном порядке. Если клиент не предоставляет сертификат, тогда сервер отклоняет запрос. Если клиент предоставляет сертификат, то веб-сервер проверяет его срок действия и поставщика сертификата (сертификат которым он подписан), если сертификат поставщика присутствует в файле SSLCACertificateFile, то проверка считается успешной и клиенту предоставляется доступ до защищенной области.

Для того, чтобы изменения конфигурационного файла веб-сервера вступили в силу необходимо перезапустить веб-сервер

```
# apachectl restart
```

Это все. Проверено на Apache/2.2.6 под Fedora 7.

URL записки: http://silverghost.org.ua/2007/12/14/apache-mod_ssl/

Ускоряем отдачу сайтов в Apache

Отдаем ява-скрипты и таблицы стилей в сжатом виде:

```
<FilesMatch «\.(js|css)$»>
SetOutputFilter DEFLATE
</FilesMatch>
```

Кешируем на стороне клиента файлы мультимедиа:

```
<FilesMatch «\.(ico|pdf|flv|jpg|jpeg|png|gif|js|css|swf)$»>
Header set Cache-Control «public»
Header set Expires «Thu, 15 Apr 2010 20:00:00 GMT»
</FilesMatch>
```

Отключаем механизм ETag, который передает значение хеша файла и определяет изменился ли файл:

```
Header unset ETag
FileETag None
```

Отключаем заголовок Last-Modified:

```
<FilesMatch «\.(ico|pdf|flv|jpg|jpeg|png|gif|js|css)$»>
Header unset Last-Modified
</FilesMatch>
```

В почту пришел вопрос об ошибке:

```
Invalid command 'Header', perhaps misspelled or defined by a
module not included in the server configuration failed!
```

Для работы инструкции «Header» необходимо загружать модуль mod_headers:

```
LoadModule headers_module modules/mod_headers.so
```

Сервер после данных манипуляций будет гораздо быстрее отдавать контент.

URL записки: http://silverghost.org.ua/2008/01/06/uskoryaem-otdachu-sajtov-v-apache/

Дополнительная защита веб-сервера

По наводке от моего товарища Lice пришлось ковырять как закрыть от PHP-скриптов доступ в другие каталоги. дело в том, что если не установлен PHP как CGI модуль, то варианта два. Либо SafeMode + doc_root, что совсем не приятно и мне не нравится. Причины расписывать тут не буду, т.к. не хочется устраивать холивар. В общем, я пошел по другому пути. В каждый виртуальный хост Апача я добавил такую конструкцию:

```
php_admin_value open_basedir /var/www/user/
```

Таким образом, я избавился от неприятной вещи по типу fileread('/etc/passwd');.

Дальше больше. Lice по FTP залил какой-то ПХП-шный shell и стали тестировать. Как я и ожидал, каталоги то он увидел, а вот скачать что-либо, кроме как из своего каталога - никак.

Вот теперь надо отрубить system и exec с кучей всего еще. В php.ini мы правим:

```
disable_functions =
"apache_get_modules,apache_get_version,apache_getenv,apache_note,
apache_setenv,disk_free_space,diskfreespace,dl,highlight_file,ini
_alter,ini_restore,openlog,passthru,phpinfo,proc_nice,shell_exec,
show_source,symlink,system,exec"
```

Отдельное огромное спасибо Lice за помощь.

SSH

Ограничение доступа по SHH через PAM

Одним из преимуществ PAM (Pluggable Authentication Module) является возможность ограничения числа сетевых пользователей, имеющих доступ к определенному сервису, на основе списка. Например, с помощью PAM можно задать ограничения на SSH-подключения.

Добавьте в файл /etc/pam.d/sshd строку:

```
auth required /lib/security/pam_listfile.so onerr=fail item=user
sense=allow file=/etc/sshd_users
```

Эта директива разрешает регистрацию пользователя через sshd, если его имя присутствует в файле /etc/sshd_users. Опции имеют следующие значения:

- onerr=fail — этот параметр не даст успешно пройти тест, если произойдет ошибка (указанный файл не найден или в файле обнаружена некорректная строка). В результате пользователю будет отказано в регистрации через sshd. Другим возможным значением параметра onerr является succeed.
- item=user — означает, что мы проверяем имя пользователя.
- sense=allow — если пользователь найден в заданном файле, пройти этот тест. Это разрешит регистрацию пользователя, если пройдут также все другие тесты. Другим возможным значением параметра sense является deny.
- file=/etc/sshd_users — задает файл, содержащий список имен пользователей (по одному имени на строку), которым разрешена регистрация через sshd.

С указанной строкой, файл /etc/pam.d/sshd будет выглядеть приблизительно так:

```
#%PAM-1.0
auth required pam_stack.so service=system-auth
auth required pam_nologin.so
auth required pam_listfile.so onerr=fail item=user sense=allow
file=/etc/sshd_users
account required pam_stack.so service=system-auth
password required pam_stack.so service=system-auth
session required pam_stack.so service=system-auth
session required pam_limits.so
session optional pam_console.so
```

Теперь вы можете добавить в файл /etc/sshd_users необходимых пользователей. Каждое имя пользователя должно быть указано отдельной строкой.

URL записки: http://silverghost.org.ua/2007/03/04/ogranichenie-dostupa-po-shh-cherez-pam/

Ограничение доступа по SHH средствами sshd

Иногда может быть небезопасно разрешать удаленных доступ для всех пользователей. Существует много способов ограничить удаленный доступ к системе. Можно, например, использовать PAM, IPwrappers или IPtables. Однако, проще всего ограничить доступ через SSH — это соответствующе настроить демон SSH.

