v14.03 (№7)

Выбираем OpenSource решение для организации корпоративных видеоконференций

user And LINUX

Больше чем user

Быстрый курс IPv6 в Linux

Находка для Штирлица или урок криптографии

Анализ безопасности компьютерных сетей

Создаем SOHO-сервер на базе Zentyal

Трансляция изображения с IP-камер

Обновлен репозиторий UALinux. Добавлено более 20-и новых приложений для Ubuntu и Ubuntu-совместимых дистрибутивов.
Подробнее http://ualinux.com/ru/46580-obnovleniya-v-repozitorii-ualinux-dlya-ubuntu

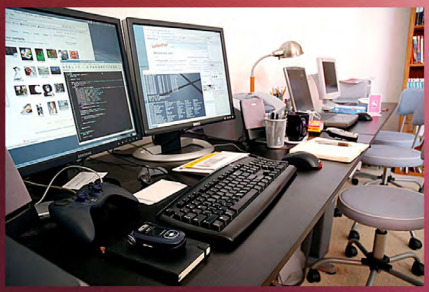
UA Linux
open source ■ open future

ubuntu BusinessPack

Операционная система, которая идеально подходит для использования на персональных компьютерах и ноутбуках. Она ориентирована на простоту использования и удобство работы.

Включена необходимая подборка программного обеспечения, которая позволяет создать удобное рабочее окружение в корпоративной среде предприятия или на домашнем компьютере.

Ubuntu Business Pack это:



- **простая установка операционной системы не требующая особых знаний;**
- **уверенность в том, что на компьютере установлено только лицензионное программное обеспечение;**
- **это низкая цена по сравнению с аналогами;**
- **создание рабочего места без дополнительных финансовых затрат. Это существенно экономит бюджет организаций;**
- **идеальное решение для перехода на Linux с Windows, если вы все еще используете windows-приложения и игры;**
- **полная поддержка в системе русского, украинского и английского языков;**
- **отсутствие необходимости затрат на антивирусную защиту.**

Программное обеспечение имеет понятный графический интерфейс и полностью совместимо с популярными форматами документов, поэтому переход не вызывает никаких проблем с переносом данных и переквалификацией сотрудников.

поддержка широкого спектра современного оборудования;
дополнительные драйвера для видео-карт, wi-fi адаптеров и принтеров;
возможность использовать Windows-драйвера для WiFi-адаптеров USB;
управление веб-камерами.

безопасность и надежная защита от вирусов;
проверка файлов на вирусы в режиме реального времени (актуально в случае запуска windows-приложений);
защита от вирусных атак системы и электронной почты;
проверка на спам.

поддержка мультимедия (аудио - видео) различных форматов (avi, divX, mp4, mkv, amr, aac, Adobe Flash и многие другие)
просмотр защищенных, зашифрованных лицензионных, двухслойных DVD и Bluray дисков

полный набор офисных компонент (тексты, таблицы, презентации) совместимых с форматами MS Office
включена поддержка импорта файлов MS Visio
поддержка различных типов архивов (RAR, ACE, ARJ и других);

поддержка windows-приложений (гарантированный запуск более 130 приложений и более 600 игр)

OpenJDK

полноценная поддержка Java-приложений;
гарантированная работа онлайн банк-клиентов, таких как Приват24
гарантированная работа онлайн-бухгалтерии, таких как iFin.

Здравствуйте, уважаемые наши читатели!

Наш коллектив рад вам представить новый номер нашего журнала.

Специально для этого номера Сергей Вовк прислал статью, посвященную написанию программы шифровки данных — «Находка для Штирлица или урок криптографии»

На современных серверах разворачивать систему без использования RAID-массива решится далеко не каждый админ. О том, как с ними работать рассказывается в «MDAdm. Работа с RAID на Ubuntu».

Как всем сегодня известно, IPv4 адреса постепенно отходят в прошлое. Приходит эра IPv6. О том, что это такое и как этим пользоваться читайте в статье «Быстрый курс IPv6 в Linux»

Каким бы хорошим админ не был, рано или поздно ему придется определять и исправлять сбои на сервере. В этом номере, в статье «Первые 5 минут устранения неполадок на сервере» даются советы о том, как максимально быстро определить проблемные места.

Так же продолжается публикация статей в разделе «CYBERCRIME». Сегодня там будут рассмотрена атака при помощи спуффинга пакетов.

Читайте наш журнал и оставайтесь с нами. Следующий выпуск постараемся сделать не менее интересным.

С уважением,
Коллектив «Больше чем Юзер»

НАД ВЫПУСКОМ РАБОТАЛИ:

Якимчук Сергей
Попов Владимир
Шарай Игорь
Россошанский Андрей
Звенигородская Анастасия
Кирильчук Виктор
Безруков Марк

Содержание

MDAdm
Работа с RAID на Ubuntu

Рано или поздно при настройке серверов приходится столкнуться с такой вещью как RAID.

На брендовых серверах они зачастую полностью хардварные, но тем не менее очень часто приходится сталкиваться и с софтварным рейдом.

Построение RAID

Построим на сервере RAID1.

Сначала создадим одинаковые разделы на дисках sdb и sdc

```
#fdisk /dev/sdb
command (m for help): n
Partition type:
p primary (0 primary, 0 extended,
4 free)
e extended
Select (default p): p
Partition number (1-4, default
1): 1
First sector (2048-16777215,
default 2048):
Using default value 2048
Last sector, +sectors or
+size{K,M,G} (2048-16777215,
default 16777215): +5G
Command (m for help): t
Hex code (type L to list codes):
83
Changed system type of partition
1 to 83
Command (m for help): w
The partition table has been
altered!
Calling ioctl() to re-read
partition table.
Syncing disks.
```

Аналогично сделаем и для диска sdc

Установим утилиту работы с RAID

```
# apt-get install mdadm
```

Теперь соберем RAID 1

```
# mdadm --create --verbose /dev/
md0 --level=1 --raid-devices=2 /
dev/sdb1 /dev/sdc1
```

После сборки состояние RAID можно просмотреть при помощи команды

```
# cat /proc/mdstat
```

В результате мы должны получить вывод что-то похожее на

```
personalities : [raid1]
md0 : active raid1 sdc1[1]
sdb1[0]
5238720 blocks super 1.2 [2/2]
[UU]
unused devices: <none>
```

Теперь можно на созданном RAID-разделе создавать файловую систему и подключать ее к системе.

```
# mkfs.ext4 /dev/md0
# mkdir /mnt/raid
# mount /dev/md0 /mnt/raid
```

Так же для дальнейшей проверки корректности работы рейда создадим на RAID-разделе файл:

```
# touch /mnt/raid/test.txt
```

Ошибка при работе с RAID

После того, как мы создали RAID он у нас определяется как устройство /dev/md0, однако после перезагрузки такого устройства в системе не будет, а вместо него появится /dev/md127. Тут можно или в дальнейшем использовать именно такое имя устройства, или, что обычно удобнее, насильно объяснить системе, что наш RAID-раздел именно /dev/md0 и ни как иначе. Для этого выполним команду:

```
# mdadm -Db /dev/md0 > /etc/
mdadm/mdadm.conf
```

В результате выполнения этой команды в файле /etc/mdadm/mdadm.conf будет строка:

```
ARRAY /dev/md0 metadata=1.2
name=ub-4:0 UUID=7da67e34:3d29e3a
1:bdf36edd:6be26e60
```

После этого необходимо обновить образ initramfs:

```
# update-initramfs -u
```

Теперь после перезагрузки наш RAID-раздел будет определяться как /dev/md0.

Деградация и восстановление RAID

Посмотрим как можно провести деградацию рейда. Естественно, в реальной системе диск обычно вылетает сам и нет необходимости специально объявлять его сбойным, но мы воспользуемся возможности утилиты mdadm и объявим часть RAID – /dev/sdb1 сбойным.

```
# mdadm /dev/md0 --fail /dev/sdb1
```

Посмотрим теперь на состояние RAID

```
# cat /proc/mdstat
```

Мы должны увидеть, что /dev/sdb1 имеет некие проблемы и рейд деградирован:

```
Personalities : [linear]
[multipath] [raid0] [raid1]
[raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[2](F)
sdc1[1]
5238720 blocks super 1.2 [2/1]
[_U]
unused devices: <none>
```

Теперь с помощью fdisk создадим на устройстве /dev/sdd раздел такого же размера, как и /dev/sdc1. После этого удалим /dev/sdb1 из RAID

```
# mdadm /dev/md0 --remove /dev/
sdb1
```

И добавим новый раздел /dev/sdd1

```
# mdadm /dev/md0 --add /dev/sdd1
```

Если мы сразу же посмотрим на состояние RAID

```
# cat /proc/mdstat
```

То мы увидим, что RAID у нас опять состоит из двух нормальных дисков и в данный момент происходит его синхронизация:

```
Personalities : [linear]
[multipath] [raid0] [raid1]
[raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdd1[2]
sdc1[1]
5238720 blocks super 1.2 [2/1]
[_U]
[=>...................] recovery
= 6.2% (329984/5238720)
```

```
finish=1.2min speed=65996K/sec
unused devices: <none>
```

Если мы теперь примонтируем наш RAID

```
# mount /dev/md0 /mnt/raid/
```

То увидим, что файл, который мы до этого создавали на месте и ничего никуда не пропало.

```
# ls /mnt/raid/
lost+found test.txt
```

По материалам сайта:
yakim.org.ua

Быстрый курс IPv6 в Linux

Возможно вы привыкли к IPv4, однако нравится вам это или нет, постепенно наступает эра IP шестой версии. Ядро, начиная с версии 2.1, обладает поддержкой IPv6, так что вам ничего не нужно доустанавливать. Убедитесь лишь, что в вашей системе установлены программы ping6, ip и ifconfig. И давайте сразу договоримся: здесь мы говорим не об «IP», а об «IP-адресах». IP – это Internet Protocol, а не IP-адреса. Небрежность в разговоре равносильна небрежности в делах, что ни к чему хорошему не приводит.

Преимущества IPv6

Итак, чем же IPv6 лучше своего предшественника? Кроме того, что мы избавляемся от проблем в IPv4-адресацией, у нас теперь есть:

- отсутствие головной боли с частными подсетями;
- использование NAT скорее исключение, чем правило;
- упрощённая маршрутизация;
- говорим «прощай» DHCP.

Основным недостатком IPv6 можно назвать длинные адреса в шестнадцатеричном виде. Четыре фрагмента адреса IPv4, разделённые точками, легки для запоминания, а вот восемь кусков шестнадцатеричных чисел запомнить куда труднее.

Моя система поддерживает IPv6?

Как узнать, включена ли поддержка IPv6 в вашем Linux?

Очень просто:

```
$ cat /proc/net/if_inet6
00000000000000000000000000000000 01
01 80 10 80       lo
fe8000000000000020b6afffeef7e 8d
02 40 20 80     eth0
```

Если вы увидите в своей системе вывод, подобный этому – значит всё отлично, поддержка IPv6 в вашей системе включёна. Вообще, все современные Linux-дистрибутивы поддерживают IPv6 «из коробки».

Пинг IPv6

Если вам нужно выполнить пинг IPv6-системы, вам понадобится утилита

ping6. Следующая команда отправляет два пакета к localhost:

```
$ ping6 -c2 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1
ttl=64 time=0.043 ms
64 bytes from ::1: icmp_seq=2
ttl=64 time=0.054 ms
--- ::1 ping statistics ---
2 packets transmitted, 2
received, 0% packet loss, time
999ms
rtt min/avg/max/mdev =
0.043/0.048/0.054/0.008 ms
```

::1 – это сокращение от 0000:0000:0000 :0000:0000:0000:0000:0001. Любая непрерывная последовательность нулей может быть заменена на пару двоеточий, а любая четвёрка, состоящая из нулей, может быть заменена на один ноль. Для нашего примера это 0.0.0.0.0.0.0.1.

Исследование сети

Хотите узнать, есть ли кто-то в вашей сети, кто имеет поддержку IPv6? Легко!

```
$ ping6 -c4 -I eth0 ff02::1
PING FF02:0:0:0:0:0:0:1(ff02::1)
from fe80::20d:b9ff:fe05:25b4
eth0: 56 data bytes
64 bytes from
fe80::20d:b9ff:fe05:25b4: icmp_
seq=1 ttl=64 time=0.301 ms
64 bytes from
fe80::20b:6aff:feef:7e8d: icmp_
seq=1 ttl=64 time=3.69 ms (DUP!)
[snip duplicate lines]
-- FF02:0:0:0:0:0:0:1 ping
statistics ---
4 packets transmitted, 4
received, +6 duplicates, 0%
packet loss, time 3000ms
rtt min/avg/max/mdev =
0.254/1.698/8.911/2.593 ms
```

Представленный в примере вывод команды ping6 говорит что да, есть два адреса: fe80::20b:6aff:feef:7e8d и fe80::221:97ff:feed:ef01. Обратите внимание, что вы обязательно должны указать имя интерфейса программе ping6, даже если этот интерфейс в вашей системе один-единственный.

ff02::1 – это сокращенный вариант адреса ff02:0:0:0:0:0:0:1, который является специальным мультикаст-адресом, предназначенным для отправки пакетов всем link-local хостам.

Адреса link-local – это все адреса в диапазоне fe80::/10, который по смыслу эквивалентен диапазону 169.254.0.0/16 в IPv4. Любой адрес из этого диапазона предназначен для автоконфигурирования ОС и пакеты, отправленные с такого адреса не пропустит ни один маршрутизатор, ограничивая их существование в пределах сегмента локальной сети. Протокол IPv6 спроектирован так, что интерфейс обязан иметь link-local адрес, даже если у него есть другой.

После того, как ваш хост обменялся данными с другими хостами по IPv6, адреса последних попадут в таблицу окружения (neighbor table) IPv6 (это что-то вроде ARP-таблицы). Содержимое neighbor table вы можете просмотреть при помощи утилиты ip:

```
$ ip -6 neigh show
fe80::221:97ff:feed:ef01 dev
eth0 lladdr 00:21:97:ed:ef:01 nud
reachable
fe80::20b:6aff:feef:7e8d dev
eth0 lladdr 00:0b:6a:ef:7e:8d nud
reachable
```

Здесь «nud reachable» означает «статус network unreachability detection установлен в состояние reachable», то есть сетевой хост доступен. Каждая запись в neighbor table временная и хранится в течение нескольких минут после последней сетевой активности в направлении хоста.

Использование сетевых имён

Оставим пока «правильные» сисадминовские методы настройки соответствия имён хостов их IP-адресам, а сейчас воспользуемся старым добрым /etc/hosts. Представим, у вас есть три хоста в одном физическом сегменте сети, имеющие имена fatfreddy, phineas, и franklin. Создадим на каждом из хостов записи в /etc/hosts:

```
fe80::20b:6aff:feef:7e8d
fatfreddy
fe80::221:97ff:feed:ef01  phineas
fe80::3f1:4baf:a7dd:ba4f
franklin
```

Теперь можно пинговать системы по имени:

```
$ ping6 -I eth0 phineas
PING phineas(phineas) from
fe80::221:97ff:feed:ef01 eth0: 56
data bytes
64 bytes from phineas: icmp_seq=1
ttl=64 time=17.3 ms
```

SSH и SCP

SSH и SCP умеют работать с IPv6. Внимание, при передаче параметров этим утилитам из командной строки, имеются определённые синтаксические особенности, так что будьте внимательны! Если у вас настроено корректное разрешение имён IPv6-хостов, то разницы при вызове утилит вообще никакой. Вы таким же образом можете подключаться к удалённой оболочке:

```
$ ssh user@remotehost
```

и копировать файлы:

```
$ scp filename user@remotehost:/
home/username/directory/
```

А вот в случае использования IPv6-адресов всё чуток сложнее.

Установка SSH-сессии:

```
$ ssh phineas@
fe80::221:97ff:feed:ef01%eth0
```

И опять же, если вы используете link-local адрес, то вы должны указать имя интерфейса с которого будете осуществлять подключение. Как показано выше, делается это путём добавления знака процента и имени интерфейса. Вызов scp имеет ещё более дурацкий синтаксис:

```
$ scp test.txt phineas@\
fe80::221:97ff:feed:ef01%eth0\]
:phineas@fe80::221:97ff:feed:
ef01%eth0's
password:
test.txt 100%   19     0.0KB/s
00:00
```

IPv6-адрес вместе именем интерфейса необходимо заключать в квадратные скобки, а квадратные скобки в свою очередь экранировать, чтобы оболочка не узрела в них спецсимволов.

Какой у меня IPv6-адрес?

Команда «ifconfig -a» выводит информацию обо всех сетевых интерфейсах, находящихся в системе: как о физических, так и о виртуальных. Получив информацию по конкретному интерфейсу вы можете воспользоваться grep, чтобы отфильтровать информацию об IPv6-адресе интерфейса:

```
$ ifconfig eth0 | grep 'inet6
addr:'
inet6 addr:
fe80::20d:b9ff:fe05:25b4/64
Scope:Link
```

Интернет

Работа в сегменте локальной сети – это, конечно, хорошо, но как обстоят дела в работой IPv6 в глобальной сети? Для того, чтобы воспользоваться IPv6 в интернет, ваш провайдер должен выдать вам «чистый» IPv6-адрес. Таких провайдеров сегодня очень мало. Свою первую IPv6-сеть автор настроил в 2004 году и если бы вы сказали ему, что спустя семь лет ситуация с IPv6 останется почти такой же, он бы вам не поверил. Да, 2012-й в этом плане не очень-то отличается от 2004-го. Тем временем, вы можете воспользоваться IPv6-over-IPv4 tunnel broker-ами (как, впрочем, вы могли это сделать и в 2004-м!), такими как SixXS или Hurricane Electric.

8 июня 2011 года – мировой день IPv6

В этот день Google, Comcast, Facebook, Yahoo!, Akamai и Limelight Networks, а также ещё некоторые провайдеры включат доступ к своим хостам по IPv6 на 24 часа. Убедитесь в том, что вы готовы принять участие в тестировании и узнайте больше на test-ipv6.com. Так же, как и в случае с IPv4, ваш провайдер должен выделить вам блок адресов IPv6. Адреса из этого блока являются глобальными адресами, находящимися в диапазоне 2000::/3. Давайте в качестве эксперимента попробуем присвоить какому-нибудь сетевому интерфейсу адрес из этого диапазона:

```
# ip -6 addr add 2001::1/64 dev
eth0
```

Теперь взглянем, что у нас получилось:

```
$ ifconfig eth0 |grep "inet6
addr:"
inet6 addr: 2001::1/64
Scope:Global
inet6 addr:
fe80::20b:6aff:feef:7e8d/64
Scope:Link
```

Удалить назначенный адрес можно при помощи всё той же утилиты ip:

```
# ip -6 addr del 2001::1/64 dev
eth0
```

В реальной жизни вам будут выделять большой блок адресов, например 256 или больше, так что вам нужно будет настроить сервер для их автоматической раздачи вашим хостам. В следующей части статьи мы с вами рассмотрим, как это делается. Также рассмотрим, как необходимо настроить файерволл и службу имён для работы с IPv6.

По материалам сайта:
gnu.su

Первые 5 минут устранения неполадок на сервере

Когда наша команда еще занималась вопросами эксплуатации, оптимизации и масштабирования, нам приходилось иметь дело с отладкой медленно работающих приложений и целых инфраструктур, часто большого размера (представьте CNN или the World Bank). Горящие сроки, экзотические стеки технологий и недостаток информации обычно гарантировали незабываемые впечатления.

Причины неполадок редко были очевидными; ниже я привожу список шагов, с которых мы обычно начинали поиск проблемы.

Не спешите бросаться на сервера, сперва нужно выяснить, что уже известно о системе и специфике проблемы. Не стоит тратить время на поиск проблемы вслепую.

Несколько обязательных вопросов, требующих ответа:

• Какие конкретно наблюдаются симптомы? Подвисания? Ошибки?

• Когда проблема была замечена впервые?

• Воспроизводится ли она?

• Есть ли закономерность (например, происходит каждый час)?

• Какие были последние изменения в системе (код, сервисы, стек приложений)?

• Влияет ли проблема на определенную группу пользователей (авторизированных, не авторизированных, с общим географическим расположением…)?

• Имеется ли документация на архитектуру (физическую и логическую)?

• Используется ли система мониторинга? Munin, Zabbix, Nagios, New Relic… Подойдет любая.

• Ведется ли (централизированное) журналирование? Loggly, Airbrake, Graylog…

Последние два пункта представляют собой наиболее удобные источники информации, но не возлагайте на них больших надежд: как ни печально, именно мониторинг и журналирование часто отсутствуют. Если не повезло, сделайте заметку, что это нужно поправить, и двигайтесь дальше.

Кто здесь?

```
$ w
$ last
```

Не критично, но обычно не стоит заниматься устранением неполадок в системе, в то время когда с ней играются другие люди. На кухне достаточно одного повара.