Добавьте в конец файла /etc/ssh/sshd_config строку, аналогичную нижеприведенной:

```
AllowUsers [имяпользователя]
```

Где [имяпользователя] — это имя пользователя, которому вы хотите разрешить доступ. Например:

```
AllowUsers joe
```

Эта директива определяет пользователей, которым разрешён доступ в систему через SSH. В примере, пользователь joe уже существует в системе. Несколько пользователей могут быть заданы через пробел.

URL записки: http://silverghost.org.ua/2007/03/04/ogranichenie-dostupa-po-shh-sredstvami-sshd/

SSH + PortKnocking

Portknocking — технология, которая позволяет открыть любой порт (в данном случае 22), постучав (запросом telnet, например) в любой другой порт. Используя ниже приведенный скрипт, можно это реализовать.

```
$IP='/sbin/iptables'
$EXTIP=<Your external IP>
$IPT -A INPUT -d $EXTIP -p tcp –dport 1500 -j LOG
$IPT -A INPUT -d $EXTIP -m state –state NEW -m tcp -p tcp –dport
22 -m recent –rcheck –name SSH -j ACCEPT
$IPT -A INPUT -d $EXTIP -m state –state NEW -m tcp -p tcp –dport
1499 -m recent –name SSH –remove -j DROP
$IPT -A INPUT -d $EXTIP -m state –state NEW -m tcp -p tcp –dport
1500 -m recent –name SSH –set -j DROP
$IPT -A INPUT -d $EXTIP -m state –state NEW -m tcp -p tcp –dport
1501 -m recent –name SSH –remove -j DROP
$IPT -A INPUT -d $EXTIP -p tcp –dport 22 -j DROP
```

Т.е. выполнив команду:

```
$ telnet <Your external IP> 1500
```

Вы откроете 22 порт. Используя же порты 1499 или 1501 — закроете 22 порт.

URL записки: http://silverghost.org.ua/2007/06/13/ssh-portknocking/

Еще раз про SSH и iptables

На просторах линукс-форума нашел:

```
iptables -A INPUT -p tcp -m state — -state NEW - -dport 22 -m
recent — -update - -seconds 20 -j DROP
iptables -A INPUT -p tcp -m state — -state NEW - -dport 22 -m
recent — -set -j ACCEPT
```

Все кто ломятся быстрее 20 сек — отлетают, т.е. определяются как боты-брутфорсеры. Просто и эффективно.

URL записки: http://silverghost.org.ua/2007/09/27/eshhe-raz-pro-ssh-i-iptables/

Networking (сети)

Source routing или использование двух каналов

Маршрутизация через несколько каналов.

Добавим две таблицы в /etc/iproute2/rt_tables:

```
10 RS
20 UTEL
```

Организация ответов сервера через тот же канал, откуда пришел пакет:

```
# First ISP
IF1='ppp0´
IP1=`/sbin/ifconfig $IF1 | grep “inet addr:” | cut -d ” ” -f 12 | cut
-d “:” -f 2`
P1=`/sbin/ifconfig $IF1 | grep “inet addr:” | cut -d ” ” -f 14 | cut -
d “:” -f 2`
P1_NET=$IP1´ /32´
# Second ISP
IF2='ppp1´
IP2=`/sbin/ifconfig $IF2 | grep “inet addr:” | cut -d ” ” -f 12 | cut
-d “:” -f 2`
P2=`/sbin/ifconfig $IF2 | grep “inet addr:” | cut -d ” ” -f 14 | cut -
d “:” -f 2`
P2_NET=$IP2´ /32´
# Add routes to gateways and default routes
/sbin/ip route add $P1_NET dev $IF1 src $IP1 table RS
/sbin/ip route add default via $P1 table RS
/sbin/ip route add $P2_NET dev $IF2 src $IP2 table UTEL
/sbin/ip route add default via $P2 table UTEL
# Add source routing
/sbin/ip route add $P1_NET dev $IF1 src $IP1
/sbin/ip route add $P2_NET dev $IF2 src $IP2
# Add default route
/sbin/ip route add default via $P1
# Add source routing using rt_tables
/sbin/ip rule add from $IP1 table RS
/sbin/ip rule add from $IP2 table UTEL
```

URL записки: http://silverghost.org.ua/2007/03/22/source-routing-ili-ispolzovanie-dvuh-kanalov/

Конфигурация vlan’ов и static-routes

Сегодня поставил перед собой цель разнести конфигурацию vlan’ов по конфигам в sysconfig/network-scripts, что более удачно и правильно, чем их подъем из скриптов. Заодно и с файлом конфигурации статических маршрутов разобрался. Как показал

поиск по гуглу, найти путнее описание как это делается задача не тривиальная. Во всяком случае у меня не сразу это вышло, но все же получилось собрать инфу и вот что вышло.

В общем все банально:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0.3
DEVICE=eth0.3
BROADCAST=10.10.10.255
HWADDR=00:17:08:58:9E:A1
IPADDR=10.10.10.1
NETMASK=255.255.255.0
NETWORK=10.10.10.0
ONBOOT=yes
VLAN=yes
```

“.3″ - это и есть номер вилана, т.е. 3-й вилан. Параметр “VLAN=yes” - обязательный, иначе система не поймет, что это вилан и ничего не подхватится. Алиас на вилан вешается подобным образом:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0.3:10
DEVICE=eth0.3:10
BROADCAST=10.10.20.255
HWADDR=00:17:08:58:9E:A1
IPADDR=10.10.20.1
NETMASK=255.255.255.0
NETWORK=10.10.20.0
ONBOOT=yes
VLAN=yes
```

Как видим, ничего сложного. “:10″ - номер алиаса в 3-м вилане.