Что делали в системе?

```
$ history
```

Всегда полезно посмотреть на историю команд в комбинации с информацией о том, кто ранее заходил в систему. Не забывайте про ответственность: то, что вы администратор, не дает вам права нарушать чужую конфиденциальность.

Маленькая заметка в уме на потом – вы можете задать переменную окружения HISTTIMEFORMAT, чтобы была возможность отслеживать время, когда выполнялись команды из истории.

Нет ничего более раздражающего, чем анализ устаревшего списка команд, не имеющих отношения к проблеме…

Что запущено?

```
$ pstree -a
$ ps aux
```

Вывод ps aux содержит, как правило, много подробной информации о процессах, тогда как pstree -a выдает наглядную и лаконичную картину запущенных процессов, вместе с родительской иерархией.

«Слушающие» сервисы

```
$ netstat -ntlp
$ netstat -nulp
$ netstat -nxlp
```

Я предпочитаю выполнять эти команды отдельно, в основном потому что я не люблю смотреть на все сервисы одновременно. Тем не менее, netstat -nalp тоже подойдет и я бы не опускал опцию -n (IP-адреса, мне кажется,воспринимаются лучше).

Определите запущенные службы и выясните должны ли они выполнятся. Посмотрите какие порты находятся в слушающем состоянии. PID слушающего процесса можно всегда найти в выводе ps aux. Это может оказаться очень полезным, особенно когда в системе одновременно запущены несколько Java или Erlang процессов.

Обычно, мы стараемся, чтобы наши системы были более или менее специализированы, с небольшим количеством сервисов на каждой из них. Если вы видите десятки слушающих портов, наверно стоит отметить это себе в уме, чтобы потом разобраться как это можно почистить или реорганизовать.

Процессор и память

```
$ free -m
$ uptime
$ top
$ htop
```

Эти команды должны ответить на несколько вопросов:

• Есть ли свободная память? Происходит ли своппинг на диск?

• Насколько загружены процессоры? Сколько ядер доступно на сервере? Перегружены ли какие-то из них?

• Что больше всего нагружает систему? Какое у системы значение средней нагрузки (load average)?

Аппаратная часть

```
$ lspci
$ dmidecode
$ ethtool
```

Обычные, невиртуализированные сервера продолжают широко использоваться, и эти команды должны помочь:

• Определить RAID-контроллер (есть ли у него батарея резервного питания?), процессор и количество доступных слотов памяти. Это может подсказать вам потенциальные причины проблемы и пути увеличения производительности.

• Выяснить правильно ли настроена сетевая карта? Не работает ли она в режиме полудуплекса? На скорости 10MBps? Есть ли ошибки приема-передачи?

Производительность ввода-вывода

```
$ iostat -kx 2
$ vmstat 2 10
$ mpstat 2 10
$ dstat --top-io --top-bio
```

Очень полезные команды для анализа общей производительности системы хранения.

• Проверяем свободное место: есть ли в системе полностью занятые файловые системы или диски?

• Используется ли своп (si/so)?

• Что занимает процессор: системные вызовы? пользовательские процессы? много ли времени крадется гипервизором (VM)?

• Моя любимая команда – dstat. Какие процессы интенсивно используют ввод-вывод? Может быть MySQL грузит дисковую подсистему? Или это какой-то PHP-скрипт?

Точки монтирования и файловые системы

```
$ mount
$ cat /etc/fstab
$ vgs
$ pvs
$ lvs
$ df -h
$ lsof +D / /* будьте осторожны,
не положите сервер */
```

• Сколько файловых систем смонтировано?

• Есть ли файловые системы, выделенные для конкретных сервисов? (MySQL например?)

• Какие указаны опции монтирования: noatime? default? Есть ли какие-то файловые системы смонтированные в режиме только для чтения?

• Есть ли свободное место на дисках?

• Нет ли больших удаленных файлов, которые продолжают удерживаться каким-либо процессом?

• Есть ли место для расширения раздела, если проблема в свободном пространстве?

Ядро, прерывания и сеть

```
$ sysctl -a | grep ...
$ cat /proc/interrupts
$ cat /proc/net/ip_conntrack /*
может занять некоторое время на
загруженных серверах */
$ netstat
$ ss -s
```

• Распределены ли прерывания равномерно по всем процессорам? Возможно одно из ядер перегружено из-за прерываний от сетевой карты, RAID-контроллера, …?

• Какое задано значение swappinness в системе? 60 подходит для персональных компьютеров, но не для серверов. Желательно, чтобы сервер никогда не использовал своп, иначе во время чтения/записи данных на диск, процессы вытесненные в своп окажутся заблокированными.

• Достаточно ли велико значение conntrack_max для существующего трафика?

• Как долго TCP-соединения могут находится в различных состояниях (TIME_WAIT, …)?

• netstat может быть немного медленным при выводе всех существующих соединений, тогда используйте ss -s, чтобы быстро получить краткую статистику.

Системные журналы и сообщения ядра

```
$ dmesg
$ less /var/log/messages
$ less /var/log/secure
$ less /var/log/auth
```

• Ищите любые сообщение об ошибках или предупреждения. Есть ли сообщения о слишком большом количестве соединений в conntrack?

• Есть ли сообщения об аппаратных ошибках или ошибках файловой системы?

• Коррелируется ли время между ошибками в журналах и предоставленной информацией о проблеме?

Задания cron

```
$ ls /etc/cron* + cat
$ for user in $(cat /etc/passwd
| cut -f1 -d:); do crontab -l -u
$user; done
```

• Есть ли задания, которые выполняются слишком часто?

• Есть ли персональные конфигурационные файлы cron, спрятанные от постороннего взгляда?

• Выполнялось ли какое-либо резервное копирование в то время, когда возникла проблема?

Журнальные файл приложений

Здесь можно много что исследовать, но вряд ли у вас будет время, чтобы детально все просмотреть. Поэтому, сконцентрируйтесь на самом очевидном, например для LAMP-сервера:

• **Apache & Nginx**; посмотрите журналы доступа и ошибок, ищите ошибки 5xx, возможные ошибки limit_zone.

• **MySQL**; посмотрите есть ли ошибки в mysql.log, следы поврежденных таблиц, работающий процесс восстановления innodb. Посмотрите журнал медленных операций и определите есть ли проблемы с диском, индексами или запросами.

• **PHP-FPM**; если включен журнал php-slow, покопайтесь в нем и попробуйте найти ошибки (php, mysql, memcache, …). Если журнал выключен, активируйте его.

• **Varnish**; проверьте отношение hit/miss в varnishlog и varnishstat. Не пропущено ли правило в конфигурации, в результате чего запросы конечных пользователей проходят до бекэнда, минуя varnish?

• **HA-Proxy**; какой статус у бекэндов? Правильно ли работает проверка здоровья бекэндов? Не переполнена ли очередь запросов на фронтэнде или бекэндах?

Заключение

После этих первых пяти минут (плюс-минус десять), у вас должно будет сформироваться более полное понимание ситуации:

• Что запущено.

• Связана ли проблема с вводом-выводом/аппаратной частью/сетевой подсистемой или конфигурацией (плохой код, настройки ядра, …).

• Есть ли знакомые шаблоны: плохое использование индексов БД, слишком много процессов apache, и т.п.

Вы даже могли уже найти непосредственную причину проблемы. Если нет, то вы находитесь в хорошей позиции для дальнейших поисков, зная, что все очевидное уже проверено.

По материалам сайта:
linux.profiua.com

Создаем SOHO-сервер на базе Zentyal

Часть 1

В первом материале мы рассмотрим установку и базовую настройку маленького сервера на базе какого-нибудь старого компьютера. Сервер под управлением Linux будет служить для сетевых (и не только) нужд небольшого офиса или дома

Вступление

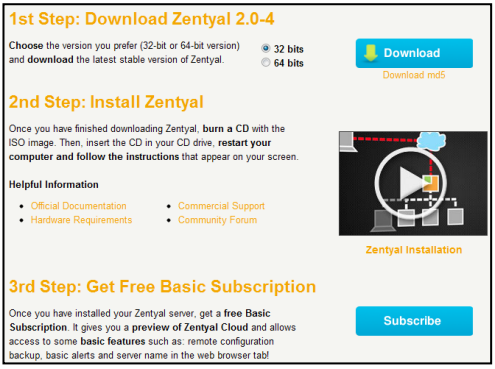
Расширять возможности роутеров за счёт альтернативных прошивок, конечно, интересно. Однако порой наступает такой момент, что добавление очередной надстройки приводит к тому, что и так уже «взмыленный» роутер перестаёт стабильно работать. В этом случае либо придётся отказаться от части функций, либо приготовиться раскошелиться на покупку более мощной модели роутера, а то и вовсе готового решения в виде небольшого сервера с предустановленным софтом. Но зачем? Ведь достаточно просто взять старый компьютер и самостоятельно настроить всё что нужно. Этим-то мы и займёмся. Можно, конечно, взять в руки напильник и превратить паровоз в истребитель, то есть установить какой-нибудь дистрибутив Linux (попутно обязательно пересобрав ядро, куда же без этого), а затем долго и муторно доводить его до нужного состояния, прикрутив под конец Webmin или что-нибудь в этому духе.

Мы же не будем мудрствовать лукаво и воспользуемся специализированным дистрибутивом Zentyal. Он имеет два важных для нас преимущества. Во-первых, у него есть унифицированный веб-интерфейс для управления всем основным модулями сервера (маршрутизация, брандмауэр, DHCP, почта и так далее). Во-вторых, он основан на Ubuntu, а значит, нам доступна вся база пакетов этого дистрибутива. Вообще-то можно установить все компоненты Zentyal на Ubuntu из специального PPA-репозитория. Есть и другой, очень похожий продукт – ClearOS. Оба дистрибутива обладают различными вариантами подписки, но нам вполне хватит и бесплатной версии. При желании и за относительно небольшие деньги можно будет получить чуть больше возможностей, что актуально скорее для организаций, нежели для дома.

Подготовка

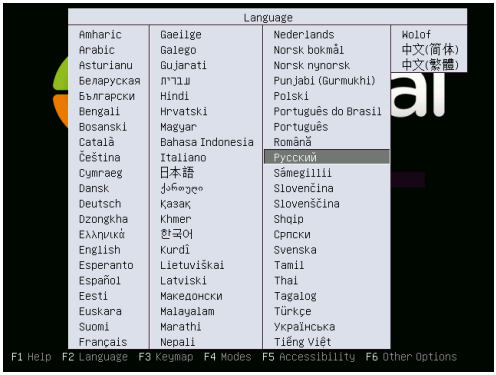
Рекомендуемая для Zentyal конфигурация ПК, который будет играть роль сервера, примерно такова: процессор уровня Pentium 4, от одного гигабайта RAM, винчестер на 80 Гбайт и минимум два сетевых интерфейса (мы же будем делать шлюз). В реальности всё зависит от ваших задач. Сетевая составляющая потребляет меньше всего ресурсов, так что вполне

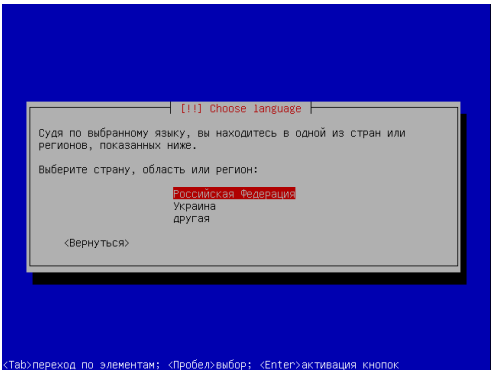
можно обойтись какой-нибудь «атомной» машинкой. Если планируется поставить антивирус, почту, фильтр и так далее, то тут, пожалуй, надо взять что-нибудь посерьёзнее. Логично, что надо купить адаптер Wi-Fi, если нужна беспроводная сеть, но в качестве альтернативы можно приобрести точку доступа (мост) – в некоторых случаях это даже лучше, так как сервер наверняка будет спрятан в каком-нибудь тихом углу, то есть физически удалён от места скопления беспроводных клиентов. На памяти экономить не стоит – она и так нынче совсем недорога. При желании можно организовать RAID, но особого смысла в этом, кажется, нет. Встроенные или софтовые решения не так надёжны, а аппаратный контроллер, пожалуй, будет излишней тратой денег в нашем случае. И ещё, разумнее всего будет выделить отдельный жёсткий диск под данные («файлопомойку», закачки в torrent и так далее) либо вообще добавить USB-накопитель. Подключать его лучше уже после установки ОС.

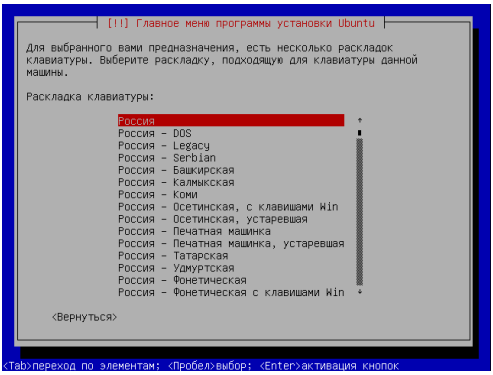


Установка Zentyal

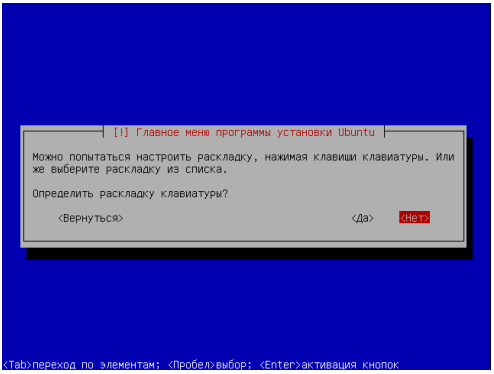
Когда машина будет подготовлена, потребуется скачать нужный ISO-образ установщика с этой странички. Прожигаем ISO на болванку или пишем на флешку. Попутно можно зарегистрироваться в Zentyal и получить базовую подписку на дополнительные сервисы, нажав на кнопку Subscribe на той же странице. Включаем в BIOS загрузку со съёмного накопителя или CD-привода, вставляем наш носитель с образом системы и перезагружаемся. Если хотите, можете выбрать русский язык в ходе установки. В меню выбираем первый пункт (delete all disk) и жмём Enter.

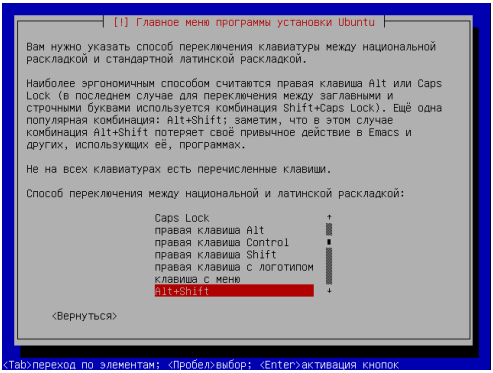


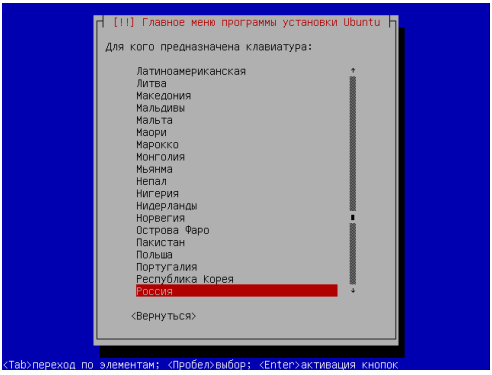




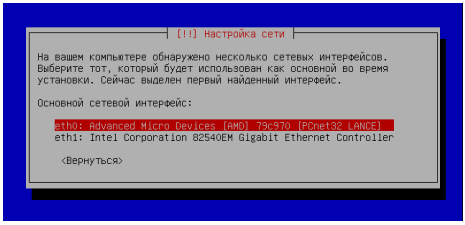
Мастер установки проведёт нас по всем основным пунктам. Первым делом необходимо настроить клавиатуру.

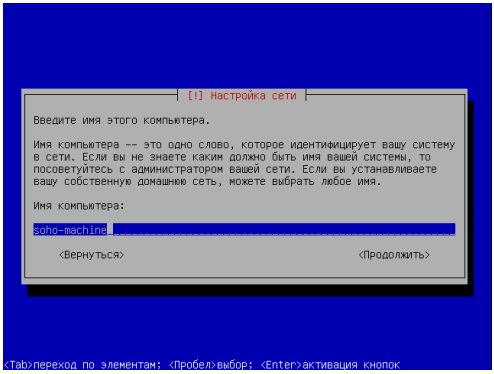




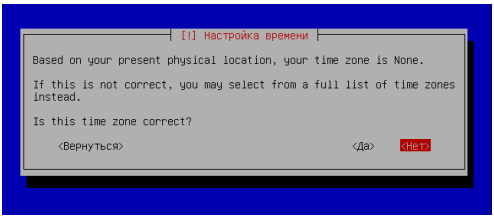


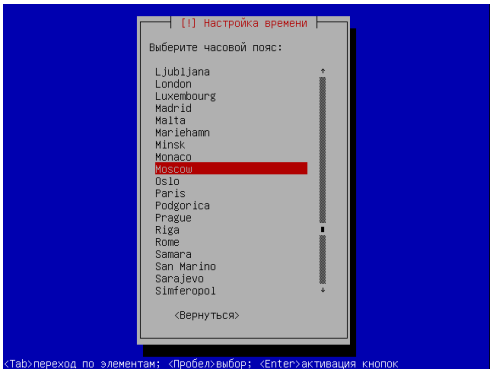
Один из сетевых интерфейсов будет смотреть во внешнюю сеть, а другой – в локальную. По большому счёту нет никакой разницы, какому интерфейсу какую роль вы отведёте. В нашем примере eth0 будет служить для локального подключения, а eth1 для выхода в Интернет.



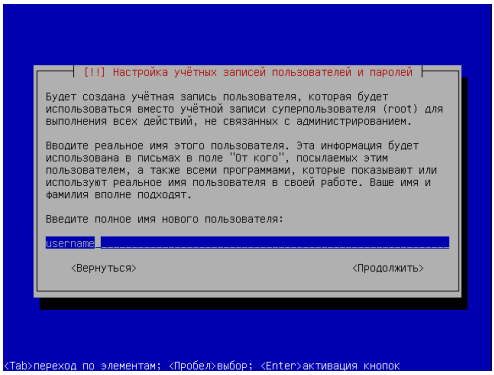


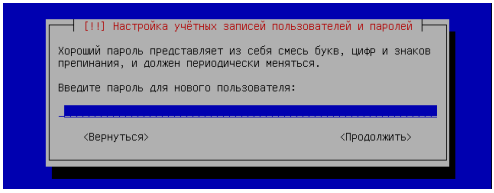
Если установщику не удалось определить ваш текущий часовой пояс, то ему надо немного помочь.



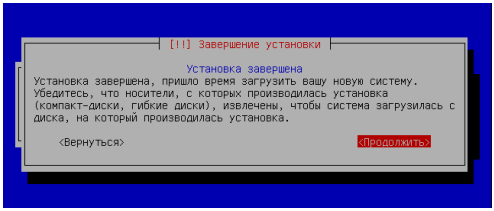


Затем инсталлятор самостоятельно разобьёт диск, отформатирует его и установит базовую систему. Под конец будет предложено создать новую учётную запись администратора.





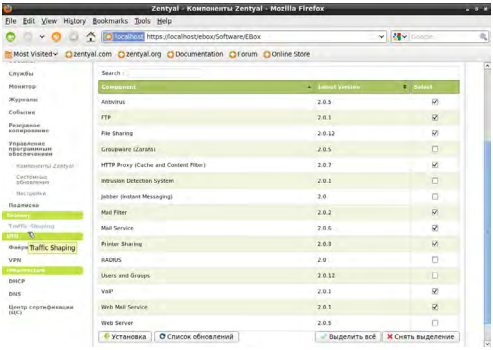
После этого будут установлены оставшиеся компоненты ОС, и нам предложат перезагрузиться. Заодно в BIOS вернём загрузку с жёсткого диска.

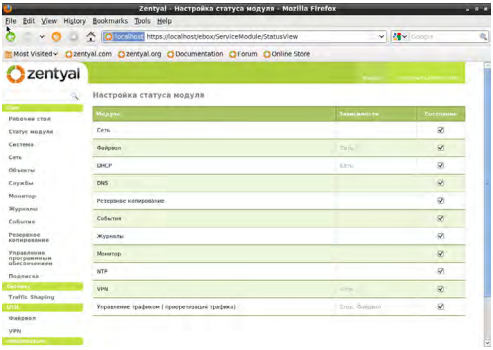


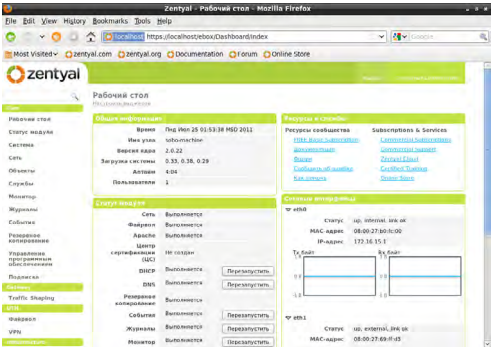
Базовая настройка

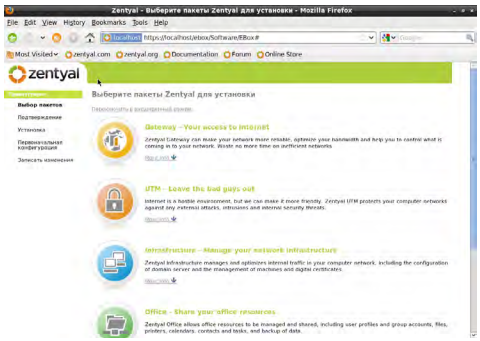
Управление Zentyal осуществляется через веб-интерфейс, который похож на интерфейс большинства роутеров. Из локальной сети он доступен по адресу https://ip_сервера/. После загрузки нам

предлагают зайти в него с помощью логина и пароля администратора, которые были заданы на этапе установки. Мы можем определить серверу одну из стандартных ролей (нам нужен Gateway) либо пропустить настройку и самостоятельно выбрать все нужные модули. Делается это в разделе «Управление программным обеспечением» –> «Компоненты Zentyal». При установке появляются рекомендации доустановить некоторые другие компоненты, которые изначально недоступны. Например, при инсталляции антивируса и SAMBA (для обмена файлами по сети) рекомендуется включить опцию сканирования расшаренных папок на наличие зловредов. Уже установленные модули включаются и отключатся в разделе «Статус модуля». Обратите внимание, что некоторые службы зависят друг от друга – пока не включишь одну из них, другая будет недоступна. Быстрый доступ к информации о текущем состоянии системы и запуску (перезапуску) основных служб доступен с главной страницы веб-интерфейса, она же – «Рабочий стол». В правом верхнем углу находится кнопка «Сохранить изменения», не забывайте нажимать на неё после смены настроек.





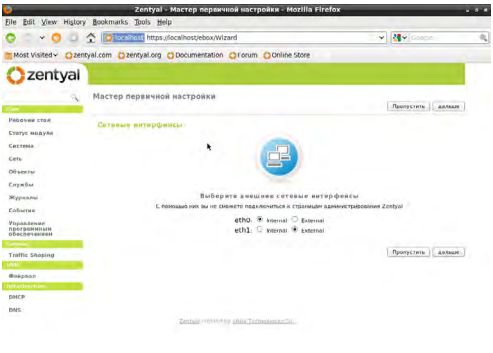


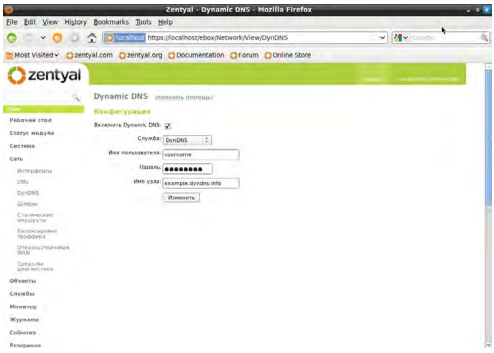


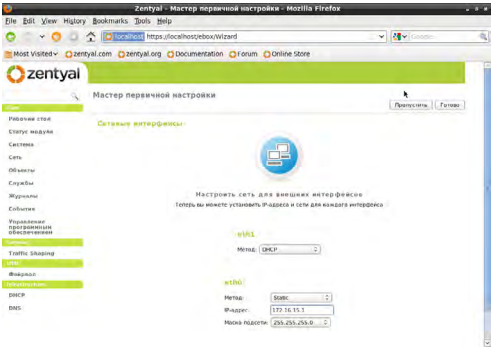
При установке части модулей будет запущен мастер настройки. Например, для настройки сетевых подключений. Для внешнего интерфейса доступны варианты ручного указания всех настроек или получения их по DHCP либо через VLAN

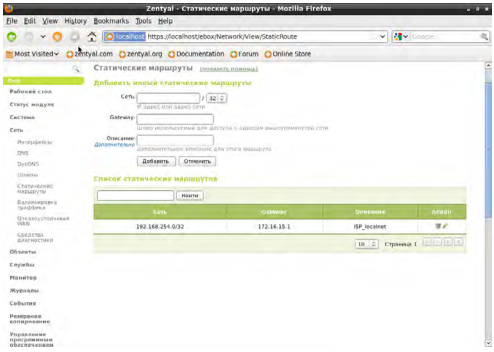
(802.1q) или ADSL (PPPOE). Увы, в данный момент готовой поддержки столь любимых нашими провайдерами PPTP/L2TP нет, и её внедрение не планируется вплоть до следующего релиза, который увидит свет осенью. Наиболее простым выходом из этой ситуации видится покупка простейшего роутера (от 500 рублей), настройка его на соединение с провайдером, прописывание статического IP для сервера и вынос оного в DMZ или полный проброс портов к нему. Для внутреннего интерфейса сервера надо указать статический IP-адрес и выбрать маску подсети. Потом настройки можно будет поменять в разделе «Сеть» –> «Интерфейсы».

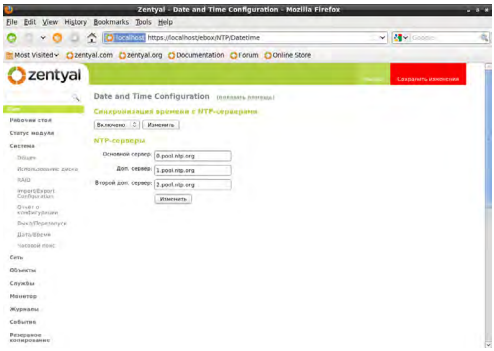
Также нам понадобятся модули NTP, DNS, DDNS и DHCP. Первые три необязательны, а вот без последнего не обойтись, если не хотите вручную прописывать сетевые настройки на всех машинах в локальной сети. В любом случае локальный кеширующий DNS-сервер, внешний домен и локальный же сервер времени полезны. Не забудьте только включить синхронизацию со сторонними NTP-серверами в разделе «Система» –> «Дата/Время». Заодно можно прописать статические маршруты, например для доступа к ресурсам локальной сети провайдера.

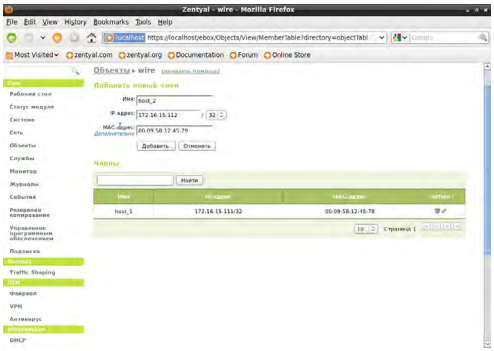




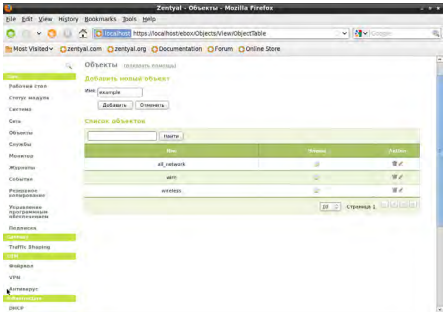




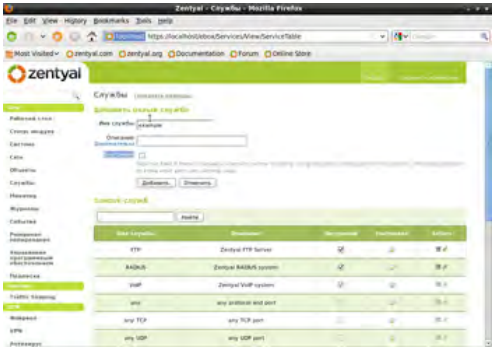


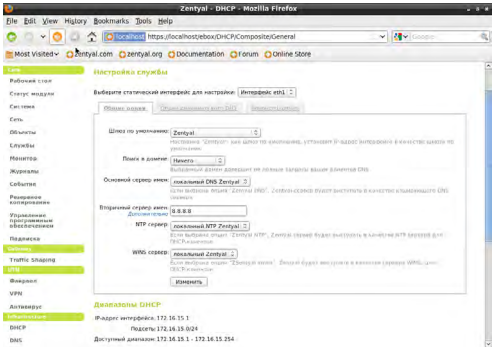


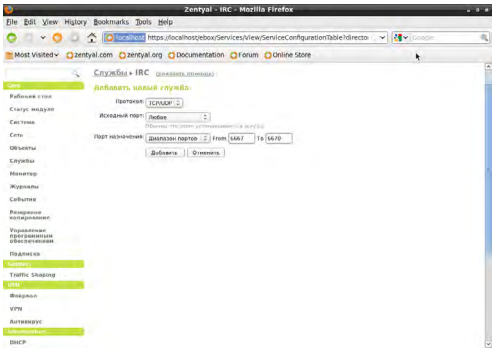
Теперь познакомимся с понятием объектов и служб в Zentyal. Объекты – это любые устройства в сети или их группы (ПК, принтеры, NAS и так далее). Изначально создаются списки объектов (группы), в которые потом добавляются нужные IP-адреса или диапазоны адресов. Для отдельно взятого хоста можно указать и MAC-адрес.

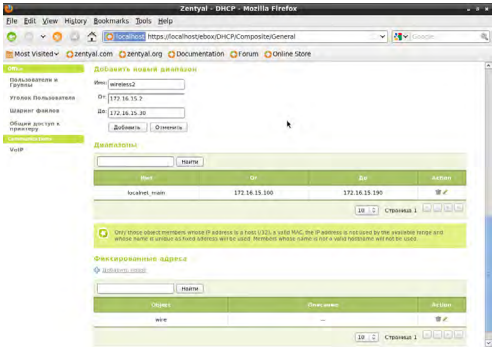


Службы в понимании Zentyal – это порты или группы портов и протоколов. При создании службы можно поставить галочку «Внутренний», если данный порт и протокол используются на сервере (например, порт 21 для FTP-сервера Zentyal). Аналогично объектам каждая служба может включать целый список портов/протоколов. Службы и объекты в дальнейшем можно использовать в других модулях вроде брандмауэра, и нужны они лишь для более гибкой и простой настройки сети.

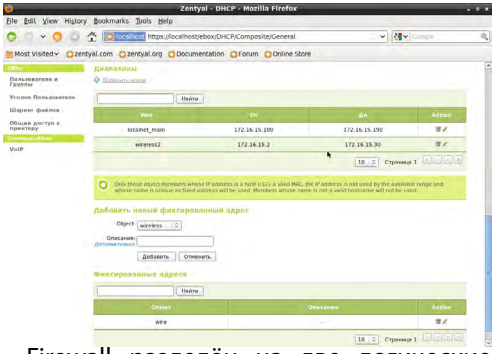




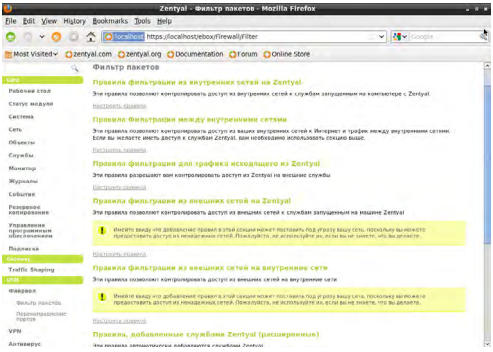


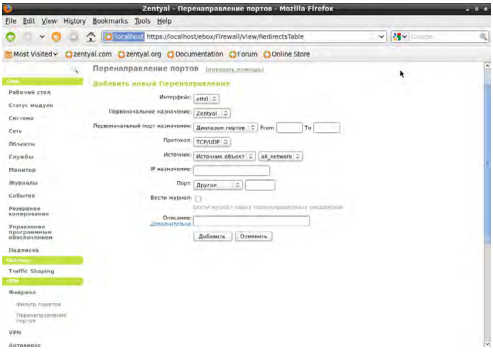


В общем случае для активации DHCP достаточно выставить такие же настройки, как на первом скриншоте ниже. После этого обязательно надо добавить диапазоны IP-адресов, которые будут раздаваться машинам, – их можно создать сразу несколько для разных групп устройств. Статический DHCP реализуется с помощью объектов. Чуть выше в нашем примере мы создали список объектов wire, в котором указали несколько машин с IP- и MAC-адресами. Так вот, нам достаточно добавить любой список объектов в разделе «Фиксированные адреса», чтобы компьютерам из этого списка присваивались заранее указанные IP-адреса в соответствии с их MAC-адресами.

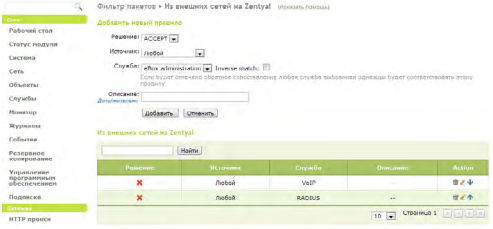


Firewall разделён на две логические части. Первая, фильтр пакетов, не столь интересна, так как позволяет настроить поведение только внутренних служб Zentyal. Вторая часть – самое обычное перенаправление (проброс) портов.

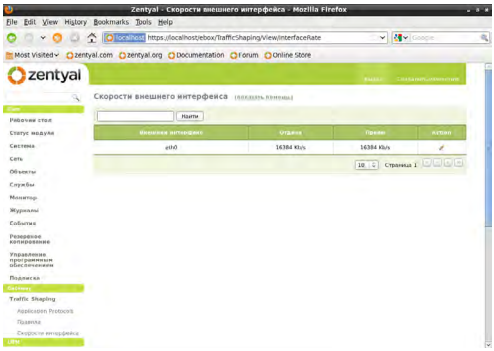


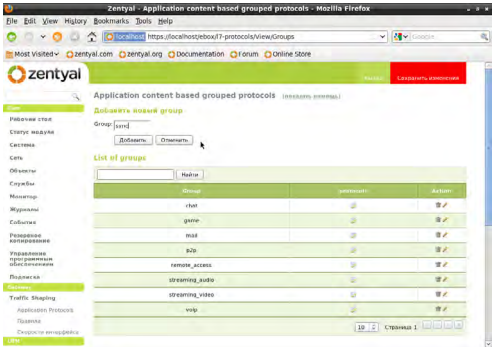


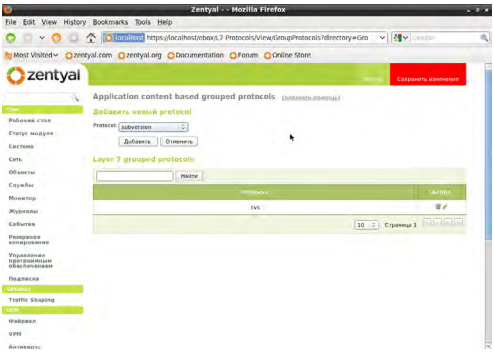
В качестве примера откроем доступ к веб-интерфейсу Zentyal извне, добавив одно правило в «Правила фильтрации из внешних сетей на Zentyal».

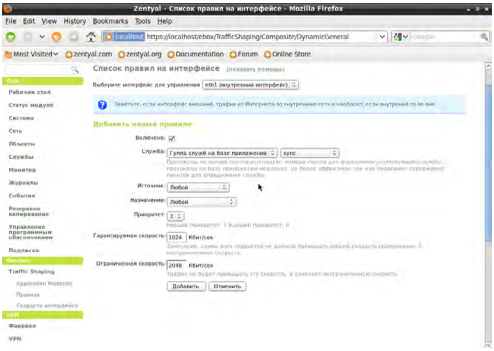


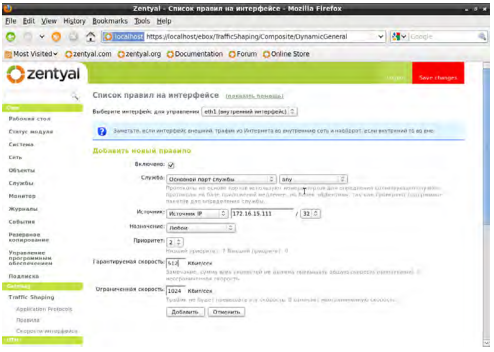
Распределение полосы пропускания настраивается в Gateway –> Traffic Shaping. Естественно, этот модуль должен быть уже установлен. Первым делом в разделе «Скорости интерфейса» надо указать максимальные входящую и исходящую скорости согласно вашему тарифу. Контроль за скоростью базируется на системе фильтров L7. В разделе Application Protocols мы можем создавать и редактировать группы протоколов. Затем необходимо добавить нужные правила для каждого из интерфейсов, выставив приоритет и задав показатели скоростей. Можно, в частности, задать ограничение для каждого из компьютера в локальной сети. Про особенности настройки QoS уже рассказывалось в этой статье – рекомендуется прочитать нужный раздел.





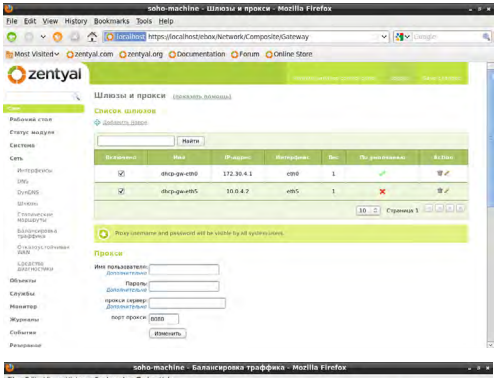


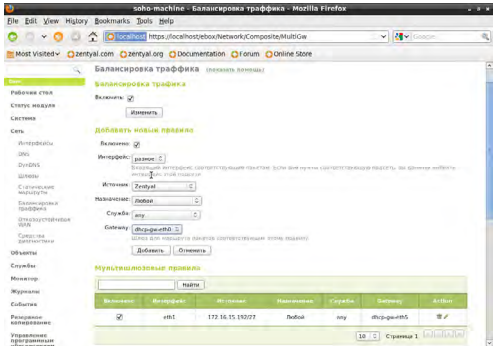




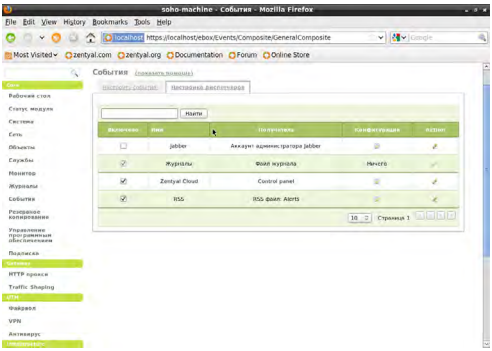
Если у вас есть несколько внешних каналов для выхода в Интернет (например, два шлюза или два ADSL-модема, необязательно с одинаковой скоростью), то можно настроить балансировку трафика.

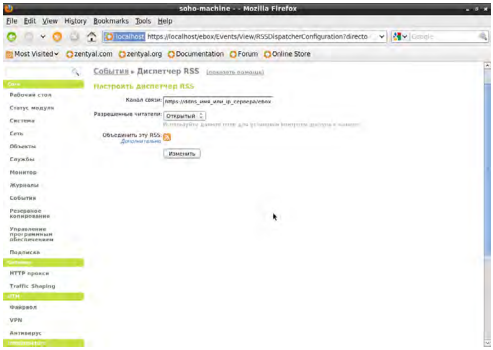
В разделе «Сеть» –> «Шлюзы» как раз и прописываются эти каналы, а для PPPOE и DHCP они создаются автоматически. У каждого внешнего соединения можно указать вес, то есть фактически приоритет выбора того или иного канала. Если скорости у внешних каналов одинаковые, то веса тоже должны быть одинаковые. В противном случае чем выше номер приоритета (больше 1), а значит и ниже скорость, тем реже будет идти обращение к нему. Непосредственно балансировка основывается на правилах, в которых можно указать, через какой шлюз и какие данные будут идти. Здесь нам в очередной раз пригодятся объекты и службы.