Со статик-роутом все еще проще:

```
# cat /etc/sysconfig/static-routes
eth0.4 net xxx.xxx.xxx.xxx/nn gw yyy.yyy.yyy.yyy
```

Запись соответствует команде “route add -net xxx.xxx.xxx.xxx/nn gw xxx.xxx.xxx.xxx”.

Все. Теперь не надо рисовать никаких своих скриптов, вешать их в rc.local. Все будет сделано на этапе поднятия сетевых интерфейсов.

Тестировалось на Fedora 8.

URL записки: http://silverghost.org.ua/2007/11/21/konfiguraciya-vlanov-i-static-routes/

Использование в Linux IEEE 802.1Q VLAN’ов совместно с Cisco Catalyst Switch

Linux:

```
/sbin/vconfig add eth1 5
/sbin/vconfig add eth1 4
/sbin/ifconfig eth1.4 192.0.0.8 netmask 255.255.255.240 up
/sbin/ifconfig eth1.5 192.0.1.8 netmask 255.255.255.240 up
```

Catalyst:

```
conf t
int fa0/0
switchport mode trunk
switchport trunk allowed vlan 4,5,1002-1005
```

URL записки: http://silverghost.org.ua/2007/03/04/ispolzovanie-v-linux-ieee-8021q-vlanov-sovmestno-s-cisco-catalyst-switch/

Как "протянуть" 802.1q tagged порт через ip-сеть

Нашел на просторах opennet'a[6].

Дано:

Есть hostA, который воткнут в каталист, в trunk (т.е. тегированный) порт, инкапсуляция 802.1q.

Есть hostB, который маршрут до hostA. маршрут живой, ip-пакеты между хостами безпроблемно бегают.

Задача: надо с hostA "притащить" виланы на hostB.

Решение: vtund + bridge.

описание клиента в vtund.conf

```
homepeer {
passwd qwerty;
type ether;
device home;
proto tcp;
compress yes;
stat yes;
persist yes;
up {
ifconfig "%% up";
program "brctl addbr br0″ ;
program "brctl addif br0 %%";
program "brctl addif br0 eth0″ ;
ifconfig "br0 up";
};
down {
ifconfig "%% down";
ifconfig "br0 down";
program "brctl delbr br0″ ;
};
}
```

описание пира из конфига сервера

```
homepeer {
passwd qwerty; # Password
type ether; # Ethernet tunnel
device work; # Device tap1
proto tcp;
compress yes;
up {
ifconfig "%% up";
};
```

6 http://opennet.ru

```
down {
ifconfig “%% down”;
};
}
```

Теперь на той Linux машине, куда кидаем порт:

```
vconfig set_name_type VLAN_PLUS_VID_NO_PAD
vconfig add work 4
ifconfig vlan4 10.1.1.1 netmask 255.255.255.0 up
vconfig add vlan4 8
ifconfig vlan8 192.168.1.1 netmask 255.255.255.0 up
```

Замечание:

Клиент - хост, с которого мы тащим порт. Он живёт в серой сети и имеет выход в internet через NAT. Сервер - машина с публичным ip.

URL записки: http://silverghost.org.ua/2007/03/08/kak-protyanut-8021q-tagged-port-cherez-ip-set/

VPN (PPTP) over NAT

Сегодня вот столкнулся с ситуацией, когда пришлось поднимать VPN через сервер с NAT. Долго я мучался и рыл доки, пока не нашел модули ядра к iptables, чем все и закончилось. Все просто. Дописываем в автозагрузку сервера с NAT (файл rc.local) загрузку этих модулей:

```
modprobe ip_gre
modprobe ip_nat_pptp
modprobe ip_conntrack_pptp
```

Ну и естественно запускаем эти же команды в консоли от рута, чтоб не перезагружать сервер. После этих манипуляций VPN бегает на “Ура”.

URL записки: http://silverghost.org.ua/2008/05/22/vpn-pptp-over-nat/

Asterisk

Алло, девушка… Смольный!

Еще совсем недавно кому-то приходилось крутить ручку телефонного аппарата и кричать в трубку, чтоб его услышали. В век новых технологий у нас в кармане почти у каждого по паре “мобилок”, голос передается без проблем через интернет, а на предприятиях стоят свои “хардовые” или “софтовые” АТС.

На самом деле я не буду делать экскурс в историю АТС, так как не знаю о них почти ничего, а вот о том, как поднять Asterisk на Ubuntu 7.10 server, я расскажу.

Начну с процесса непосредственно установки:

```
$ sudo apt-get install asterisk
```

Далее нам надо завести хотя бы одного пользователя для теста. Это тоже не сложно. В /etc/asterisk/users.conf мы добавляем такую секцию:

```
[200]
type=friend
host=dynamic
```

```
username=200
secret={password}
nat=no
canreinvite=no
context=home
callerid="User1"  <200>
allow=gsm
allow=ulaw
allow=alaw
```

Естественно, что {password} нужно заменить паролем, а 200 - это номер телефона. Подключаем программу-звонилку (софт-телефон) к Asterisk с логином 200 и паролем {password} и вуаля. Теперь уже можно сделать первый звонок, хотя бы на номер 1000, где Вас поздравят с установкой Asterisk, правда пока на английском языке. Но не можем же мы разговаривать только с роботами? Добавляем по аналогии с [200] номер [201], который мы выдадим другу. Но пока еще звонить друг другу мы не можем, так как у нас правил набора номеров, т.е. кто и куда может звонить.