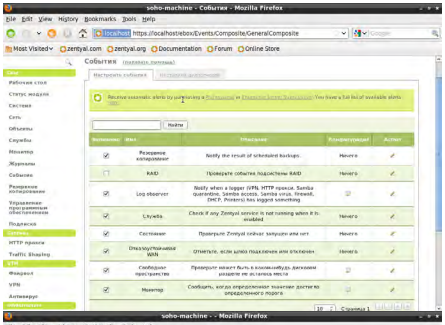


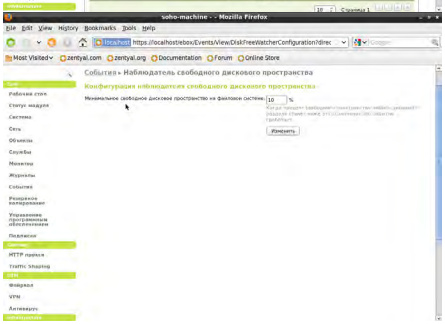


Наличие сразу нескольких шлюзов даёт ещё одно преимущество – возможность автоматического переключения между ними в случае, если один из них перестаёт работать. Но предварительно немного познакомимся с системой событий Zentyal. Нас интересует событие «Отказоустойчивая WAN», которое надо включить. У некоторых событий есть настраиваемые параметры, например можно указать в процентах объём свободного места на жёстком диске, при достижении которого будет сгенерировано оповещение. Оповещение о событии может быть доставлено до администратора несколькими способами – для нас актуальны только RSS или сообщение в Jabber. Одновременно все логи событий записываются в журналы, которые потом можно просмотреть в соответствующем разделе.

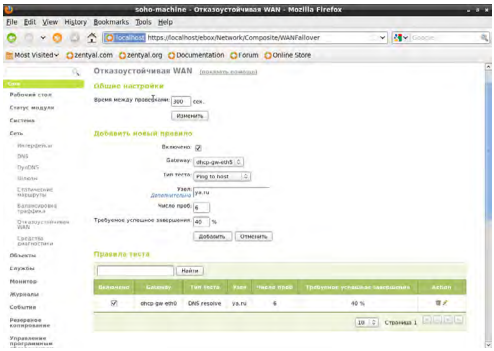


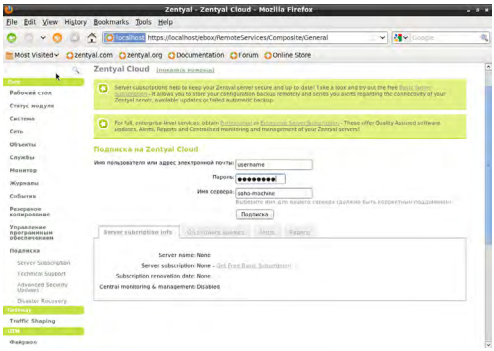






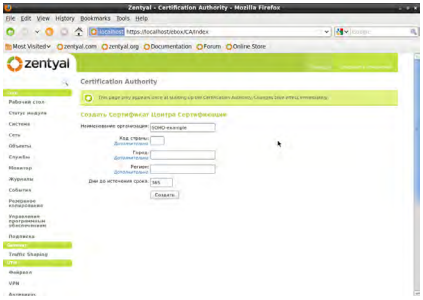
Так вот, после включения отказоустойчивой WAN идём в разделе «Сеть» в пункт, который называется точно так же. Здесь мы добавляем правила проверки работоспособности каждого шлюза путём «пинга» непосредственно шлюза, какого-либо хоста, HTTP-запроса или запроса DNS. Тут же добавляем интервал запуска проверки и выставляем число попыток. Если шлюз не проходит проверку, то он временно отключается до восстановления его работоспособности, а все запросы автоматически перенаправляются на другой(ие) шлюз(ы).



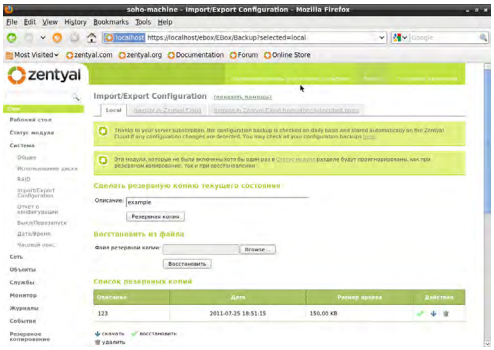


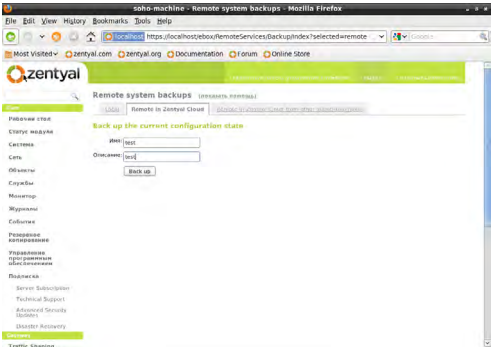
Дополнительные настройки

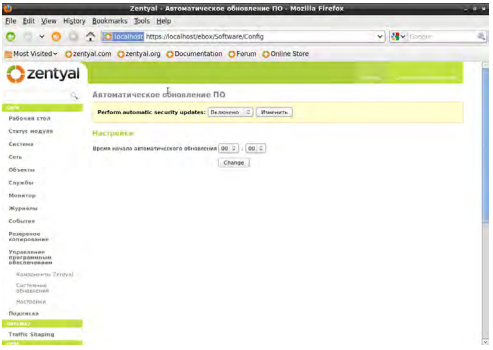
Если вы решили оформить базовую подписку на сервисы Zentyal, то вам на почту должны были прийти логин и пароль. Перед тем как подключить её, надо сгенерировать сертификаты (цифровые ключи) в «Центре сертификации». Они же в дальнейшем понадобятся нам для создания VPN-подключений к серверу. Для корневого сертификата достаточно указать название организации и срок его действия. После чего в разделе «Подписка» –> Server Subscription достаточно ввести присланные логин и пароль. Особенного смысла в этом, если честно, нет – можно только посмотреть в деморежиме на возможности, доступные в платных вариантах подписки (резервное копирование, управление группой серверов, удалённое обновление и так далее).



В разделе «Система» –> Import/Export Configuration есть возможность сохранить и восстановить текущие настройки сервера. Файл с настройками рекомендуется скачать и сохранить на другой машине или съёмном накопителе. Также можно сохранить конфигурацию в сервисе Zentyal. Это, пожалуй, единственная польза от него, кроме возможности посмотреть, в Сети ли сейчас сервер, и автоматического уведомления по почте, если он вдруг отключился.







Наконец, последнее, что рекомендуется сделать при первичной настройке, – обновить систему из раздела «Системные обновления», нажав на «Список обновлений», отметив галочками нужные пакеты и потом нажав «Обновление». Маленький совет – лучше не выбирать все пакеты скопом, а обновлять их небольшими пачками. Альтернативный вариант – просто выполнить в консоли (User Console) две команды:

```
sudo apt-get update && sudo apt-get upgrade
```

Ну и включить напоследок автоматическое обновление ПО в настройках.

На этом, пожалуй, прервёмся. В следующей части мы разберём создание групп и пользователей, настройку файлообменника, установку торрент-клиента и ряд других вещей.

По материалам
3dnews.ru

Трансляция изображения с IP-камер

Иногда возникает задача организации трансляции видеопотока с веб-камеры в сеть. Технических вариантов решения данной задачи существует множество. Рассмотрим некоторые из них.

Начальные условия:

Две IP-камеры hikvision 2CD7133E, которые отдают видеопоток по RTSP протоколу, закодированый h264, mpeg, mjpeg;

Сервер под управлением Ubuntu.

В качестве сервера для трансляции потока используем Wowza Stream Server. Продукт платный, хотя есть и демо-версия. Поскольку он написан на java, то java-машина уже должна быть установлена. Устанавливаем сервер:

```
sudo dpkg -i ~/WowzaMediaServer-2.2.4.deb
```

Создаем каталог приложения:

```
sudo mkdir -p /usr/local/WowzaMediaServer/conf/live
```

Копируем настройки

```
cp -v /usr/local/WowzaMediaServer/conf/Application.xml /usr/local/WowzaMediaServer/conf/live/
```

Редактируем скопированный файл:

Назначьте свойству Streams/StreamType значение: live

Назначьте свойству HTTPStreamers значение: cupertinostreaming, smoothstreaming, sanjosestreaming

Назначьте свойству Streams/LiveStreamPacketizers значение: cupertinostreamingpacketizer, smoothstreamingpacketizer, sanjosestreamingpacketizer

Назначьте RTP/Authentication/PlayMethod значение: none

Добавьте следующее свойство в контейнер MediaCaster/Properties:

forceInterleaved

true

Boolean

Используя текстовый редактор, создайте файл /usr/local/WowzaMediaServer/content/camera.stream

Введите содержимое в виде полного пути RTSP/RTP URL к Вашей камере, в зависимости от количества камер создаем соответственно camera1.stream, camera2.stream и т. д.

Используя текстовый редактор, отредактируйте файл /usr/local/WowzaMediaServer/conf/admin.password и добавьте username (имя пользователя) и password (пароль), которые будут использоваться для пуска и останова публикации видео от камеры.

Запутите Wowza Media Server 2

Откройте браузер и введите в адресной строке: http://ip.ad.re.ss:8086/streammanager и введите Ваши admin username и password, которые Вы назначили в файле admin.password

Кликните по ссылке [start-receiving-stream] сразу под разделом live application folder

Выберите MediaCaster Type: rtp

Введите camera.stream в поле Stream Name

Кликните OK

Теперь у вас запущена трансляция потока.

Переходим к размещению на сайте.

Создаём в DocumentRoot сайта директорию flowplayer и скачиваем туда flowplayer. После этого создаём файлы start.html, index.html, index1.html, flash.html, flash1.html, html5.html, html51.html.

Листинг start.html:

```
<table width="640" height="240"
border="0" scrolling="no"
align="center" cellspacing="0"
cellpadding="0">
<tr height="240" valign="top">
<td width="320" height="240">
<iframe style="border: 0px;
padding:0px; width:320px;
height:240px !important;
position:absolute; z-index:1;
margin-top:0px; margin-right:0px;
border:0px;" scrolling="no"
align="middle" src="index.html"
></iframe>
</td>
<td width="320" height="240">
<iframe style="border: 0px;
padding:0px; width:320px;
height:240px !important;
position:absolute; margin-
top:0px; margin-left:-
9px !important; border:0px;
z-index:0" scrolling="no"
align="middle" src="index1.html"
></iframe>
</td>
</tr>
</table>
```

Листинг index.html:

```
<html><head>
<meta http-equiv="content-
type" content="text/html;
charset=UTF-8">
<script type="text/javascript"
src="flowplayer-3.2.6.min.js"></
script>
</head><body>

<div id="page">
<!--<a href="null" style="displa
y:block;width:320px;height:240px"
id="player"></a>-->
<script type="text/javascript">
var br = navigator.userAgent.
toLowerCase().indexOf("safari");
var br1 = navigator.userAgent.
toLowerCase().indexOf("chrome");
if (br != -1 && br1 == -1)
{
location='html5.html';
}
else
{
location='flash.html';
}
</script>
</div>
</body></html>
```

Листинг flash.html:

```
<html><head>
<meta http-equiv="content-
type" content="text/html;
charset=UTF-8">
<script type="text/javascript"
src="flowplayer-3.2.6.min.js"></
script>
</head><body>
```

```
<div id="page">
<script>
var output = '<a href="null" ';
output += 'style="display:block;w
idth:320px;height:240px" ';
output += 'id="player"></a>';
document.write(output);
flowplayer("player", "flowplayer-
3.2.7.swf",
{
clip: {
url: 'camera.stream',
live: true,
provider: 'rtmp'
},
plugins: {
rtmp: {
url: '../flowplayer.rtmp-
3.2.3.swf',
netConnectionUrl: 'rtmp://ip.ad.
re.ss:1935/live'
}
}
});
</script>
</div>
</body></html>
```

Листинг html5.html:

```
<html><head>
<meta http-equiv="content-
type" content="text/html;
charset=UTF-8">
</head><body>
<div id="page">
<script>
var output = '<video
controls="controls" wigth="515"
height="386">';
output += '<source src="http://
ip.ad.re.ss:1935/live/camera.
stream/playlist.m3u8">';
output += '</video>';
document.write(output);
</script>
</div>
</body></html>
```

Файлы index1.html, flash1.html, html51.html отличаются от index.html, flash.html, html5.html только именем потока: в оригинальных файлах это camera.stream, а в файлах с единичкой - camera1.stream.

Основная идея тут в том, чтобы показывать рядом картинку с двух камер и при этом в зависимости от браузера для показа видео использовать технологию flash или html5.

При таком решении нагрузка на процессор составляла всего 7%.

В какой-то момент заказчик изъявил желание просматривать видео с камер на iPad. К сожалению Safari вообще не поддерживает flash, а в случае html5 не позволяет одновременно показывать сразу два потока.

В такой ситуации выбор пал на motion, который хотя и является детектором движения, умеет просто ретранслировать поток. Хотя пришлось перенастроить камеры чтобы они отдавали картинку в Jpeg.

Устанавливаем motion:

```
apt-get install motion
```

Переходим к конфигурированию. Открываем файл /etc/motion/motion.conf и приводим к виду:

```
daemon on
quiet on
width 640
height 480
framerate 10
quality 20
auto_brightness off
threshold 4500
noise_level 64
brightness 0
contrast 0
```

```
saturation 0
hue 0
ffmpeg_cap_new off
target_dir /var/lib/motion/
snapshots
webcam_localhost off
# Можно увеличить если нужно по-
высить качество картинки
webcam_quality 20
webcam_maxrate 10
output_all off
output_motion off
output_normal off
thread /etc/motion/thread2.conf
thread /etc/motion/thread1.conf
```

Создаём файл /etc/motion/thread1.conf с указанием пути к html потоку с камеры и порту, на котором будет вестись трансляция:

```
netcam_url http://ip.ad.
re.ss:port/streaming/channels/1/
preview
webcam_port 10000
```

Для второй камеры отличие только в адресе и номере порта:

```
netcam_url http://ip.ad.
re.ss:port/streaming/channels/1/
preview
webcam_port 10001
```

Создаём файлы index.html, 251.html, 252,html:

Листинг index.html:

```
<table width="1030" height="386"
border="0" scrolling="no"
align="center" cellspacing="0"
cellpadding="0">
<tr height="386" valign="top">

<td width="515" height="386">
<iframe style="border: 0px;
padding:0px; width:515px;
height:386px !important;
position:absolute; z-index:1;
margin-top:0px; margin-right:0px;
border:0px;" scrolling="no"
align="middle" src="252.html" ></
iframe>
</td>
<td width="515" height="386">
<iframe style="border: 0px;
padding:0px; width:515px;
height:386px !important;
position:absolute; margin-
top:0px; margin-left:-
9px !important; border:0px;
z-index:0" scrolling="no"
align="middle" src="251.html" ></
iframe></td>
</tr>
</table>
```

Листинг 251.html:

```
<script language="javascript">
var BaseURL = "http://ip.ad.
re.ss:10001/";
var File = "stream.mjpg";
theDate = new Date();
output = '<IMG SRC="';
output += BaseURL;
output += File;
output += '" WIDTH="515">';
setInterval(document.
write(output),80);
</script>
```

Файл 252.html отличается от 251.html только номером порта.

Поскольку поток отдаётся в формате mjpeg то далеко не все браузеры смогут его показать. Но firefox вполне может.

По материалам
ylsoftware.com

Магазин "TOTAL"

- персональные компьютеры;
- компьютерные комплектующие;
- ноутбуки, нетбуки, планшеты;
- принтеры, МФУ, расходники;
- сетевое оборудование;
- CD/DVD диски, флеш-накопители;
- и многое другое.

г. Кривой Рог, ул. Адмирала Головко, 40, Терновской р-н
тел. (067)-698-87-79, (097)-692-73-38

Изменяем разрешение экрана

На днях установил своему хорошему знакомому Linux Mint. И все бы хорошо, но вот максимальное разрешение экрана, на его 19-дюймовом мониторе, было 1024 x 768 вместо 1440 x 900. Пришлось немного побороться с данной проблемой.

Решений я нашел много, но остановился на двух самых распространенных. В одном советовали изменить системный файл Xorg.conf, а в другом – воспользоваться командой xrandr. Первый случай отпал почти сразу, потому как файла Xorg.conf в системе не оказалось, а вот создать его я так и не смог. О позор мне! Поэтому я опишу тот способ, который сработал на все 100%.

Итак в бой! Выполним в Терминале команду:

```
xrandr
```

После этого Вы должны увидеть вывод команды, в которой должны быть указаны поддерживаемые разрешения. В моем случае нужного разрешения я не увидел, и скорее всего, что Вы тоже не увидите.

А вот собственно и сам вывод:

```
Screen 0: minimum 8 x 8, current
1280 x 1024, maximum 16384 x
16384
DVI-I-0 disconnected (normal left
inverted right x axis y axis)
VGA-0 connected 1280×1024+0+0
(normal left inverted right x
axis y axis) 338mm x 270mm
1280×1024 60.0*+ 75.0
1280×960 60.0
1152×864 75.0
1024×768 75.0 70.1 60.0
800×600 75.0 72.2 60.3 56.2
640×480 75.0 72.8 59.9
DVI-I-1 disconnected (normal left
inverted right x axis y axis)
HDMI-0 disconnected (normal left
inverted right x axis y axis)
```

Из этого вывода команды понятно, что мой монитор подключен к VGA выходу видеокарты (VGA-0), а незадействованные выходы оказались такие: DVI-I-1 и HDMI-0.

Поскольку монитор 19-дюймовый, то мне нужно было разрешение 1440 x 900. Точное разрешение экрана я узнал в интернете по модели монитора. Далее выполним команду:

```
cvt 1440 900 60
```

Я получил следующий вывод:

```
# 1440x900 59.89 Hz (CVT 1.30MA)
hsync: 55.93 kHz; pclk: 106.50
MHz
Modeline «1440x900_60.00  106.50
1440 1528 1672 1904 900 903 909
934 -hsync +vsync
```

Из верхнего вывода команды, копируем текст, который выделен красным цветом и выполняем следующую команду, которая создаст новый режим с нужным нам разрешением:

```
xrandr --newmode "1440x900_60.00"
106.50 1440 1528 1672 1904 900
903 909 934 -hsync +vsync
```

Теперь добавим его в систему:

```
xrandr --addmode VGA1
1440x900_60.00
```

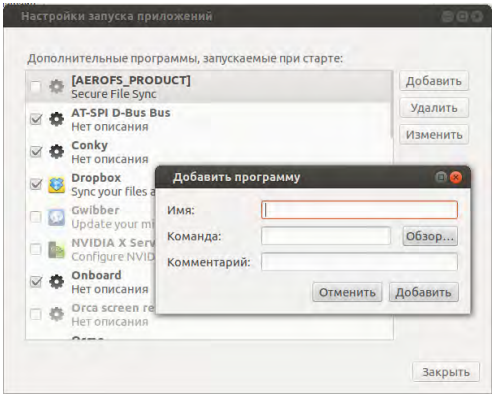
Запустим новый режим:

```
xrandr --output VGA1
--mode 1440x900_60.00
```

По идее у Вас должно поменяться разрешение экрана на нужное, у меня по крайней мере было именно так. Теперь, если все вышло как надо, нужно сделать так, чтобы данное разрешение экрана загружалось вместе с загрузкой системы. Потому, что если этого не сделать, то измененное (нужное Вам) разрешение экрана сбросится сразу же после перезагрузки системы. Для этого создадим скрипт. Откроем текстовый редактор Gedit, либо любой другой, и впишем следующий текст:

```
#!/bin/sh
xrandr --newmode "1440x900_60.00"
106.50 1440 1528 1672 1904 900
903 909 934 -hsync +vsync
xrandr --addmode VGA1
1440x900_60.00
xrandr --output VGA1 --mode
1440x900_60.00
```

и сохраняем, например с названием Display.sh в Домашнюю папку. Главное, чтобы после названия скрипта, обязательно было окончание .sh. Теперь нужно сделать этот скрипт исполняемым. Для этого правой кнопкой мыши по этому файлу –> Свойства –> Права –> (ставим галочку) –> Разрешить выполнение файла как программы. Теперь добавим этот скрипт в Автозагрузку системы. Все, должно работать!



По материалам
softhelp.org.ua

Устанавливаем драйвера принтеров Canon

В данной статье я расскажу как установить драйвера для принтеров Canon, марок PIXUS, MX, MP, IP и MG в Ubuntu 14.04/13.10/13.04/12.10/12.04, а также Linux Mint 16/15/14/13.

Если модель вашего принтера Canon есть в списке, который представлен ниже, то вы без проблем сможете его установить.