Устраняем этот недостаток путем прописывания секции [home] в /etc/asterisk/extentions.conf:

```
[home]
exten => 200,1,Dial(SIP/200)
exten => 201,1,Dial(SIP/201)
```

Перезапускаем Asterisk и уже можно звонить друг другу.

В следующий раз я расскажу как можно Asterisk соединить с сервисом sipnet.ru и совершать звонки в Москву и Питер бесплатно.

URL записки: http://silverghost.org.ua/2008/02/20/alo-devushka-smolnyj/

Asterisk + sipnet.ru

Пришла пора рассказать, как я привязывал к Astrisk'у сервис от sipnet.ru.

Первым делом, нам надо в users.conf добавить секцию:

```
[sipnet]
secret = {sipnet_password}
provider =
trunkstyle = customvoip
username = {sipnet_number}
trunkname = sipnet
callerid =
hasexten = no
hassip = yes
hasiax = no
registeriax =
registersip = yes
host = sipnet.ru
dialformat = ${EXTEN:1}
context = home
group =
insecure = invite
fromuser = {sipnet_number}
fromdomain = sipnet.ru
contact = 200
disallow=all
```

```
allow = alaw
allow = ulaw
allow = g729
nat = no
canreinvite = nonat
dtmfmode = info
```

Далее надо подправить в файле extensions.conf секцию [home]. Вернее даже переименовать ее в [local] и добавить пару новых:

```
[nabor_sipnet]
exten => _7495XXXXXXX,1,Set(CALLERID(all)=”SipPhone”
<{sipnet_number}>)
exten => _7495XXXXXXX,2,Dial(SIP/sipnet/${EXTEN},120)
exten => _7495XXXXXXX,3,PlayBack(noanswer)
exten => _7495XXXXXXX,4,HangUp
exten => _7495XXXXXXX,305,PlayBack(busy)
exten => _7495XXXXXXX,306,HangUp

exten => _3579XXXXXXX,1,Set(CALLERID(all)=”SipPhone”
<{sipnet_number}>)
exten => _3579XXXXXXX,2,Dial(SIP/sipnet/${EXTEN},120)
exten => _3579XXXXXXX,3,PlayBack(noanswer)
exten => _3579XXXXXXX,4,HangUp
exten => _3579XXXXXXX,305,PlayBack(busy)
exten => _3579XXXXXXX,306,HangUp

[home]
include => nabor_sipnet
include => local
```

Я звоню в основном в Москву и на Кипр, потому у меня и написаны два экстеншена с кодами. Собственно говоря, заменяем в конфигах фигурные скобки на то, что в них написано и перезапускаем Астериск. Вот и все. Ничего сложного.

Единственная проблема, с которой я столкнулся, это не работает связь через NAT между внутренними номерами. Но думаю с этим тоже разберусь.

URL записки: http://silverghost.org.ua/2008/03/09/asterisk-sipnetru/

Asterisk + клиенты за NAT

Пример секции пользователя за NAT конфигурационного файла users.conf:

```
[200]
type=friend
host=dynamic
username=200
secret={user_password}
nat=yes
qualify=yes
canreinvite=no
sipreinvite=no
context=home
callerid=”User” <200>
allow=alaw
allow=ulaw
allow=gsm
```

nat - означает, что пользователь может находиться за NAT'ом.

qualify - периодическая проверка доступности устройства.

canreinvite - прогонять весь голосовой трафик через Asterisk, не совместимо с nat=yes.

sipreinvite - прогонять весь SIP-трафик через Asterisk, также не совместимо с nat=yes, если используется SIP протокол.

URL записки: http://silverghost.org.ua/2008/03/12/asterisk-klienty-za-nat/

Разное

Postfix + ClamAV-milter

Прикрутил вот к своему почтовому серверу clamav-milter.

Оказывается все предельно просто прикручивается. Ставится clamav + clamav-server + clamav-milter + freshclam[7]. Настраиваются конфиги. Сложного там ничего. Единственное с чем мне пришлось побороться и в ЖЖ меня ткнули носом (проглядел один момент), это параметр User в /etc/clamd.d/milter.conf. Он должен быть «postfix», т.е. имя юзера под которым работает Postfix, иначе почтовый сервер не сможет получить доступ к сокету clamav-milter'а.

Далее в main.cf почтовика добавляем две строки:

```
smtpd_milters = unix:/var/run/clamav-milter/clamav.sock
milter_default_action = accept
```

Перезапускаем почтовик и имеем счастье. Для обновления антивируса добавляем в крон:

```
0 */3 * * * root /usr/bin/freshclam 1>/dev/null 2>/dev/null
```

URL записки: http://silverghost.org.ua/2007/09/28/postfix-clamav-milter/

Установка logwatch в Ubuntu

Все никак не доходили руки до установки logwatch. Дело в том, что logwatch не устанавливается без MTA и подсовывает для установки Exim. Но у меня уже стоит Communigate и он не опознается как MTA.

Пришлось ковырять форумы и спрашивать народ на канале #ubuntu-ru. В общем лечится все это довольно просто. Устанавливаем equivs.

```
# apt-get install equivs
```

Далее нам надо создать .control файл для определения псевдо-MTA и создания deb-пакета.

```
# equiv-control ~/Communigate.control
# equiv-build ~/Communigate.control
# dpkg -i ~/mta-local_1.0_all.deb
```

Собственно теперь ставим сам logwatch и наслаждаемся ежедневными отчетами в

7 http://www.clamav.net

почте. :)

```
# apt-get install logwatch
```

Отдельное спасибо за терпение и помощь omnil с канала #ubuntu-ru. Советую. Там прекрасные люди. :)

URL записки: http://silverghost.org.ua/2008/03/10/communigate-pro-kak-mta-ili-ustanovka-logwatch/

Использование cron

Сегодня я хочу рассказать Вам о том, как пользоваться утилитой cron. Понимаю, что многие скажут - “Нафига это надо. Читайте маны. Уже много раз написано.”, но все таки, думаю, что многие еще не разобрались с этим делом, да и еще одна дока лишней не будет.