Canon PIXUS 550	Canon MP140	Canon iP1900
Canon PIXUS 560	Canon MP160	Canon iP2200
Canon PIXUS 850	Canon MP190	Canon iP2500
Canon PIXUS 860	Canon MP210	Canon iP2600
Canon PIXUS 865	Canon MP240	Canon iP2700
Canon PIXUS 950	Canon MP490	Canon iP3300
Canon PIXUS 990	Canon MP500	Canon iP3500
	Canon MP510	Canon iP3600
	Canon MP520	Canon iP4200
Canon MX320	Canon MP540	Canon iP4500
Canon MX330	Canon MP550	Canon iP4700
Canon MX350	Canon MP560	Canon iP4800
Canon MX360	Canon MP600	Canon iP5200
Canon MX410	Canon MP610	Canon iP6600
Canon MX420	Canon MP630	Canon iP7500
Canon MX420	Canon iP100	Canon MG5100
Canon MX860	Canon iP1800	Canon MG5200
Canon MX870	Canon iP1000	Canon MG6100
Canon MX880	Canon iP1500	Canon MG8100

Для этого нужно добавить нужный репозиторий и обновить список пакетов системы командами в Терминале:

```
sudo add-apt-repository
ppa:michael-gruz/canon-trunk
sudo apt-get update
```

Ну а теперь нужно вписать именно вашу модель принтера в команду для установки драйвера. Команда будет иметь следующий вид:

```
sudo apt-get install cnijfilter-
ip100series
```

или так:

```
sudo apt-get install cnijfilter-
mp140series
```

либо вот так:

```
sudo apt-get install cnijfilter-
mx320series
```

Надеюсь у вас все получиться и ваш принтер заработает как часы. Удачи!

По материалам
softhelp.org.ua

Подробнее о командах архивирования и сжатия в системе Linux

Потребность упаковывать и сжимать файлы в виде одного архива возникла примерно в то же самое время, когда компьютеры впервые получили жесткие диски, и эта потребность не исчезла до сегодняшнего дня. Каждый день в виде архивов загружается на сайты и скачивается с сайтов все, начиная от документов и фотографий и заканчивая программами и драйверами устройств. Большинство пользователей знакомы с файлами .zip, однако есть гораздо больше архиваторов, которые не так известны, как .zip. В этом руководстве мы расскажем вам о различные командах, которые есть в Linux, и о том, как правильно их использовать.

Tar и gzip

Исторически сложилось так, что инструментом архивирования, используемым в Linux по умолчанию, является команда tar. Первоначально она означала «Tape Archive» («Архив на магнитной ленте»), но это было тогда, когда ленты были основным носителем для переноса данных. Команда tar является очень гибкой и она может создавать, сжимать, обновлять, распаковывать и тестировать архивные файлы. По умолчанию расширением для несжатого архива tar (иногда называемым файлом tar или архивом tarball) является расширение .tar, тогда как для сжатых архивов tar чаще всего используют расширение .tgz (означающее, что архив tar сжат с помощью команды GNU zip). В действительности в архивах tar предлагается несколько различных методов сжатия, в том числе bzip2, zip, LZW и LZMA.

Чтобы создать несжатый архив tarball всех файлов, имеющихся в каталоге, используйте следующую команду:

```
tar cvf somefiles.tar *
```

`c` означает create (создать), `v` расшифровывается как verbose (означающее, что команда tar выведет список файлов, которые она архивирует), а `f` говорит о том, что следующий параметр является именем файла архива, в данном случае somefiles.tar. Символ `*` точно также, как и в большинстве команд командной оболочки Linux, обозначает все файлы в каталоге.

Архив somefiles.tar создается в текущем каталоге. Теперь его можно сжать с помощью таких инструментальных средств как gzip, zip, compress или bzip2.

Например:

```
gzip somefiles.tar
```

Комада gzip сожмет архив и добавит расширение .gz. Теперь в текущем каталоге вместо файла somefiles.tar будет находиться файл somefiles.tar.gz.

Этот двухшаговый процесс, когда создается архив и он сжимается, может быть сведен к одному шагу с помощью операции tar, встроенной в команду сжатия:

```
tar cvzf somefiles.tgz *
```

Эта команда создаст сжатый архив gzip (архив tarball), который называется somefiles.tgz. Дополнительный параметр z указывает команде tar, что сжатие будет осуществляться в tarball. Вместо параметра z можно использовать параметры j, J или Z, которые соответственно указывают команде tar использовать алгоритмы сжатия bzip2, xz и LZW. С помощью параметра xz указывается использования алгоритма сжатия LZMA2, который представляет собой точно такой же алгоритм, как и популярная в системе Windows программа 7-Zip.

7-Zip

С помощью команды 7zr можно создать файл, совместимый с алгоритмом сжатия 7-Zip. Чтобы создать архив .7z, используйте следующую команду:

```
7zr a somefiles.7z *
```

Параметр a означает add (добавить), т. е. добавить все локальные файлы в архив somefiles.7z. Затем этот файл можно отправить пользователю системы Windows и он сможет извлечь содержимое архива без каких-либо проблем.

В системе Linux не рекомендуется использовать команду 7zr для резервного копирования, поскольку в таких архивах 7-zip не сохраняется информация о владельце/группе файла. Команду 7zr можно использовать для создания архива tarball (в котором сохраняется информация о владельце файла). Вы можете сделать это с помощью конвейера Unix следующим образом:

```
tar cvf - * | 7zr a -si somefiles.
tar.7z
```

Символ дефиса после параметра f сообщает команде tar о том, что результат нужно посылать в стандартный выходной поток Unix – в stdout, а не в файл. Поскольку как мы используем конвейер, выходной поток команды tar будет направлен в команду 7zr, которая будет ожидать ввода данных из стандартного входного потока, поскольку указан параметр -si. Затем команда 7zr создаст архив с именем somefiles.tar, в котором будет находиться архив tarball, в котором, в свою очередь, будут находиться файлы. Если вы не знакомы с использованием конвейера, то вы можете в два этапа можете создать стандартный архив tarball, а затем сжать его с помощью команды 7zr:

```
tar -cvf somefiles.tar *
7zr a somefiles.tar.7z somefiles.
tar
```

Извлечение данных

Извлечение файлов из этих различных архивов также осуществляет достаточно просто; ниже приведена краткая шпаргалка для извлечения файлов из различных архивов, созданных выше.

Чтобы извлечь простой архив tarball, используйте команду:

```
tar xvf somefiles.tar
```

Где параметр x означает eXtract (извлечение).

Чтобы извлечь сжатый архив tarball, используйте команду:

```
tar xvzf somefiles.tgz
```

Параметр *z* указывает команде tar о том, что для сжатия исходного архива была использована команда gzip.

Вместо параметра z вы можете в зависимости от того, какой применялся алгоритм сжатия при создании архива, указать параметр j, J или Z.

Чтобы извлечь файлы из файла 7-Zip, используйте команду:

```
7zr e somefiles.tar.7z
```

Заключение

В системе Linux предлагается широкий спектр команд архивирования и сжатия. Попробуйте поэкспериментировать с командами zip и xz; они работают очень похоже на другие команды, уже упомянутые здесь. Если у вас возникнут проблемы, то за дополнительной подсказкой вам следует обратиться к описанию на страницах man, например, man xz.

По материалам
rus-linux.net

Терминал Linux. Команды навигации в терминале

Часть 1

Что из себя представляет корневая файловая система Linux?

Прежде чем приступить к командам в терминале, поговорим о директориях в Linux. Здесь нет диска C:\\ D:\\ и прочих.

Когда я начинал свой путь в Linux, для меня это было наиболее непонятно. Поэтому хочу прояснить этот момент.

Не важно на сколько вы разобьете разделов ваш жесткий диск. В системе будет:

"/" - корневая директория

Затем уже идут все остальные каталоги, которые служат для чего-то.

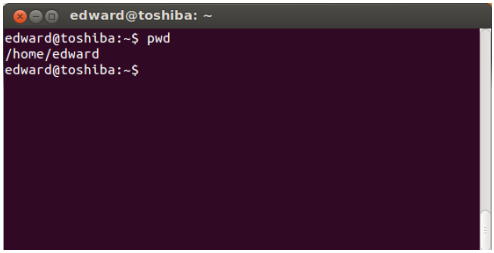
Директория /home - хранит каталоги пользователей, в которых можно, как правило, выполнять большинство действий без пароля администратора. Ваш домашний каталог находится по адресу - /home/Ваше_Имя_пользователя.

Директории /mnt и /media - вот это важно, в эти каталоги монтируются другие физические диски, флешки и прочие носители информации. В Ubuntu в каталог /media монтируются диски, флешки, то есть в этом каталоге создается папка с названием вашего электронного носителя.

Остальные каталоги системные и рассказу о них стоит выделить целую статью. Перейдем к командам в терминале.

Команды навигации в терминале

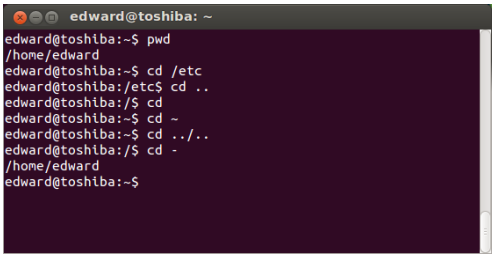
Когда вы открываете терминал в Ubuntu. То вы находитесь в своем домашнем каталоге:



Значок ~ (тильда) - означает домашний каталог текущего пользователя. Чтобы узнать текущую директорию, в терминале достаточно набрать команду: pwd. Выполнение данной команды представлено на скриншоте выше.

Для навигации в терминале используется команда "cd". Сначала записывается команда cd, затем путь, куда нужна перейти. Выглядит так:

Данной командой переходим в каталог /home cd /home. Но это не всё, что можно делать с помощью команды cd:



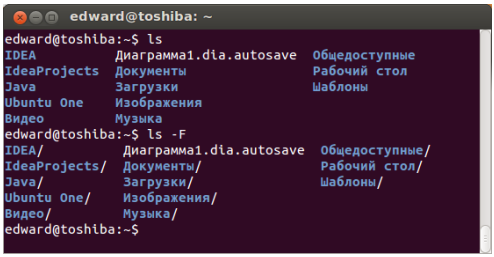
cd .. перейти в директорию уровнем выше

cd ../.. перейти в директорию двумя уровнями выше

cd ~ перейти в домашнюю директорию

cd - перейти в директорию, в которой находились до перехода в текущую директорию

Следующая команда используемая в навигации "ls". Если ввести ls, то отобразится содержимое текущей директории:



ls -F отобразить содержимое текущей директории с добавлением к именам символов, храктеризующих тип

ls -l показать детализированое представление файлов и директорий в текущей директории

ls -a показать скрытые файлы и директории в текущей директории

Важно! Когда вы вводите команду cd, вам не обязательно вводить полностью имя директории, достаточно нажать клавишу "TAB", произойдет автодополнение названия каталога. Если имя не заполнилось, значит, есть еще один каталог с таким названием. Нажав дважды клавишу "TAB", выведется список с каталогами, которые имеют в своем имени начальные буквы, которые вы ввели.

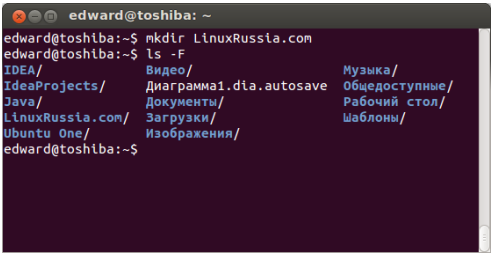
Это касается не только "cd", но и других команд. Например, когда Вы хотите отредактировать какой-то файл. Вы выполниликоманду ls, увидели, что у файла имя

состоит из 30 знаков, то достаточно ввести первые буквы его имени и нажать клавишу "TAB" и произойдет автодополнение.

Команды удаления, копирования в терминале

Итак с навигацией разобрались, теперь научимся удалять и копировать каталоги и файлы в терминале

Создание каталогов выполняется командой "mkdir":

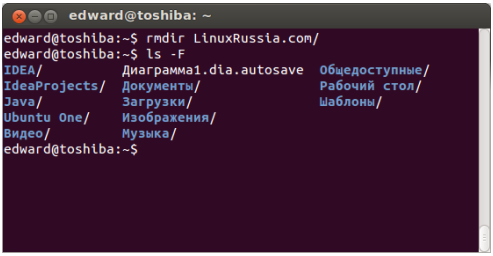


mkdir dir1 создать директорию с именем 'dir1'

mkdir dir1 dir2 создать две директории одновременно

mkdir -p /dir1/dir2 создать дерево директорий

Для удаления директорий используется команда "rmdir":



Для удаления файлов используется команда "rm". Работает очень просто, пишете команду и название файла:

rm file1 удалить файл с именем 'file1'

Но "rm" позволяет не только удалять файлы, но и каталоги:

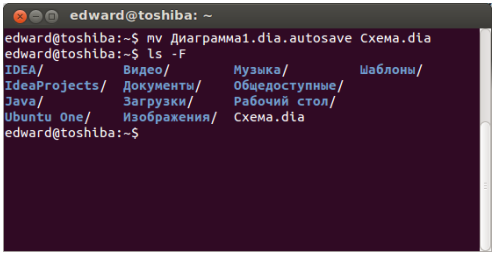
rm -r dir1 удалить директорию с именем 'dir1' и рекурсивно всё её содержимое

rm -r dir1 dir2 удалить две директории и рекурсивно их содержимое

Так же используется параметр -f, это означает, что при удалении не будет запрашиваться подверждение на удаление, команда будет выглядеть так:

```
rm -f file1
rm -rf dir1
```

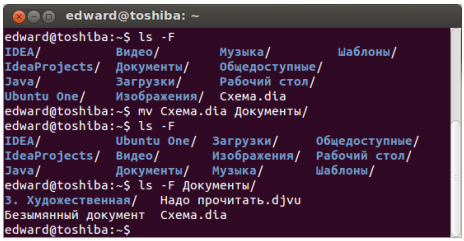
Для перемещения и переименовывания файлов и каталогов используется команда mv:



Всё просто, чтобы переименовать файл, записываем команду:

mv название_файла новое_название

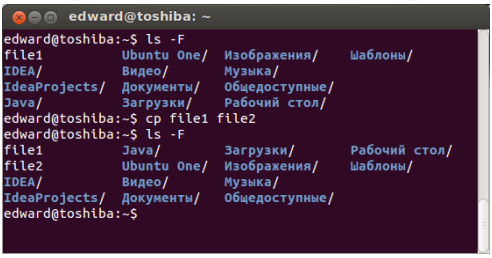
Чтобы переместить файл:



mv название_файла путь/

Ну и осталось копирование. Для копирования используется команда "cp":

cp file1 file2 скопировать файл file1 и назвать его file2



cp -a dir1/ dir2/ - копировать директорию dir1 в директорию dir2

cp file1 dir1/ - копировать файл с именем file1 в директорию с именем dir1

Вот основные команды навигации в терминале Linux.

Важно! У каждой команды присутствует множество параметров, чтобы узнать их, просто наберите команду и параметр --help

Например: rm --help

По материалам
linuxrussia.com

Находка для Штирлица
или урок криптографии

Иногда нужно скрыть данные от чужих глаз. Например, при передаче тайного электронного письма нужно гарантировать, чтобы его мог прочитать только получатель и больше никто другой. Пароли также вернее хранить зашифрованными, а ключ прятать отдельно! Для этого существуют криптографические программные средства, но давайте напишем что-то свое.

Чтобы расшифровать данные было невозможно, даже если известен алгоритм, при шифровании используют пароль (ключ). Ключом может служить слово, фраза, набор символов, текстовый файл. Надежный ключ должен быть длинным и состоять из знаков, лишённых смысла.

Для шифрования байтов данных будем использовать логическую побитовую операцию xor – "исключающее или". Схема этой операции такая:

0 ^ 0 = 0; 0 ^ 1 = 1; 1 ^ 0 = 1; 1 ^ 1 = 0

Операцию xor в языке C/C++ выполняет оператор ^. С мы и используем для написания программы.

Для шифрования операция xor удобная тем что она обратимая. Если есть уравнение A = B ^ C, где переменные A, B, C – это байты, то также верно уравнение B = A ^ C.

Алгоритмы шифрования бывают разные. Будем использовать простой метод. Его суть в том, чтобы поочередно обрабатывать каждый байт входного файла и байт ключа. Над этими байтами выполняется операция ^ и байт результата записывается в выходной зашифрованный файл. После обработки последнего байта ключа, в алгоритм шифрования поступает первый байт ключа, потом второй и так по кругу. Например, файл с текстом «планета земля» будет зашифрован ключом zyx по схеме:

```
планета земля
^^^^^^^^^^^^^
zyxzyxzyxzyxz
```

Как и самая xor, такой алгоритм обратим: его выполнение над зашифрованными байтами при ТОМ ЖЕ ключе даст расшифровку.

О формате зашифрованного файла

Зашифрованные файлы будут иметь особенные расширения - .xef. Первый байт зашифрованного файла указывает на длину оригинального расширения. Это нужно для того чтобы можно было определить где находятся байты оригинального расширения, а где зашифрованные данные.

Следующие байты – это оригинальное расширение файла. После них записаны зашифрованные данные.

А теперь к работе. Создаем текстовый файл xor_encoding.c и пишем туда следующую программу:

```
#include <stdio.h> // для printf, getchar и файловых функций
#include <string.h> // для strcmpi, strcpy, strcat, strlen, strrchr
#include <time.h> // для clock
// В компиляторах под Линукс может не быть функции strcmpi.
// Если strcmpi не определена, определяем ее как strcmp:
#ifndef strcmpi
#define strcmpi strcmp
#endif
// Разница strcmpi от strcmp в том, что strcmpi
// сравнивает строки без учета регистра.

// Размер буфера для чтения, шифрования, и записи в выходной файл.
#define BUFER_SIZE 32768 // 32 кбайт

// Пароль для шифрования/дешифрования:
const char password[] = "vqx2ef9kjr34p56-fw6t5u2fr5tej%f142z";

int main(int argc, char* argv[]) {
        // Декларируем переменные:
        unsigned char bufer[BUFER_SIZE]; FILE *f_in, *f_out;
        unsigned int i; int pc, last_pc = -1, is_encoding;
        unsigned long processed_bytes = 0, file_size; size_t n;
        char szOutFileName[512], szOriginalExt[256];
        const char *p = &password[0]; char *pDot; unsigned char orig_ext_len;
        clock_t start_time, end_time;
// p – указатель на текущий символ пароля.

        if (argc < 2) { // если не задан входной файл
                printf("Input file was not specified.\n");
                return 1; // выход из программы
        }

        // argv[1] – это первый аргумент программы: имя входного файлу.
// 2-й аргумент функции fopen – это режим открытия: r – на чтение;
```

```
// w – на запись; b – двоичный режим.
        // Открываем входной файл на чтение:
        f_in = fopen(argv[1], "rb");
        if (f_in == NULL) { // ошибка открытия
                printf("Cannot open input file \"%s\".\n",
                    argv[1]);
                return 2; // выход из программы
        }

        // В file_size получаем размер файла:
        fseek(f_in, 0, SEEK_END); // идем на конец файла
        file_size = ftell(f_in); // ftel вернет текущую позицию
        fseek(f_in, 0, SEEK_SET); // идем на начало

        // Выводим размер входного файла:
        printf("Input file size: %lu bytes.\n", file_size);
        // Выводим имя входного файла:
        printf("Input file: \"%s\".\n", argv[1]);
        // В szOutFileName копируем имя входного файла:
        strcpy(szOutFileName, argv[1]);

        // Поиск точки справа-налево в szOutFileName:
        pDot = strrchr(szOutFileName, '.');
        // Если точка есть и после точки стоить строка xef:
        if ( pDot && (strcmpi(pDot, ".xef") == 0) ) {
                is_encoding = 0; // файл будем расшифровывать
                *pDot = '\0'; // расширения .xef у выходном файле не будет

  // С первого байта читаем длину оригинального расширения:
                orig_ext_len = fgetc(f_in);
                if (orig_ext_len) { // если есть  оригинальное расширение
                        // Читаем его в szOriginalExt:
                        fgets(szOriginalExt, orig_ext_len + 1, f_in);
                        // Добавляем расширение до имени выходного файла:
                        strcat(szOutFileName, ".");
```

```
                strcat(szOutFileName, szOriginalExt);
        }
        processed_bytes = (1 + orig_ext_len);

    } else { // входной файл обыкновенный, его будем шифровать
        is_encoding = 1;
        if (pDot) { // если есть расширение
            // Запоминаем оригинальное расширение:
            strcpy(szOriginalExt, pDot + 1);
            // С имени файла удаляем оригинальное расширение:
            *pDot = '\0';
        } else // файл без разширения
            szOriginalExt[0] = '\0'; // расширение пустое
// До имени входного файлу добавим .xef - это будет выходной файл:
        strcat(szOutFileName, ".xef");
    }

    // Выводим имя выходного файла:
    printf("Output file: \"%s\".\n", szOutFileName);
    // Сообщаем что происходить: шифрование или расшифровывание:
    printf("\t%s...\n", is_encoding ? "Encoding" : "Decoding");

    // Проверяем: существует ли уже выходной файл:
    // Пробуем открыть выходной файл на чтение:
    f_out = fopen(szOutFileName, "rb");
    if (f_out) { // если файл открыли, значить он существует
        fclose(f_in); // закрываем входной файл
        fclose(f_out); // закрываем выходной файл
        // Выводим предупреждение:
        printf("Output file \"%s\" already exists.\nPress enter to exit.\n",
            szOutFileName);
        getchar(); // делаем паузу до тех пор пока не нажмут enter
        return 3; // выход из программы
    } // конец блока if (f_out)
```

```
        // Открываем выходной файл на запись:
        f_out = fopen(szOutFileName, "wb");
        if (f_out == NULL) { // ошибка открытия
                fclose(f_in); // закрываем входной файл
                printf("Cannot open output file \"%s\" for writing.\n",
                    szOutFileName);
                return 4; // выход из программы
        }

        if (is_encoding) { // если шифрование
  // В 1-й байт файла f_out пишем сколько символов в расширении:
                fputc( (unsigned char)strlen(szOriginalExt), f_out );
                fputs(szOriginalExt, f_out); // пишем расширение
        }

        start_time = clock(); // фиксируем время начала
        for (;;) { // этот цикл for закончится после вызова break
                // n - сколько байтов реально прочитано из файла.
                // Из входного файла f_in пытаемся
                // читать BUFER_SIZE байтов в массив bufer:
                n = fread(bufer, sizeof(char), BUFER_SIZE, f_in);
  // Если байт не прочитан, завершаем цикл for (;;):
                if (n == 0) break;

                // Сканируем каждый байт в массиве bufer i шифруем(или расшифро-
вываем):
                for (i = 0; i < n; i++) {
  // Выполняем операцию ^ над байтом bufer[i] и текущим байтом пароля:
                        bufer[i] = ( bufer[i] ^ (unsigned char)*p );
  // Указатель на текущий символ пароля переводим на следующий символ:
                        p++;
  // Если указатель p вышел на конец пароля, ставим его на начало:
                        if ((*p) == '\0') p = &password[0];
                } // конец тела цикла for (i = 0; i < n; i++)

                // Записываем зашифрованный(или расшифрованный) фрагмент в
```

файл f_out:

```
                fwrite(bufer, sizeof(char), n, f_out);

                processed_bytes += n; // прирост количества обработанных байтов
                // В pc вычисляем текущий процент прогресса:
                pc = (int)
  ( ((double)processed_bytes) / ((double)file_size) * 100.0 );
                if (pc != last_pc) { // если процент изменился
                        // Выводим сколько байтов обработано и процент:
                        printf("Processed bytes: %lu ( %d %% )\r",
                            processed_bytes, pc);
                        last_pc = pc; // запоминаем процент
                } // конец блока if
        } // конец тела цикла for (;;)

        end_time = clock(); // фиксируем время завершения
        // Выводим сколько секунд затрачено на операцию:
        printf("\nDone. Elapsed time: %f seconds\nPress enter to exit.\n",
          ( (float)(end_time - start_time) ) / (float)CLOCKS_PER_SEC);
        fclose(f_in); // закрываем входной файл
        fclose(f_out); // закрываем выходной файл
        getchar();
        return 0;
  } // конец тела функции main
```

Функция clock возвращает сколько времени (специальная величина) прошло после запуска программы. Чтобы получить значение в секундах, нужно эту величину поделить на константу CLOCKS_PER_SEC.

Скомпилируем программу и запустим ее указав имя какого-нибудь файла в качестве первого аргумента.

Насладимся эффектом.

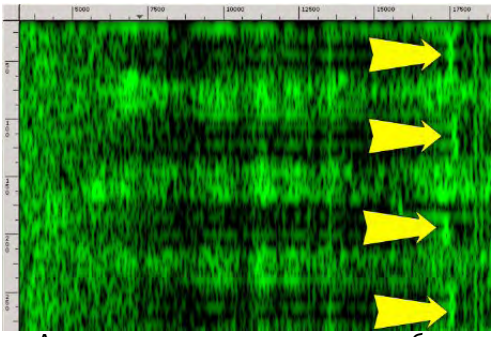
Сергей Вовк.
grayvovk@meta.ua

Техника определения RSA-ключей через анализ изменения шума от компьютера

Представлен новый выпуск пакета GnuPG 1.4.16 (GNU Privacy Guard) в котором добавлена защита от интересной техники атаки (CVE-2013-4576), позволяющей восстановить используемый для шифрования RSA-ключ на основании анализа изменения звукового фона, исходящего от компьютера в процессе расшифровки данных. Ветка GnuPG 2.0 указанной проблеме не подвержена.

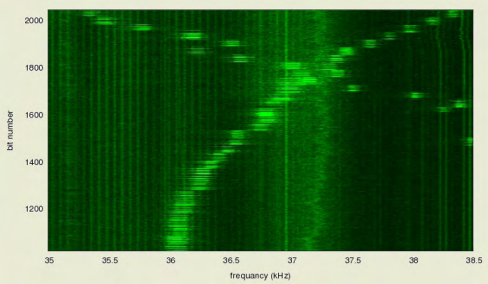
При выполнения различных операций многие компьютеры издают характерный шум, возникающий в результате вибрации некоторых электронных компонентов в блоке регулирования напряжения (конденсаторы и катушки индуктивности), который пытается поддерживать постоянное напряжение в условиях колебаний энергопотребления, возникающих при выполнении различных операций CPU. В итоге, разные RSA-ключи можно отличить по разному характеру звука в процессе их использования.

Звуки в помещении и шум от механических частей компьютера не мешают проведению атаки, так как такой шум в основном лежит в диапазоне ниже 10KHz, в то время как при атаке анализируются высокочастотные звуковые колебания. Выполнение нескольких задач в системе также не влияет на возможность проведения атаки, так как разные задачи могут быть связаны с разными акустическими сигнатурами.

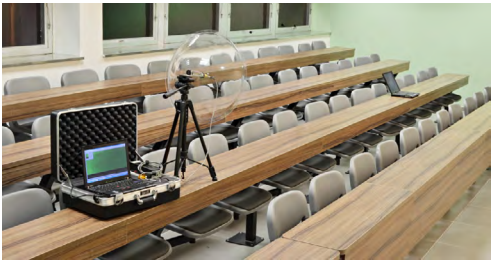


Атака проводится следующим образом. В непосредственной близости (около 30 см) от целевой системы размещается сенсор, в качестве которого может выступать обычный смартфон. Для успешной атаки требуется инициирование автоматической расшифровки произвольных сообщений на прослушиваемой системе, например, должен быть запущен почтовый клиент, автоматически расшифровывающий сообщения. Смартфон периодически отправляет на данную систему немного изменённые сообщения, зашифрованные публично доступным ключом, и контролирует изменение акустических колебаний, исходящих от целевой системы.

На основании изменения характера шума бит за битом реконструируется закрытый ключ (характер входного потока постепенно приводится к звуковой сигнатуре искомого RSA-ключа). На определение 4096-битового RSA-ключа, используемого на ноутбуке, уходит примерно час.



В качестве сенсора может быть использован высокочувствительный направленный микрофон с параболическим отражателем, в этом случае успешная атака была продемонстрирована при удалении от целевой системы на четыре метра. Без параболического отражателя, атаку удалось совершить с расстояния в один метр. Сообщается, что аналогичные атаки могут быть выполнены через измерение электрического потенциала, который также меняется в зависимости от выполняемых вычислений (достаточные для проведения атаки данные можно снять через удалённые концы кабелей VGA, USB или Ethernet или просто прикоснувшись рукой к корпусу).



По материалам
opennet.ru

ED.ua Школьный Электронный Дневник

Школа | Учительская | Профиль | Оплата | Обучение

Социальный проект компании "ВИТ" – Школьный электронный дневник

- Электронная база данных
- Персональный сайт школы
- Новости, события, праздники
- Связь с учителями и родителями
- Домашнее задание, оценки, замечания и поощрения
- Мобильная версия сайта
- Электронная очередь детских садов
- Отчеты, статистика, рейтинг школ

а также:

различные акции, скидки, праздники для наших пользователей!

Функции постоянно добавляются и модернизируются!

с ED.ua сбудется моя мечта!

ПОТОМУ ЧТО НА САЙТЕ ED.UA ЕСТЬ ПОЛНОЕ ДОМАШНЕЕ ЗАДАНИЕ!

ФЕОДОСИЯ
ФЛП Касьянова О. В. :
тел: +380991605920
+380950244989
http://ed.ua

ЛУГАНСК
ФЛП Турецкая З. В. :
тел: +380500311340
+380990631993
http://m.ed.ua

98112, Украина, АР Крым, г.Феодосия, ул. Крымская, 21-А Тел.: (06562) 7-44-79

Выбираем OpenSource решение для организации корпоративных видеоконференций

Различного рода совещания являются неотъемлемой частью процесса управления в компании любого размера. Но одно дело, когда сотрудники находятся в одном здании, другое, когда их приходится собирать из разных городов и даже стран. Именно поэтому сегодня все более популярными становятся видеоконференции, позволяющие повысить эффективность общения и сэкономить на командировках.

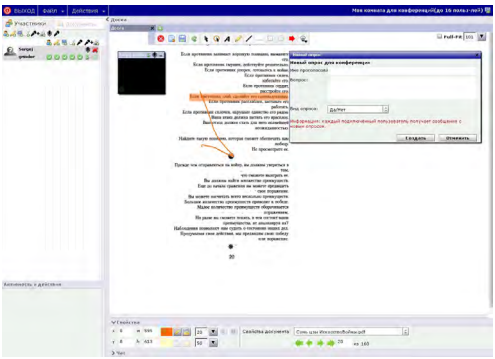
Apache OpenMeetings

В настоящее время существует большое количество решений для организаций видеоконференций, отличающихся по назначению: внутреннее, внешнее использование, вещание, типу: персональные или групповые (комнатные), виду: точка-точка и многоточечные, реализации: аппаратные или программные, используемым протоколам, стоимости и так далее. Рыночная цена коммерческих и особенно аппаратных решений довольно высока, однако применение свободного ПО поможет снизить затраты. Система веб-конференций Apache OpenMeetings позволяет организовать проведение аудио- и видеосовещаний в многоточечном режиме, когда к серверу подключено несколько десятков человек. За несколько лет проект сменил несколько команд и лицензий, в том числе был в Google Code (под лицензией Eclipse Public License). Последняя его дислокация – инкубатор Apache, соответственно, поменялась и лицензия, на Apache License 2.0. Последняя официальная версия 2.0 вышла в конце июля 2012 года. Главный плюс в том, что для видеосовещания не требуется установка дополнительного ПО – достаточно веб-браузера с плагином для поддержки технологии Flash. Предусмотрена возможность записи и последующего проигрывания совещаний и экспорта в AVI/FLV файл, импорт в конференцию документов более чем 20 форматов и изображений. Участники могут скачать загруженный файл и совместно редактировать, вводя текст поверх оригинала, рисовать и помечать интересные места. Сами конференции могут быть открытыми и частными. Поддерживается два режима:

- Совещание – до 16 участников, каждый может передавать аудио- и видеоданные;
- Лекции – до 200 участников, передача аудио и видео только у модератора/лектора.

Предусмотрен также обмен текстовыми сообщениями в окне чата или приватными (используется встроенный Jabber сервис). Настройки позволяют создать

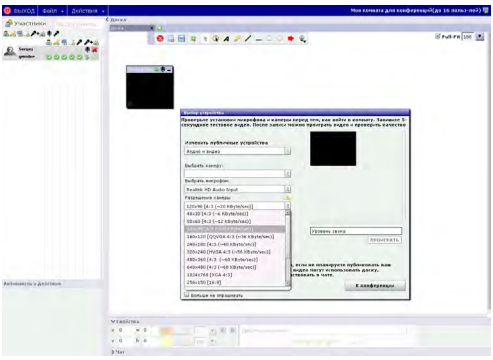
опрос. Модератор, организующий конференцию, отправляет всем участникам приглашение, содержащее прямую ссылку, он же управляет всеми доступными им возможностями. У каждого зарегистрированного пользователя имеется календарь событий с напоминанием о событиях (через электронную почту или iCal). При подключении выбирается вариант участия (видео+аудио, только видео или аудио, рисунки), разрешение и устройство. Настройки в комнате просты и понятны каждому, пользователь, впервые воспользовавшийся сервисом, быстро освоится. Возможна интеграция OpenMeetings с другими продуктами – сервером VoIP Asterisk, системой управления обучения Moodle, Drupal, Joomla, SugarCRM и некоторыми другими.



Совместная работа с документом в Openmeetings

Реализовано три уровня доступа – пользователь, модератор и администратор сервера. Для аутентификации возможно использование внутренней базы или сервиса LDAP/ Active Directory (в $RED5_HOME/webapps/openmeetings/conf можно найти готовые шаблоны для подключения). Возможна работа нескольких серверов OpenMeeting в кластере, одна установка может обслуживать несколько организаций. Интерфейс Openmeetings переведен на несколько языков, среди которых есть русский. Встроенный редактор локализованных сообщений (LanguageEditor) позволяет при необходимости скорректировать перевод. Внешний вид можно изменить при помощи тем. Построен Openmeetings с использованием технологий Java и XML. Для организации сервера задействуются: веб-сервер Apache Tomcat, Open Source Flash/RTMP Server Red5, OpenOffice.org/LibreOffice. В качестве базы данных может быть использована MySQL, PostgreSQL, Oracle, DB2 или Apache Derby. Соединение с сервером осуществляется по протоколам http (порт 5080), rtmp (порт 1935), rtmpt (порт 8088). Встроенный менеджер создания резервных копий упрощает операции по резервированию и восстановлению работоспособности сервера и переносу в другую систему. Компоненты мультиплатформенные, поэтому сервер будет работать на любой *nix системе или Windows. Установку Openmeetings сложной назвать нельзя, процесс просто требует должной внимательности, в последующем эксплуатация особых хлопот не вызывает. Требования к оборудованию невысоки, минимальные, которые указанны на сайте – компьютер с процессором 1 ГГц CPU и 1 Гб ОЗУ. Но в случае конвертирования документов, загрузки файлов и записи ви-

део, этой мощности не хватит. В качестве рекомендуемых указан компьютер 2x/4x 2 ГГц (32/64 бит) и 4 Гб ОЗУ. Для организации 100 соединений достаточно компьютера класса Pentium 4 с 2 Гб ОЗУ.



Выбор устройства при соединении к конференции в Openmeetings

Как рассчитать пропускную способность?

Практически половина всех попыток внедрений систем видеоконференций проваливается из-за неготовности сетевой инфраструктуры. Поэтому еще на этапе выбора поставщика нужно оценить возможности своей сети и требования к пропускной способности. Возможно, они потребуют модернизации для поддержки QoS на уровне, достаточном для проведения видеоконференций. Каждый производитель обычно дает приблизительные расчеты для одного канала. Например, для Apache OpenMeetings каждое подключение к серверу требует 256 кбит/сек. Хотя клиент может выбрать подключение с меньшим качеством, уменьшая требование до 160 кбит/сек. В итоге для сервера нужно обеспечить (N – количество участников):

- входящий канал – (256 x N) кбит/с;
- исходящий канал – ((256 x N x (N – 1)) кбит/с.

Для клиентской системы:

- входящий канал – (256 x (N – 1)) кбит/с;
- исходящий канал – 256 кбит/с.

Отдельный поток в BBB требует 30-50 кбайт/сек. Приблизительные расчеты для BBB можно найти в FAQ – goo.gl/Pii7Y. В том же Skype для видеоконференций рекомендуется более широкий канал – 4 Мбит/сек (прием) 512кбит/сек (передача).

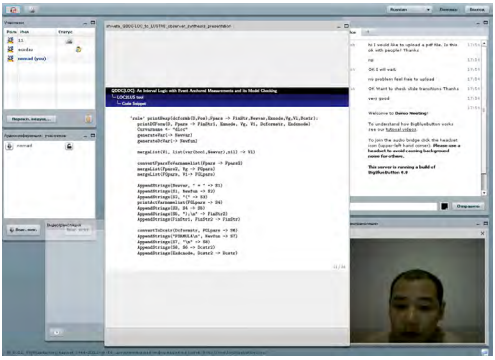
BigBlueButton

Первая версия BigBlueButton (bigbluebutton.org) была написана 2007 году одним из сотрудников Карлтонского университета г. Оттава, Канада (Carleton University), при поддержке программы развития инновационных технологий и управления. Изначально продукт носил имя Blindside, но позже название было изменено на BigBlueButton, чтобы отразить простоту в использовании – для начала конференции нужно всего лишь нажать символическую синюю кнопку. Именно в простоте BBB превосходит более функциональный и оснащенный, а значит и чуть более сложный OpenMeetings. Проект некоторое время искал свое место и сегодня ориентирован на организации, предлагающие услуги дистанционного образования, позволяя проводить обучение через интернет. Особую роль в этом процессе отводится одной из функций – видеоконференции. Но BBB с таким же успехом может быть использован для простого общения, проведе-

ния брифингов и вебинаров. В 2009 году была организована компания Blindside Networks для оказания платной поддержки пользователям продукта. Наиболее серьезный толчок в разработке продукта произошел в 2010 году после участия в Google Summer of Code. Именно тогда был добавлен API, позволяющий подключать сторонние приложения, и сегодня встроить BBB можно в Sakai, WordPress, Moodle, Joomla, Redmine, Drupal, Matterhorn, LAMS и некоторые другие. Эта возможность более всего востребована пользователями BBB, поэтому из настроек сервера был убран интерфейс администратора: разработчики просто не видят смысла его развивать, так как управление ложится на плечи того, кто встраивает приложение.

В случае отдельного сервера все установки можно без проблем произвести при помощи конфигурационных файлов BigBlueButton и возможностей веб-сервера. Проект находится на стадии активной разработки, причем следует отметить особую щепетильность в этом вопросе. Например, выходу версии 0.8 Bailetti предшествовали 4 беты и 3 RC. Недавно проект присоединился к бизнес-инкубатору для открытых проектов WebFWD («Web Forward»), который поддерживает Mozilla.

BigBlueButton обеспечивает многопользовательские аудио- и видеоконференции, чат и обмен личными сообщениями (в качестве клиента поддерживаетсятолько собственный Java-апплет BigBlueButton), запись лекций (слайды, аудио и чат) для дальнейшего воспроизведения (используется HTML 5, поддерживается пока FF и

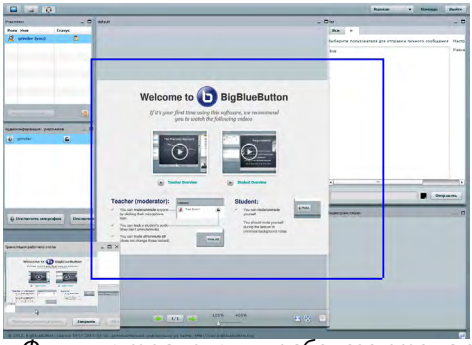


BigBlueButton позволяет организовать конференцию нажатием одной кнопки

Chrome), предоставление общего доступа к рабочему столу для практического показа работы с приложениями и ОС, загрузку презентации в формате PDF (и любом другом, поддерживаемым OpenOffice.org/LibreOffice), функции рисования и виртуальную указку. Реализован автоматический перевод при общении в чате пользователей на разных языках. Для подключения к серверу пользователю достаточно использовать веб-браузер с поддержкой Adobe Flash, то есть это может быть любой компьютер, работающий под управлением Windows, *nix или Mac OS X. Ведется разработка клиента для Android. Конференции могут быть двух видов: открытые (может подключиться любой зарегистрированный пользователь) и закрытые. В случае приватной конфы список допущенных формирует сам выступающий, высылая им данные для доступа. Пользователи в конференции могут быть в роли выступающего, модератора (по умолчанию получает создатель конференции) и слушателя. Работа

виртуального лектора мало отличается от реального: кроме видео, он загружает документы, используя указку, акцентирует внимание на важных моментах, включает аудио выбранного слушателя. Модератор может назначить любого пользователя выступающим, тогда все внимание будет переключено на него. Интерфейс пользователя позволяет приблизить отдельные фрагменты, чтобы лучше рассмотреть их, привлечь внимание «подняв руку», общаться в групповом или приватном чате. Модератор полностью контролирует возможности присутствующих, при необходимости отключает пользователя или переводит в режим «только просмотр». Поддерживается разрешение 320×240, 640×480, 1280×720, на количество подключений BBB каких-либо ограничений не накладывает.