Cron - это утилита, которая позволяет запускать некоторые скрипты (задания) в определенное время (не только единоразово, но и периодически). Таким образом, cron - это своего рода планировщик заданий.

Конфигурируется он довольно просто. Есть основной файл crontab (обычно располагается в каталоге /etc), где прописываются задания, которые будут выполняться. По умолчанию он имеет вид:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts –report
/etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts –report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts –report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / &&
run-parts –report /etc/cron.monthly )
#
```

При детальном рассмотрении мы видим упоминание каталогов “/etc/cron.*”. Давайте пока разберемся с синтаксисом времени, а уж потом вернемся к каталогам. Синтаксис используется следующий:

Минуты Часы День_месяца Месяц День_недели Пользователь Команда

Т.е. запись

```
17 * * * * root cd / && run-parts –report /etc/cron.hourly
```

буквально означает следующее - в 17 минут каждого часа (знак *) и каждого дня запускать команду “cd / && run-parts –report /etc/cron.hourly” от имени пользователя root.

Отсюда становится понятно назначение каталогов:

- cron.hourly - каталог, содержащий задания, запускаемые ежечасно.
- cron.daily - каталог, содержащий задания, запускаемые ежедневно.
- cron.weekly - каталог, содержащий задания, запускаемые еженедельно.
- cron.monthly - каталог, содержащий задания, запускаемые ежемесячно.

В данном случае, в качестве заданий используются запускаемые скрипты или программы, которые размещают в этих каталогах.

Для примера поставим задачу. Нам необходимо запускать скрипт /home/user/script.sh каждые 3 часа от имени пользователя “user”. Формируем базовую строку:

```
* * * * * user /home/user/script.sh
```

Теперь ее исправляем до нужного нам состояния, т.е. указываем, что скрипт будет выполняться каждые 3 часа в 0 минут часа каждую субботу и воскресенье:

```
0 */3 * * 0,6 user /home/user/script.sh
```

Или же можно записать иначе:

```
0 0,3,6,9,12,15,18,21 * * 6,7 user /home/user/script.sh
```

что, в принципе, равнозначно. Вы наверное спросите почему опечатка в днях недели. Я отвечу, что воскресенье может указываться как 0 (американский стандарт), так и 7 (Российский стандарт).

Для более полной картины приведу вырезку из “man 5 crontab”:

```
field          allowed values
—————          ——————
minute         0-59
hour           0-23
day of month   1-31
month          1-12 (or names, see below)
day of week    0-7 (0 or 7 is Sun, or use names)
```

Думаю, что переводить этот кусок особого смысла не имеет, т.к. и так все понятно.

Кроме того, есть еще один каталог, который может быть весьма полезен пользователям, у которых есть SSH-доступ на сервер, но нет root-прав. Это каталог /var/spool/cron, где располагаются индивидуальные пользовательские задания. На пример, Вам необходимо добавить в крон задание, но прав на правку crontab-файла нет. Создаем в домашнем каталоге (или любом другом) файл cron примерно такого содержания:

```
SHELL=/bin/bash
MAILTO=user
0-59 * * * * /home/user/script.sh
```

Выполняем команду:

```
crontab /home/user/cron
```

в результате чего в каталоге /var/spool/cron появляется файл с именем пользователя, в данном случае user, и содержимым файла /home/user/cron.

Для просмотра списка заданий можно использовать команду “crontab -l”, для удаления - “crontab -r”, для правки - “crontab -e”.

Вот собственно и все. Не все так сложно, как кажется на первый взгляд.

URL записки: http://silverghost.org.ua/2008/05/12/ispolzovanie-cron/

Связка snmpd + mrtg

Давненько я хотел разобраться как мониторить сетевые интерфейсы локальной машины через snmp и рисовать графики через mrtg. Еще больше мне хотелось через тот же snmp мониторить загрузку CPU и памяти.

Вчера собственно я с этим и разобрался, о чем и хочу Вам поведать. :)

Ставим необходимый софт:

```
$ sudo apt-get install snmpd mrtg
```

После чего правим файл /etc/snmp/snmpd.conf всего лишь в одном месте. Было:

```
com2sec paranoid  default         public
#com2sec readonly  default         public
#com2sec readwrite default         private
```

Стало:

```
#com2sec paranoid  default         public
com2sec readonly  default         public
#com2sec readwrite default         private
```

Теперь перезапускаем snmpd:

```
$ sudo /etc/init.d/snmpd restart
```

SNMPd готов к использованию. Остается отрисовка графиков - mrtg. Правим файл /etc/mrtg.cfg, приводя его к примерно следующему виду:

```
# Global configuration

HtmlDir: /path/to/htdocs
ImageDir: /path/to/htdocs/images
LogDir: /path/to/htdocs/logs
EnableIPv6: no