В своей работе BBB использует более десятка OpenSource приложений: FreeSWITCH, Nginx, Flash медиасервер Red5, MySQL, ActiveMQ, Tomcat, Redis, Grails, Xuggler, OpenOffice.org, Image Magick, SWFTools и другие. Ранее в этом списке был Asterisk, но при реализации функции записи разработчики столкнулись с тем, что эта функция в FreeSWITCH не требует дополнительных усилий, и потому отказались от Asterisk/app_konference. Веб-интерфейс BBB переведен на 40 языков, в этом списке есть и русский. Для работы BigBlueButton рекомендуется сервер с CPU Dual Core 2.6 ГГц, 2 Гб ОЗУ и место на жестком диске с учетом записи трансляций. Количество пользователей, которые смогут одновременно общаться на сервере, зависит от мощности оборудования и пропускной



Функция трансляции рабочего стола в BigBlueButton

способности канала. Отдельный поток требует 30-50 кбайт/сек. Приблизительные расчеты можно найти в FAQ (goo.gl/Pii7Y), там же приводятся данные стресс-теста. Для подключения клиентов по умолчанию используется стандартный 80 порт, который не должен быть занят другим приложением. В правилах брандмауэра должны быть открыты порты 80 (HTTP), 935 (RTMP) и 9123 (общий рабочий стол). Доступен исходный код, позволяющий установить BBB на любой компьютер, работающий под управлением Linux, FreeBSD, Mac OS X или Windows. Для Ubuntu и CentOS есть готовые пакеты и репозитарии. Сервер BBB может работать в облачной среде вроде Amazon EC2. Документация на сайте проекта весьма подробна, в ней можно найти ответы практически на все возникающие вопросы – по установке (есть готовые конфиги), конфигурированию, API, локализации, настройке отдельных компонентов (VoIP, nginx и т.п.) и прочим моментам. Свои вопросы можно задать в списке рассылки, предлагается несколько видеоруководств. Доступен образ VM и

демосервер, позволяющие познакомиться с основными возможностями BBB, не устанавливая систему.

А что Asterisk?

В популярном VoIP сервере Asterisk конференцию можно организовать при помощи стандартного приложения MeetMe(app_meetme.so), поддерживающего динамическое создание конференций, защиту паролем, разделение ролей, запись и многое другое. Управление производится при помощи голосового меню. Но качество связи не всегда приемлемое. Кроме того доступно несколько альтернативных решений – ConfBridge(переработанный MeetMe), app_conference (appkonference.sf.net) и его форк app_konference. Так app_conference позволяет организовать аудио и видеоконференцию с несколькими пользователями в приемлемом качестве. При этом он не микширует видеопотоки от участников (только аудио), а просто пересылает нужным абонентам (аудио при этом микшируется). Что существенно снижает требование к оборудованию.

По материалам
tux.in.ua

Инструменты для исследования сетей с интерфейсом командной строки

После рассмотрения программ ping, telnet и dig, продолжим обзор инструментов для исследования сетей с интерфейсом командной строки.

Часть 2

Traceroute

Программа traceroute является инструментом для определения маршрута следования пакетов в сетях, работающих по межсетевому протоколу (Internet Protocol (IP)). Также доступен вариант программы для работы по протоколу IPv6 с названием traceroute6.

Программа traceroute доступна практически во всех UNIX-подобных операционных системах. Также существуют аналоги со сходными функциями, такие, как tracepath в современных дистрибутивах Linux и tracert в операционных системах Microsoft Windows. Операционные системы на основе Windows NT также имеют в своем составе утилиту PingPath с аналогичной функциональностью.

Tracerute отправляет последовательность пакетов протокола межсетевых управляющих сообщений по адресу удаленного узла. Определение адресов промежуточных маршрутизаторов, преодолеваемых пакетами, тесно связано с управлением параметром времени жизни пакета (TTL) в рамках межсетевого протокола. Маршрутизаторы при пересылке пакета уменьшают это значение на единицу и отбрасывают пакет когда значение TTL становится равным нулю, возвращая сообщение об ошибке ICMP (ошибке истечения времени ICMP (ICMP Time Exceeded)) отправившему пакет узлу.

В ходе работы traceroute повышает значение TTL для каждой успешно отправленной группы пакетов. Первые три пакета отправляются с значением TTL равным единице, поэтому ожидается, что первый маршрутизатор не будет производить пересылку пакетов. Следующие три пакета имеют значение TTL равное двум, поэтому второй маршрутизатор отправит ошибку отправившему пакеты узлу. Это продолжается до тех пор, пока пакеты не достигнут узла назначения и он не отправит сообщение о приеме в виде эхо-ответа ICMP (ICMP Echo Reply).

Traceroute использует возвращаемые пакеты с сообщениями ICMP для создания списка узлов, которые преодолевают пакеты с данными по пути к узлу назначения. Три значения времени, возвращаемые для каждого узла по пути следования пакетов являются значениями задержек (времени ожидания) и обычно измеряются в миллисекундах для каждого пакета из последовательности.

Базовый синтаксис:

```
traceroute <целевой узел>
```

Пример использования:

```
traceroute www.google.it
traceroute to www.google.it
(72.14.234.104), 30 hops max, 60
byte packets
1 192.168.0.1 (192.168.0.1) 1.905
ms 5.552 ms 9.496 ms
2 * * *
3 host141-189-static.38-88-b.
business.telecomitalia.it
(88.38.189.141) 50.136 ms 53.519
ms 55.674 ms
4 r-bo83-vl19.opb.interbusiness.
it (80.21.70.162) 58.609 ms
63.730 ms 66.937 ms
5 172.17.5.69 (172.17.5.69)
74.281 ms 76.749 ms 79.629 ms
6 172.17.8.165 (172.17.8.165)
91.052 ms 42.074 ms 42.009 ms
7 172.17.5.241 (172.17.5.241)
49.091 ms 50.790 ms 53.211 ms
8 te1-9-1-0.milano50.mil.seabone.
net (93.186.128.165) 54.800 ms
56.805 ms 59.189 ms
9 te3-2.milano53.mil.seabone.net
(195.22.205.227) 60.866 ms 63.446
ms 65.927 ms
10 72.14.198.233 (72.14.198.233)
68.039 ms 70.902 ms 73.316 ms
11 209.85.249.54 (209.85.249.54)
75.359 ms 78.234 ms 80.581 ms
12 72.14.232.63 (72.14.232.63)
41.714 ms 40.862 ms 41.676 ms
13 mil01s07-in-f104.1e100.net
(72.14.234.104) 46.361 ms 49.293
ms 52.165 ms
```

Whois

WHOIS (произносится как фраза "who is", в переводе означающая "кто это") является протоколом запросов и ответов, широко используемым для осуществления запросов к базам данных, хранящим списки зарегистрированных пользователей или представителей интернет-ресурсов, доменных имен, блоков IP-адресов или автономных систем, при этом протокол также используется и для получения более широкого круга информации. Протокол описывает хранение и доставку содержимого базы данных в форме, пригодной для чтения человеком. Протокол Whois описан в спецификации RFC 3912, использование протокола может осуществляться при помощи одноименной программы.

Базовый синтаксис:

```
whois <домен>
```

Пример использования:

```
whois wikipedia.com
Registrant: Wikimedia Foundation,
Inc. 149 New Montgomery Street
Third Floor San Francisco,
California 94105 United States
Registered through: GoDaddy.com,
Inc. (http://www.godaddy.com)
 Domain Name: WIKIPEDIA.COM
Created on: 12-Jan-01
Expires on: 10-Jan-15
Last Updated on: 01-Mar-06
Administrative Contact: Admin,
DNS dns-admin@wikimedia.org
Wikimedia Foundation, Inc. 149
New Montgomery Street Third Floor
San Francisco, California 94105
United States +1.4158396885 Fax
-- +1.4158820495
Technical Contact: Admin,
DNS dns-admin@wikimedia.org
Wikimedia Foundation, Inc. 149
New Montgomery Street Third Floor
San Francisco, California 94105
United States +1.4158396885 Fax
-- +1.4158820495
Domain servers in listed order:
NS0.WIKIMEDIA.ORG NS1.WIKIMEDIA.
ORG NS2.WIKIMEDIA.ORG
```

Netstat

На компьютерах под управлением GNU/Linux программа netstat позволяет увидеть состояние открытых сетевых соединений.

При запуске программы без параметров командной строки, будет выведен список активных сокетов в системе, например:

```
kaos@kaos:~$ netstat
Active Internet connections (w/o
servers)
Proto Recv-Q Send-Q Local Address
Foreign Address State
tcp 0 0 proxy.surfnet.iac:33297
mg-in-f125.:xmpp-client
ESTABLISHED
tcp 0 0 proxy.surfnet.iac:33995
by1msg5176516.phx.:1863
ESTABLISHED Active
UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type State
I-Node Path unix 3 [ ]
DGRAM 13826 /dev/log unix 2 [ ]
DGRAM 5109 @/org/kernel/udev/
udevd unix 2 [ ]
DGRAM 10443 @/org/freedesktop/
hal/udev_event unix 3 [ ]
STREAM CONNECTED 34721 /
tmp/orbit-kaos/linc-4876-0-
ebf915e300c0 unix 3 [ ] STREAM
CONNECTED 34720
```

Параметры командной строки должны следовать после дефиса (-), а не слеша (/). Наиболее используемые параметры командной строки:

- -a: Показать все активные TCP-соединения и все TCP- и UDP-порты, на которых принимаются соединения.
- -e: Показать статистику работы локальной сети (по технологии Ethernet), включающую в себя количество отправленных и принятых байт и пакетов. Параметр может комбинироваться с параметром -s.
- -g: Показать информацию о членстве в группах многоадресной передачи данных (Multicast) для протоколов IPv4 и IPv6 (должно быть доступно только на современных операционных системах).
- -i: Показать список сетевых интерфейсов и статистику их использования.
- -n: Показать активные TCP-соединения, при этом адреса и номера портов будут представлены в числовом виде и не будет производиться установление имен ресурсов.
- -p: Показать названия процессов, использующих сокеты (работает только в Linux при наличии прав пользователя root; для Windows аналогичная задача выполняется при помощи параметра -b).
- -r: Показать содержимое таблицы маршрутизации (эквивалентно выводу команды route print в Windows).
- -u: Показать состояние UDP-сокетов.
- -t: Показать состояние TCP-сокетов.

Примеры использования:

```
netstat -tlnp
```

Выводит список всех портов, на которых принимаются соединения и соответствующий каждому из них идентификатор процесса (PID). Идентификатор процесса будет выведен только в том случае, когда пользователь имеет привилегии суперпользователя.

```
netstat -an | grep ESTABLISHED |
awk '{print $5}' | awk -F:
'{print $1}' | sort | uniq -c |
awk '{ printf("%st%st",$2,$1) ;
for (i = 0; i < $1; i++)
{printf("*")}; print "" }'
```

Выводит график в ASCII-представлении, отображающий количество установленных соединений с различными IP-адресами. Пример вывода:

```
10.100.0.22 1 *
10.100.0.23 2 **
10.100.1.51 3 ***
120.116.18.134 1 *
87.11.50.125 4 ****
```

```
netstat -ant | awk '{print $NF}'
| grep -v '[a-z]' | sort | uniq
-c
```

Выводит количество соединений и их состояние. Пример вывода:

```
1 CLOSE_WAIT
11 ESTABLISHED
63 LISTEN
21 TIME_WAIT
```

Mtr

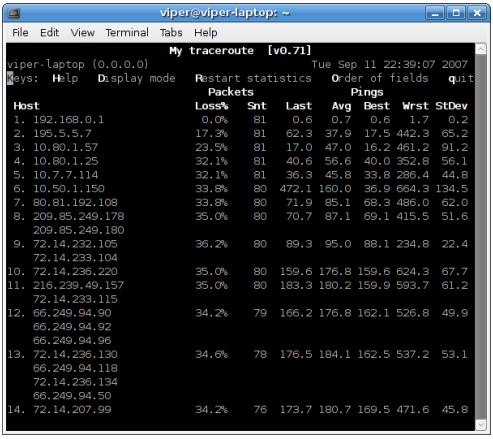
Программа mtr является инструментом для диагностики сети и комбинирует в себе возможности программ traceroute и ping.

После запуска mtr производит исследование соединения между узлом, на котором она запущена и целевым узлом, заданным параметром командной строки путем отправки пакетов со специально установленными низкими значениями параметра TTL. После этого отправка пакетов с низким значением параметра TTL продолжается, параллельно ведется запись времени приема ответных сообщений от промежуточных маршрутизаторов. Это позволяет mtr подсчитать и вывести процент принятых ответов и время ответа маршрутизаторов по пути следования пакетов до целевого узла. Внезапное повышение количества потерянных пакетов или времени ожидания ответа обычно говорит о плохом качестве (или просто о перегрузке) сети.

Базовый синтаксис:

```
mtr <имя узла>
```

Вывод команды "mtr google.com":




```
mtr --report --report-cycles
10 www.google.com > google_net_
report.txt
```

Эта команда позволяет поместить отчет, составляемый программой mtr, в текстовый файл. В режиме создания отчета mtr выводит результаты работы в виде отформатированного текста, включая в него описание необходимого количества пинг-циклов, требующихся для выполнения команды. Этот текстовый отчет может быть впоследствии без труда прикреплен к сообщению электронной почты.

По материалам
rus-linux.net

Простой WiFi-анализатор

При организации соединения между несколькими компьютерами посредством Wi-Fi часто возникает необходимость оценить качество приёма в различных точках помещения. А если говорить об организации Wi-Fi-моста и/или настройке направленных антенн то необходимо ещё учитывать соседние Wi-Fi-точки.

Полноценное исследование эфира требует достаточно высоких затрат, которые обычно не оправданы. Чаще всего просто нужен сканер эфира, который показывает доступные точки и качество их сигнала. Для смартфонов таких приложений существует великое множество, а вот под Linux кроме громоздкого Kismet ничего подходящего найти не удалось. В итоге оказалось проще написать свой инструмент.

Основная идея при написании своего инструмента заключалась в том, что вся необходимая информация содержится в выводе команды:

```
iwlist wlan0 scan
```

Но вывод этой команды не удобен для восприятия. В итоге было решено написать скрипт, который в бесконечном цикле будет запускать эту команду, парсить её вывод и отображать результат. Листинг полученного скрипта:

```
#!/usr/bin/perl

use strict;
use warnings;
use diagnostics;
use Math::Round;

# Если число аргументов не равно единице
if (@ARGV != 1) {
        # Печатаем краткую справку
        print "Usage:\n";
        print "  $0 ifname\n\n";
        # Завершаем работу
        exit;
}

# Будем использовать полученные данные для очистки экрана в дальнейшем
my $clear_screen = `clear`;

# Получаем имя интерфейса
my $ifname = shift;

while (1) {
        # Сканируем эфир
        my $scan_result = `iwlist $ifname scan`;
        # Получаем код ошибки
        my $error_code = $?;
        # Завершаем работу если что-то не так
        exit if $error_code;
        # На скорую руку разбиваем результат сканирования на элементы
        my @scan_results_tmp = split /Cell \d+/is, $scan_result;
        # Начинаем полноценный разбор
        my @scan_results = ();
```

```
        # Перебираем элементы
        foreach my $hotspot_line
(@scan_results_tmp) {
                # Если нет номера
канала значит это мусор, который
надо пропустить
                next if $hotspot_
line !~ m{Channel\:}is;
                # Строим элемент
                my %hotspot = ();
                # Номер канала

$hotspot{'Channel'} = $hotspot_
line;

$hotspot{'Channel'} =~
s{^.+Channel\:(\d+).+$}{$1}is;
                # SSID
                $hotspot{'SSID'}
= $hotspot_line;
                $hotspot{'SSID'}
=~ s{^.+ESSID\:"((.+?)?)".+$}{$1}
is;
                # Наличие шифро-
ваний

$hotspot{'Crypted'} = $hotspot_
line;

$hotspot{'Crypted'} =~
s{^.+Encryption key:(\w+?)\s.+$}
{$1}is;
                # Уровень сигна-
ла. И сразу переводим его в про-
центы
                my $q1 =
$hotspot_line;
                $q1 =~
s{^.+Quality=(\d+)/\d+.+$}{$1}is;
                my $q2 =
$hotspot_line;
                $q2 =~
s{^.+Quality=\d+/(\d+).+$}{$1}is;
$hotspot{'Quality'} = round($q1 *
100 / $q2);
                push @scan_
results, \%hotspot;
        }
        # Сортируем
        @scan_results = sort {
sprintf("%02d", $a->{Channel})
cmp sprintf("%02d",
$b->{Channel}) } @scan_results;

        # Очищаем экран
        print $clear_screen;
        # Печатаем данные
        print sprintf("  % 2.2s
[% 9.9s] [% 32.32s] [% 7.7s]\n",
"Ch", "Quality", "SSID",
"Crypt");
        print sprintf('%1$s'x61 .
"\n", "-");
        foreach my $hotspot (@
scan_results) {
                print sprintf("
%02d [% 8d%%] [% 32.32s] [%
7.7s]\n",
                        $hotspot-
>{Channel},
                        $hotspot-
>{Quality},
                        $hotspot-
>{SSID},
                        $hotspot-
>{Crypted});
        }
        # Делаем паузу
        sleep 1;
}
```

Скрипт принимает один единственный параметр – имя интерфейса, на котором надо осуществлять мониторинг. Например так:

```
./wifiscan.pl wlan2
```

Вывод скрипта выглядит примерно так:

```
  Ch [  Quality] [
SSID] [  Crypt]
-------------------------------
----------------------------
  01 [      93%] [
Yuldash House] [     on]
  02 [      94%] [
serr_dom] [     on]
  04 [      94%] [
home-of-moose] [     on]
  06 [      93%] [
wifi.tattele.com] [    off]
  06 [      93%] [
KEENETIC 4G] [     on]
  06 [      94%] [          MTS_
telefon_2-65-60-30] [     on]
  06 [      94%] [
home.net] [     on]
  08 [      93%] [
YOTA] [     on]
  10 [      93%] [
Estucador] [     on]
  10 [      93%] [
DIR-300NRUB6] [     on]
```

Если нужны будут какие-то другие параметры то их получение и отображение легко дописать в скрипт. Автор использовал этот скрипт при экспериментах с баночными антеннами и с помощью этого скрипта смог найти нужное направление на необходимую точку.

По материалам
ylsoftware.com

Linux-бэкдор, организующий скрытый канал связи в легитимном сетевом трафике

Компания Symantec в результате разбора атаки на одного из крупных хостинг-провайдеров выявила новый вид бэкдора для GNU/Linux. Бэкдор выполнен в форме разделяемой библиотеки, перехватывающей ряд стандартных вызовов, таких как read, EVP_CipherInit, fork и ioctl. Библиотека связывается с работающими на системе сетевыми процессами, такими как sshd, берёт на себя обработку указанных вызовов и получает контроль над трафиком поражённых серверных приложений.

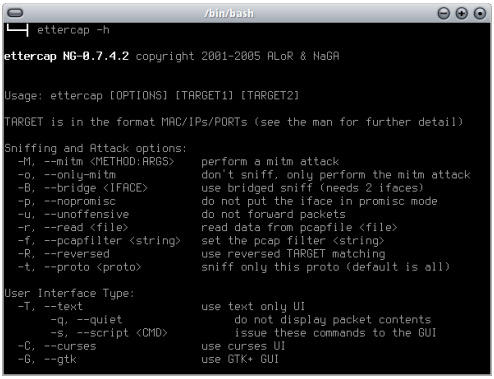
При работе бэкдор не сохраняет файлов, не создаёт сокетов и не инициирует сетевые соединения, прикрываясь в своей активности поражённым серверным процессом, что затрудняет его обнаружение. Управляющие команды передаются в составе штатного сетевого трафика, через интеграцию подставных блоков. Бэкдор отслеживает появление в нормальном, не вызывающем подозрение, трафике маску ":!;:", при обнаружении которой декодирует следующий за ней блок данных. Данные зашифрованы с использованием шифра Blowfish и следуют в формате Base64.

Через закодированные блоки, примешанные в обычный трафик, могут передаваться управляющие команды для выполнения произвольной shell-команды или отправки в ответ накопленных бэкдором данных. Основной функцией бэкдора является перехват и накопление конфиденциальных данных, таких как пароли, ключи шифрования и email-адреса. В частности, поддерживается перехват паролей и SSH-ключей пользователей, подсоединяющихся к поражённой системе.

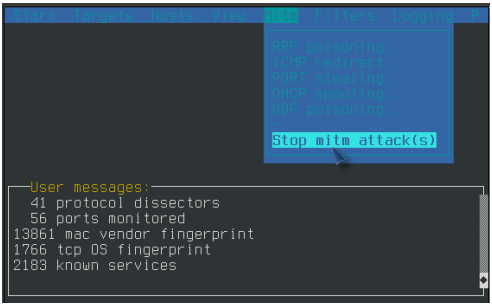
В качестве методов выявления бэкдора может использоваться анализ наличия маски ":!;:" в трафике. Но этот метод не эффективен, так как бэкдор может длительное время не проявлять себя. Более надёжным вариантом является сохранение дампа памяти работающего процесса sshd и поиск в нём специфичных для бэкдора строковых данных, таких как "key=", "dhost=", "hbt=3600", "sp=", "sk=" и "dip=". Подробности об атаке, в результате которой был установлен бэкдор, не сообщаются. Как правило, проникновение в систему осуществляется через перехват пароля администратора или эксплуатацию неисправленных уязвимостей, например, в панелях управления хостингом.