############################
# Сетевой интерфейс маршрутизатора

Title[eth0]: Output interface
PNGTitle[eth0]: Interface to Internet
MaxBytes[eth0]: 64000
AbsMax[eth0]: 128000
Options[eth0]: growright, bits
SetEnv[eth0]: MRTG_INT_IP="192.168.0.1"  MRTG_INT_DESCR="eth0"
Target[eth0]: 2:public@localhost:
PageTop[eth0]: <h1> Output interface</h1>
YLegend[eth0]: bits/s
ShortLegend[eth0]: b/s
Legend1[eth0]: Incoming Traffic
Legend2[eth0]: Outgoing Traffic
Legend3[eth0]: Maximum Incoming Traffic
Legend4[eth0]: Maximum Outgoing Traffic
LegendI[eth0]:  In:
LegendO[eth0]:  Out:
WithPeak[eth0]: ymwd

############################
```

```
# CPU

LoadMIBs: /usr/share/snmp/mibs/UCD-SNMP-MIB.txt
Target[cpu]:ssCpuRawUser.0&ssCpuRawUser.0:public@127.0.0.1+ssCpuR
awSystem.0&ssCpuRawSystem.0:public@127.0.0.1+ssCpuRawNice.0&ssCpu
RawNice.0:public@127.0.0.1
RouterUptime[cpu]:  public@127.0.0.1
MaxBytes[cpu]: 100
Title[cpu]: CPU Load
PageTop[cpu]: <H1>Active CPU Load %</H1>
Unscaled[cpu]: ymwd
ShortLegend[cpu]: %
YLegend[cpu]: CPU Utilization
Legend1[cpu]: Active CPU in % (Load)
Legend2[cpu]:
Legend3[cpu]:
Legend4[cpu]:
LegendI[cpu]: Active
LegendO[cpu]:
Options[cpu]: growright,nopercent

##########################
# Memory

LoadMIBs: /usr/share/snmp/mibs/HOST-RESOURCES-MIB.txt
Target[mem]: .
1.3.6.1.4.1.2021.4.6.0&.1.3.6.1.4.1.2021.4.6.0:public@localhost
PageTop[mem]: <H1>Free Memory</H1>
Options[mem]: nopercent,growright,gauge,noinfo
Title[mem]: Free Memory
MaxBytes[mem]: 1000000
kMG[mem]: k,M,G,T,P,X
YLegend[mem]: bytes
ShortLegend[mem]: bytes
LegendI[mem]: Free Memory:
LegendO[mem]:
Legend1[mem]: Free memory, not including swap, in bytes
```

Итого мы имеем отрисовку графиков сетевого интерфейса, загрузку процессора, использования памяти. Остается только нарисовать общий файл со всеми графиками:

```
$ sudo indexmaker /etc/mrtg.cfg > /path/to/htdocs/index.html
```

Советую заглянуть в комментарии к данной записи, там есть дополнение для Cisco.

URL записки: http://silverghost.org.ua/2008/05/04/svyazka-snmpd-mrtg/

Учимся использовать screen

Вы залогинились на ваш удаленный сервер через ssh, радостно стучите по клавиатуре, делая свои какие-то админские дела и опа! Символы перестали печататься и вывалилось то самое сообщение, которое, порой, вызывает непреодолимое желание разбить клавиатуру. Connection closed. Знакомая ситуация? Да-да, ваша сессия только что аварийно оборвалась и все придется делать заново… Этого можно избежать, если использовать screen. Он не только позволит сохранить вашу сессию в целости и сохранности, но еще и позволит держать открытыми несколько сессий в одном окошке терминала

Что такое screen?

Сначала посмотрим man-страницу: “Screen - это полноэкранный оконный менеджер, который позволяет разделить физический терминал между несколькими независимыми процессами (обычно интерактивными оболочками)”. У screen есть несколько отличительных особенностей, которые могут сильно помочь при выполнении задач на удаленных серверах через ssh. Я расскажу о трех, самых часто используемых мной фичах: многооконность, логирование и сессии. За более подробными деталями придется обратиться к man-странице.

Установка screen

Скорее всего, screen уже есть в вашей системе. Проверить это можно командой

```
$ which screen
```

Если which не дал результатов, то установите screen при помощи пакетного менеджера вашей системы. На моих серверах, в основном, CentOS и Debian, поэтому я ставлю screen так:

```
# yum install screen (ДЛЯ CentOS)
# apt-get install screen (ДЛЯ Debian)
```

Попадается и FreeBSD, в этом случае я использую порты:

```
# cd /usr/ports/sysutils/screen
make install clean
```

Использование screen

Screen запускается из командной строки также, как и любое приложение :)

```
$ screen
```

Вы можете получить сообщение о том, что screen запущен, а можете и не получить… Зависит от вашей системы. Если вы не получили сообщения, то вы можете подумать, что ничего не произошло. Однако это не так. Вы *уже* внутри терминала, запущенного в screen. Это нормальный полнофункциональный шелл, за исключением нескольких специальных команд. Screen использует клавиатурную комбинацию Ctrl+A для подачи команд терминалам внутри себя. Попробуйте нажать Ctrl+A, а затем ? Вы увидите примерно следующее:

```
Screen key bindings, page 1 of 2.

Command key: ^A Literal ^A: a

break ^B b lockscreen ^X x reset Z
clear C log H screen ^C c
colon : login L select ” ‘
copy ^[ [ meta a silence _
detach ^D d monitor M split S
digraph ^V next ^@ ^N sp n suspend ^Z z
displays * number N time ^T t
fit F only Q title A
flow ^F f other ^A vbell ^G
focus ^I pow_break B version v
help ? pow_detach D width W
```

```
history { } prev ^P p ^? windows ^W w
info i readbuf < wrap ^R r
kill K redisplay ^L l writebuf >
lastmsg ^M m remove X xoff ^S s
license , removebuf = xon ^Q q
[Press Space for next page; Return to end.]
```

Screen воспринимает командные клавиатурные комбинации после нажатия Ctrl+A. Вы можете изменить это поведение при помощи конфиг-файла $HOME/.screenrc

Многооконность

Screen, как и большинство оконных менеджеров, поддерживает работу с несколькими окнами. Это очень удобно для выполнения параллельных задач без открытия новых ssh-сессий. Например, у меня постоянно открыто четыре или пять сессий с несколькими задачами в каждой. Раньше мне бы пришлось открыть порядка 15 терминалов, логинов, сессий.. Утомительно, не так ли? Эти неудобства полностью решает screen. Теперь я вполне могу обойтись одним терминалом.

Новое окно открывается комбинацией клавиш “Ctrl+a c”. После нажатия вы увидите новый терминал с вашим приглашением в том же окне. При этом предыдущие окна также продолжают работать. Давайте попробуем: запустите screen и в нем top

```
Mem: 506028K av, 500596K used, 5432K free,
0K shrd, 11752K buff
Swap: 1020116K av, 53320K used, 966796K free
393660K cached

PID USER PRI NI SIZE RSS SHARE STAT %CPU %ME
6538 root 25 0 1892 1892 596 R 49.1 0.3
6614 root 16 0 1544 1544 668 S 28.3 0.3
7198 admin 15 0 1108 1104 828 R 5.6 0.2
```

Теперь откройте новое окно, нажав “Ctrl+a c”

```
[boombick@server ] $
```

Вернитесь обратно, нажав Ctrl+a n

```
Mem: 506028K av, 500588K used, 5440K free,
0K shrd, 11960K buff
Swap: 1020116K av, 53320K used, 966796K free
392220K cached

PID USER PRI NI SIZE RSS SHARE STAT %CPU %ME
6538 root 25 0 1892 1892 596 R 48.3 0.3
6614 root 15 0 1544 1544 668 S 30.7 0.3
```

top остался в прежнем состоянии. Вы можете создать несколько окон и переключаться между ними используя Ctrl+a n для переключения на следующеее окно и Ctrl+a p для переключения на предыдущее. При этом каждый запущенный процесс останется в рабочем состоянии.

Отключаемся от screen

Есть два способа отключиться от screen: первый - это просто разлогиниться. Вы можете использовать клавиатурную комбинацию Ctrl+a K или просто набрать exit. Этот

способ “убьет” текущее окно, если у вас их несколько или совсем остановит screen. Второй способ заключается в *отсоединении*. Этот способ оставляет текущий процесс запущенным и просто отключает вас от терминала. Например, если вы через ssh-сессию запускаете какой-то очень длительный процесс, не требующий вашего внимания, то вы можете просто отключиться от screen при помощи Ctrl+a d. Это вернет вас обратно в исходную оболочку. Все процессы, запущенные в screen, при этом остаются работающими и вы можете подключиться к ним позже.

Подключение к сессии

Вы компилируете большую программу на удаленном сервере, используя screen. И конечно же, по законам Мэрфи, соединение обрывается по независящим от вас причинам. Не стоит впадать в панику, screen все сохранил :) Просто соединитесь с сервером еще раз и посмотрите запущеные под screen процессы при помощи

```
[root@server root]# screen -ls
There are screens on:
31619.ttyp2.server (Detached)
4731.ttyp2.server (Detached)
2 Sockets in /tmp/screens/S-root.
```

В этом примере запущено две screen-сессии. Для подключения к нужной из них используйте команду

```
[root@server root]#screen -r 31619.ttyp2.server
```

Просто используйте screen с флагом r и именем сессии для повторного подключения. Это очень удобно. Можно, например, запустить какой-то длительный процесс на работе и, вернувшись домой, продолжить контроль за его выполнением.

Логирование

Мне кажется очень важным порой сохранять полный лог своих действий. К счастью, screen легко с этим справляется. Просто активируйте логирование нажатием Ctrl+a H. Screen будет продолжать логирование на протяжении всего процесса работы. Бывает очень полезно вернуться назад и посмотреть порядок необходимых действий.

И еще немного…

Screen может вести мониторинг активности окна. Если вы качаете что-то большое, компилируете программу или просто выполняете длительный процесс, вы можете долгое время наблюдать пустой терминал без признаков активности. А процесс, тем временем, продолжает выполняться. Или, наоборот, смотреть на поток отладочной информации, ожидая окончания процесса. Для начала слежения перейдите в терминал, который вы хотите наблюдать и нажмите Ctrl+a M для слежения за активностью (сработает при появлении новой информации) или Ctrl+a _ для слежения за бездействием (сработает при прекращении поступления информации в терминал). Затем вы можете спокойно переключиться в другое окно или создать новое. При наступлении события, screen предупредит вас об этом сообщением с номером окна в заголовке терминала. Для быстрого переключения в это окно используйте Ctrl+a ” (это символ кавычки). Вы увидите список всех активных окон на данный момент. Для перехода в нудное можно использовать стрелки или просто набрать номер нужного окна. Для прекращения наблюдения перейдите в нужное окно и отмените мониторинг той же командой. Например, для прекращения наблюдения за активностью нажмите Ctrl+a M.

Источник: http://boombick.org/blog/posts/22

URL записки: http://silverghost.org.ua/2007/11/28/uchimsya-ispolzovat-screen/

Ubuntu + webcam

Произошел тут у нас случай на работе. Повадился кто-то за комп на выходных садиться и по сайтам лазить. И пароль ведь подсмотрел. Решили мы это тело проучить, пароль пока менять не стали, а поставили веб-камеру с настроенным сервером вещания в инет. :)

Собственно про настройку сервера под Ubuntu я и хочу рассказать. Значит нам нужен сам сервер вещания — camserv:

```
$ sudo apt-get install camserv
```

Поставили, теперь идем настраивать:

```
$ sudo mcedit /usr/share/doc/camserv/examples/webcam.html
```

Заменяем строчку «your.camserv.host» на наш IP адрес и стартуем camserv:

```
$ sudo /etc/init.d/camserv start
```

После чего открываем этот html файл в браузере и наслаждаемся картинкой. При желании его можно положить в Apache и расшарить в инет, но тут уже надо писать не внутреннюю ИПшку, а внешнюю и пробрасывать на машину с camserv порт 9192. Но это тема отдельная, т.к. тут уже участвует iptables и Apache.

Переезд состоялся или танцы с бубном вокруг MySQL

Сегодня дошли руки до своего сервера.

В общем почистил я его пылесосом, на проце поменял пасту и решил его “обновить”. Переставил в другое место и занялся переустановкой системы. Давно хотел попробовать Ubuntu в качестве сервера. В общем установка заняла минут 15-20 времени, дальше пошел перенос данных с другого винта из под Федоры.

BIND9 перенес без проблем. В Апаче пришлось по-включать некоторые модули, для работы SSL, rewrite, включить AddDeafultCharset. долго не мог понять почему не заводятся виртуальные хосты нормально, потом нашел. Надо было в /etc/apache2/sites-available/default добавить порт 80 после *.

В общем все заработало, оставались базы MySQL. С этим была проблема, которую удалось решить после некоторых танцев с бубном. Оказывается, что в Ubuntu по умолчанию в базе есть еще один юзер, для служебных нужд. Зовется он debian-sys-maint. Если тупо заменить базы из Федоры поверх баз Убунты, то Мускул отваливается и не стартует.

Делаем финт ушами: выдираем строку из Убунты из базы mysql.user. Останавливаем сервер. Подменяем базы. Правим права на файлы баз. После чего стартуем mysqld_safe руками и применяем параметры юзера debian-sys-maint.

После всего этого прибиваем процессы mysqd_safe и mysqld и стартуем уже нормально. У меня получилось и все заработало.

URL записки: http://silverghost.org.ua/2008/02/11/pereezd-sostoyalsya-ili-tancy-s-bubnom-vokrug-mysql/

Скрипт бекапа баз данных MySQL

Долго я искал нормальный скрипт для резервного копирования баз данных MySQL и ничего подходящего для себя не нашел.

Чего мне не хватало:

1. Пакетный бекап баз.
2. Разные каталоги для бекапов.
3. Ротация резервных копий.

Пришлось мне писать свой скрипт для этого дела. В общем на Ваш суд:

MySQL DB Backup[8] умеет хранить резервные копии баз данных в отдельных каталогах для каждой базы, что позволяет разнести базы по каталогам пользователей; позволяет управлять резервированием баз данных из одного места, что гораздо удобнее, чем ручное копирование; проводить ротацию файлов копий, ограничив количество этих копий; автоматически менять владельца и группу файла для корректного доступа пользователей к файлам резервных копий.

Инструкция по установке

1. Внесите в массив параметры баз данных и путей к каталогам резервных копий, настройте остальные параметры конфигурации в файле mysqldbbackup.ini.

Один из разделов конфигурации обязательно должен называться Options. В нем находятся настройки хранения, ротации, доступа к базе данных. Для примера:

```
[Options]
removedays = 1,4,6
nob = 10
compress = 9
dbuser = root
dbpass = Mega$uperPa$sword
```

– removedays отвечает за настройку дня недели, в которые будет проводиться удаление старых файлов резервных копий. Дни недели указываются через запятую (0 — воскресенье). Можно указать “*”, что означает каждый день;

– nob (number of backups) отвечает за количество хранимых бекапов;

– compress - коэффициент сжатия (0 — 9);

– dbuser - имя суперполльзователя (обычно root);

– dbpass - пароль для указанного пользователя.

Далее идут разделы для копируемых баз данных:

```
[DataBase]
db = dbname
archpath = /path/to/backup
owner = user:group
```

– DataBase - идентификатор базы данных;

– db - имя базы данных;

– owner - имя пользователя и группа, выставляемые на файл бекапа после его

8 http://silverghost.org.ua/download/scripts/mysqldbbackup.tar.bz2

создания.

2. Установите права запуска на скрипт 700, владельца и группу root.

3. Внесите в crontab запуск скрипта по расписанию:

```
0 1 * * * root php /path/to/mysqldbbackup.php
```

4. Все. Скрипт находится в рабочем состоянии.

URL записки: http://silverghost.org.ua/2008/05/03/skript-bekapa-baz-dannyx-mysql/

OpenFire IM gateway plugin и русский язык

Обновил свой OpenFire и поставил плагин “IM Gateway”. Проблема вылезла стразу. Подключается, работает, но не понимает русский язык. Иду в админку, прописываю windows-1251 кодировку, не помогает.

В общем перерыл кучу всего и только в одном месте нашел как заставить его работать нормально.

Надо в файле /usr/share/openfire/plugins/gateway/web/WEB-INF/options/icq.xml исправить кодировку на windows-1251, перезагрузить OpenFire и все будет ок. :)

Я теперь отказался от всех асечных клиентов и безумно этому рад. :)

URL записки: http://silverghost.org.ua/2008/04/28/openfire-im-gateway-plugin-i-russkij-yazyk/