По материалам
opennet.ru

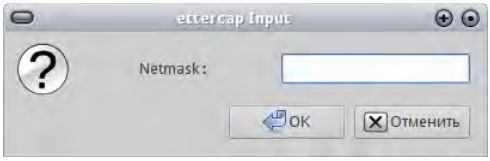
Анализ безопасности компьютерных сетей

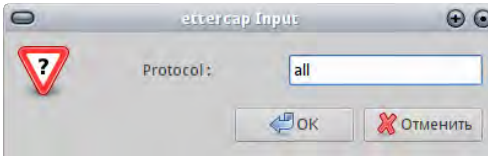


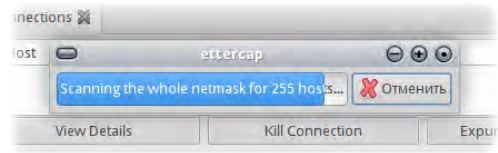
Ettercap – небольшая консольная утилита, имеющая GTK графический интерфейс (GUI), предназначенная для анализа безопасности компьютерных сетей.



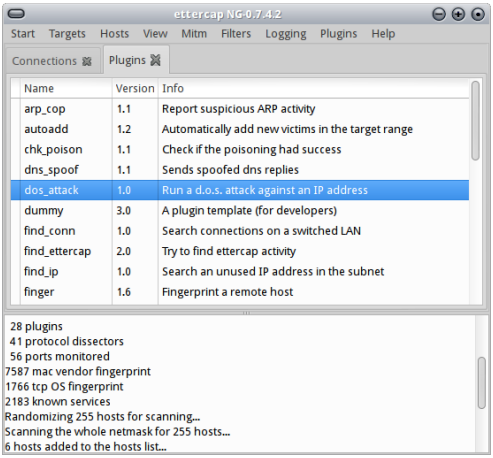
Основное предназначение Ettercap проведение MITM атаки (атака "человек посередине"), имеет возможность анализа сетевого трафика, фильтрации контента на лету и множество других интересных возможностей...

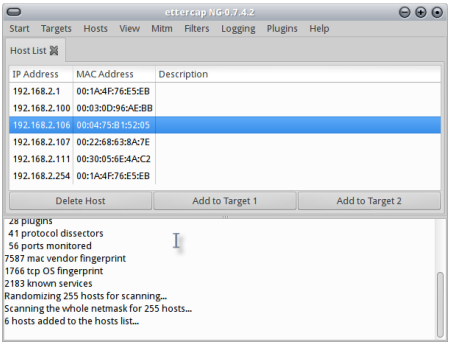


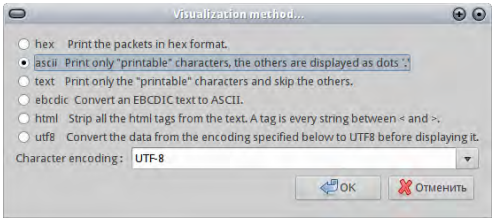


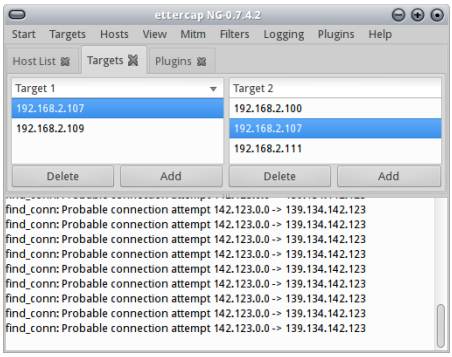


Ettercap поддерживает как активные, так и пассивные вскрытия протоколов (поддерживает протоколы Telnet, FTP, POP3, IMAP, SMB, HTTP и многие другие), включает большое количество функций для анализа сети и узла.





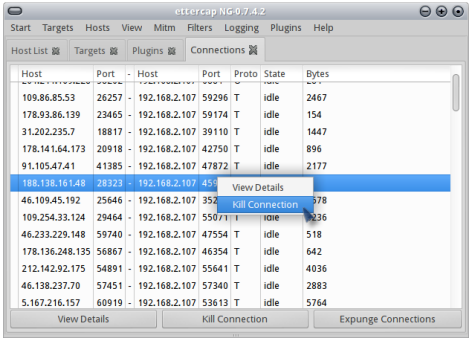




Ettercap имеет множество функций и расширяется за счёт большого количества плагинов, для работы требует прав администратора (root), полученные в результате выполнения атаки log-файлы могут быть сразу просмотрены (например если в ходе перехвата, в трафике были замечены пароли, они будут показаны), настройка Ettercap производится правкой конфигурационного файла:

/etc/ettercap.conf

Особую популярность утилита получила за возможность выполнять ARP-spoofing (техника сетевой атаки).



По материалам
zenway.ru

Кибератаки в подробностях: СПУФИНГ ПАКЕТОВ

Мы начали серию статей о наиболее известных типах кибератак, которые оказывают воздействие на IT-инфраструктуру организаций с детального описания атак отказа в обслуживании. В этот раз мы рассмотрим атаку, осуществляемую при помощи спуфинга пакетов, которая является одной из любимых атак взломщиков и широко используется для эксплуатации уязвимостей сетей. Также мы рассмотрим обстоятельства, при которых эта атака может затронуть Linux-системы и обсудим способы ее сдерживания.

Спуфинг по определению является мистификацией или обманом кого-либо (слово происходит от английского слова "spoof", в переводе означающего мистификация). Для понимания механизма осуществления атаки при помощи спуфинга пакетов, нам необходимо исследовать в подробностях структуру IP-пакета. Многие кибератаки основываются на дефектах, допущенных при проектировании фундаментальных механизмов функционирования сетей и спуфинг не является исключением. Рассмотрите рисунок 1.

Рисунок 1: Структура Ethernet-фрейма и IP-пакета

Как известно, стандартный Ethernet-фрейм представляет из себя область данных с различными заданными полями, такими, как MAC-адреса источника и назначения, поле контрольной суммы, поля синхронизации (preambles) и другие поля. В случае протокола TCP/IP, TCP-пакет дополняет IP-дейтаграмму и они находятся в стандартном Ethernet-фрейме. TCP-пакет содержит информацию о постоянном соединении, в то время, как IP-пакет содержит информацию о исходном и целевом адресе и порте. Задачей различных уровней OSI является объединение данных из этих пакетов.

Как мы упоминали в прошлой статье, соединение в TCP/IP устанавливается при помощи трехэтапного рукопожатия (SYN, SYN-ACK, ACK). Это рукопожатие служит для установления соединения между двумя сетевыми картами, которые затем используют последовательности пакетов и подтверждения о доставке пакетов для отправки или получения данных. Этот

процесс обмена данными обычно заканчивается рукопожатием FIN/FIN-ACK.

Поля с IP-адресами отправителя и целевой системы устанавливают то, какие системы участвуют в обмене данными, а поля с номерами портов указывают на то, какие приложения участвуют в обмене данными. Поля IP-адресов в IP-пакетах заполняются автоматически при помощи кода реализации высших уровней сетевого стека, тем не менее, злонамеренно настроенные пользователи с помощью специального программного обеспечения могут изменить данные в этих полях; при помощи этих действий и производится атака, основанная на спуфинге пакетов, процесс ее осуществления показан на рисунке 2.

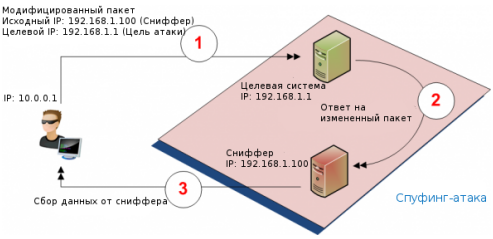


Рисунок 2: Атака при помощи спуфинга пакетов

В случае простейшей атаки, основанной на спуфинге пакетов, взломщик генерирует IP-пакеты с адресом целевой системы, а поле IP-адреса источника модифицируется таким образом, что вместо IP-адреса отправителя задается адрес узла, который может использоваться для сбора данных об атакуемой системе или на нем может работать сниффер.

TCP/IP стек на атакуемой системе спроектирован таким образом, что он должен ответить отправителю пакета при его приеме. После этого узел со сниффером принимает пакет, отправленный в ответ. Стоит упомянуть о том, что если на атакуемом узле установлен сниффер пакетов, он покажет, что пакеты приходят с компьютера, который взломщик использует для сбора данных, что не соответствует действительности. Таким образом, злоумышленнику с успехом удается скрыть свой адрес и в то же время собрать информацию о пакетах, выданную сниффером, поскольку его реальный IP-адрес не раскрывается в процессе атаки.

Теперь давайте разберемся в причинах, по которым злоумышленники предпочитают использовать спуфинг. Хотя сокрытие информации о себе является приоритетной задачей, стремления взломщика распространяются шире. Собирая информацию о пакетах при помощи сниффера на выделенном для этого узле, взломщики могут получить большой объем информации об атакуемой системе. Важная информация, такая, как открытые порты, тип операционной системы, сетевые приложения уровня 7 OSI, используемые типы криптографических алгоритмов и многие другие данные могут быть собраны без раскрытия адреса злоумышленника.

В качестве примера сбора информации можно привести ситуацию, когда взломщик отправляет пакет с подмененными данными для того, чтобы узнать, работает ли Web-сервер на атакуемой системе. Как только информация о используемых службами портах извлекается при помощи сниффера, следующая серия пакетов с подмененными данными отправляет-

ся атакуемой системе для установления сессии telnet и получения заголовков от Web-сервера, откуда может быть получена информация об используемой операционной системе, типе Web-сервера и системах безопасности, используемых на атакуемой системе.

Следует знать, что для взломщиков нет необходимости в постоянном доступе к узлу с установленным сниффером; фактически использование мощных программ для исследования пакетов может помочь атакующему в использовании только своего компьютера для отправки пакетов с подмененными данными, сборе данных на этом же компьютере и сохранять при этом невидимость для атакуемой системы.

Опытные взломщики могут использовать спуфинг пакетов также и для получения информации о состоянии межсетевого экрана. Как мы знаем, межсетевой экран осуществляет постоянный процесс фильтрации пакетов на основании изменяемых правил, что в конечном итоге сводится к решению: какие пакеты должны быть пропущены, а какие - отброшены. Когда один узел соединяется с другим узлом, защищенным межсетевым экраном, первый узел сначала проводит TCP-рукопожатие с межсетевым экраном.

Если соединение разрешается правилами, межсетевой экран самостоятельно устанавливает другое соединение с удаленным узлом и обмен данными между узлами происходит при участии межсетевого экрана в качестве промежуточного звена. Также, когда удаленный узел пытается отправить подтверждение о приеме данных узлу, инициировавшему соединение, межсетевой экран должен перехватить его и проверить, ожидает ли узел это подтверждение, а если не ожидает - отбросить пакет. Все это относится к современным межсетевым экранам, но не относится к устаревшим недорогим межсетевым экранам, большинство из которых пропускают ACK-пакеты, предоставляя возможности для атак при помощи спуфинга.

Существуют три причины, по которым сети являются уязвимыми к атакам, осуществляемым при помощи спуфинга пакетов. Первая причина состоит в том, что содержимое пакета может быть модифицировано без лишних сложностей при помощи доступных широкому кругу пользователей свободных утилит. Второй причиной является то, что множество приложений даже сегодня все еще используют комбинацию исходного и целевого IP-адреса как безопасный метод аутентификации пакета.

В качестве примера можно привести Web-сервера, позволяющие или не позволяющие осуществлять HTTP-запросы на основании изменяемого списка IP-адресов. Такие системы становятся бесполезными в том случае, если исходный адрес в пакете подменен на адрес из списка разрешенных. Третьей причиной является то обстоятельство, что маршрутизаторы осуществляют пересылку пакета основываясь только на целевом адресе и по умолчанию не обращают никакого внимания на адрес отправителя.

Типы атак с применением спуфинга пакетов

Зная основы осуществления спуфинга пакетов, давайте подробнее рассмотрим

методы осуществления атак. На техническом уровне существуют два типа спуфинга "слепой" спуфинг ("blind" spofing) и "неслепой" спуфинг ("non-blind" spoofing). Как мы обсудили ранее, порядковые номера пакетов и номера подтверждений лежат в основе передачи данных, поэтому их подмена является важным требованием при осуществлении атак.

При атаке, основанной на слепом спуфинге, атакующий отправляет множество пакетов целевой системе для установления порядковых номеров пакетов и номеров подтверждений. Как только номера установлены, атакующему не составляет труда подготовить новый набор измененных пакетов для сбора передаваемых данных. В случае атаки, основанной на неслепом спуфинге, атакующий должен находиться в одной подсети с целевой системой для того, чтобы без лишних сложностей узнать порядковые номера пакетов и номера подтверждений. Как только эти номера установлены, атакующий может разорвать существующее соединение целевой системы с другим узлом и переустановить его при помощи подмены порядковых номеров пакетов на своем узле.

Расширенной версией этой атаки является атака перехвата с участием человека (man-in-the-middle), при которой перехватываются данные всей сессии для их расшифровки и похищения.

Теперь давайте рассмотрим вопрос о том, как базовые понятия, относящиеся к спуфингу пакетов, применяются взломщиками для вмешательства в работу сетевых служб, работающих по протоколу TCP.

Подмена номеров портов (IP port spoofing)

При этом типе атаки номер исходного порта модифицируется с целью ввести в заблуждение устройства NAT и межсетевые экраны, а также скрыть свое присутствие в сети. Межсетевые экраны, не обслуживаемые должным образом могут использовать устаревшие правила, согласно которым могут быть открыты определенные порты на узлах с известными IP-адресами. Данный вид атаки использует это обстоятельство.

Эта достаточно серьезная атака, так как злоумышленник может находиться за пределами сети и при этом иметь доступ к внутреннему трафику.

Подмена записей ARP (ARP spoofing)

Поскольку спуфинг подразумевает модификацию или создание пакетов, современные взломщики не ограничиваются только подменой IP-полей. В этом типе атаки, также известном как ARP poisoning, атакующий отправляет измененные ARP-пакеты в локальную сеть для ассоциации MAC-адреса своей сетевой карты с IP-адресом целевой системы.

Исходя из этого, трафик, предназначенный для целевой системы, теперь будет направляться на сетевую карту злоумышленника благодаря связи между IP- и MAC-адресом. Таким образом эта атака является атакой перехвата с участием человека. Атакующий может скрыть факт проведения атаки, отправляя все принятые данные на адрес реального получателя и, поскольку в этом процессе потери данных

не происходит, практически невозможно определить узел, похищающий данные.

Подмена записей DNS (DNS spoofing)

Эта атака также известна как DNS poisoning и она приводит к более серьезным последствиям. В данном случае сервер доменных имен подвергается воздействию, приводящему к изменению записей, соответствующих именам доменов с целью перенаправления на узел злоумышленника с заданным IP-адресом. Это приводит к тому, что Web- и email-трафик отправляется на узел злоумышленника. Эта атака проводится путем создания нескольких пакетов с измененными значениями полей IP-адреса, порта и типа службы.

Последствия этой атаки могут быть очень плачевными, например, может быть заменена главная страница сайта или похищены сообщения электронной почты.

Атаки на Email- и Web-службы (Email and Web spoofing)

Говоря о 7 уровне OSI, следует упомянуть, что аналогичная техника применяется также и для создания поддельных адресов электронной почты, Web-запросов и гиперссылок. Обычно это делается с целью вброса и распространения вредоносного программного обеспечения (вирусов и троянов) среди пользователей. Помните, что спуфинг применяется при организации атак отказа в обслуживании, так как злоумышленник всегда хочет остаться незамеченным.

Защита систем, основанных на свободном программном обеспечении

Хотя атаку, основанную на спуфинге пакетов довольно сложно сдерживать, существует ряд превентивных мер, которые администраторы сетей должны применить в рамках своей инфраструктуры. Так как атаки, основанные на спуфинге пакетов, начинают свою работу на уровне 2 OSI, реальная защита от них может быть реализована в критических компонентах сети, таких, как маршрутизаторы, межсетевые экраны, свитчи и других.

Развертывание современного межсетевого экрана и активизация его возможностей для борьбы со спуфингом пакетов может быть первым уровнем защиты. В качестве ежедневной работы администратор должен исследовать при помощи сценария системные журналы межсетевых экранов, маршрутизаторов и свитчей в поисках множества дубликатов ACK-пакетов.

Также необычно большое количество SYN-пакетов, оставшихся без ответа может говорить о применении спуфинга пакетов. Действительно работающим механизмом определения наличия атаки при помощи спуфинга является проверка серии пакетов для заданного множества исходных и целевых IP-адресов и определение того, нет ли среди них сторонних IP-адресов в рамках данной сессии обмена данными.

Эта задача достаточно сложна для выполнения с помощью сценария оболочки, поэтому для сложных конфигураций сетей развертывание системы устранения

проникновений может оказать значительную помощь администратору. Существует правило при исследовании пакетов, которое формулируется следующим образом: если IP-адрес интерфейса из внутренней сети появляется в системном журнале, где фиксируются данные об обращениях к внешнему интерфейсу, то это четко указывает на наличие проблемы.

Свободные системы на основе Linux поставляются со встроенной, но мало известной возможностью, называемой проверкой исходного адреса (source address verification). Это функция ядра, которая при включении позволяет отбрасывать пакеты, которые выглядят так, как будто пришли из внутренней сети, но на самом деле это не так. Последние версии ядра, поставляемые в составе таких дистрибутивов, как Ubuntu и CentOS, поддерживают эту возможность, а в том случае, если ваш дистрибутив не поддерживает эту возможность, пришло время обновиться. Модификация файла hosts.conf и добавление в него параметра nospoof on является еще одним уровнем защиты, который следует попробовать применить.

В плане определения наличия атаки для небольших сетей на основе Linux существует отличная утилита под названием arpwatch, которая очень полезна. Она отслеживает IP и MAC-адреса, записывает все изменения и может использоваться со сценариями для уведомления администраторов о возможной атаке. Сценарии могут использоваться также для исследования системного журнала и поиска аномалий в плане измененных исходных адресов.

Заключение

Атаки на основе спуфинга пакетов являются атаками, которые сложно нейтрализовать. Они могут привести к серьезным потерям данных, при этом существуют пути определения и противодействия этим атакам. Настройка межсетевых экранов, свитчей и маршрутизаторов является важным шагом для защиты сетей от атак на основе спуфинга и системные администраторы должны знать об этом.

Компании, работающие с финансами являются наиболее частыми жертвами атак такого типа и их команды управления IT-инфраструктурой должны предпринимать необходимые шаги для предотвращения финансовых потерь и недопущения причинения ущерба их репутации. Установка систем обнаружения проникновения без сомнения помогает в контроле безопасности сетевой IT-инфраструктуры.

По материалам
rus-linux.net

Хотите стать частью команды UserAndLINUX?

Если вы давно используете Linux, либо только начали интересоваться продуктами OpenSoure, либо же просто интересуетесь компьютерными и техническими новинками, то мы с радостью примем вас в нашу дружную команду!

Каждый из нас именно так и попал в команду журнала UserAndLINUX – мы просто любим Linux и считаем, что обязаны нести эту любовь в массы. И мы знаем, как отплатить нашей любимой операционной системе – создать сообщество людей с общими интересами, поддерживать друг друга и помогать новичкам в этом интересном деле.

И не имеет значения, опытный ли вы программист, или одаренный школьник, или дизайнер, инженер, секретарь... Ваши идеи в совокупности с нашими могут помочь другим людям узнать, что такое Linux и с чем его едят!

Чем же вы можете нам помочь?

У вас есть творческие способности, креативное чувство стиля? Обладаете вкусом? Создавайте красивые темы и фоны (людям нравятся красивые картинки!) с командой наших дизайнеров!

Владеете иностранным языком? Переводите статьи с англоязычных ресурсов, чтобы наши читателя всегда имели удовольствие читать свежие интервью и новости со всего мира!

Любые навыки, которыми вы владеете, могут помочь команде UserAndLINUX – присоединяйтесь!
Предлагайте свои идеи. Учитесь вместе с нами. Развивайте новые умения, способности.

Спрашивайте. Задавайте вопросы, не стесняйтесь! Нам всегда нужны люди. Не думайте, что вы не сможете помочь, потому что вы не умеете программировать, администрировать или только начинаете работать в Linux как в системе, которая установлена у вас на компьютере.

Существует миллион способов внести свой вклад.
Присоединяйтесь к работе над журналом UserAndLINUX и приложением «Больше чем USER»!

Оставляйте свои вопросы, координаты и описание того, чем бы вы хотели помочь:
magazine@ualinux.com
на форуме ***http://ualinux.com/forum/userandlinux***
в нашей группе Вконтакте - ***https://vk.com/userandlinux***
или группе на Facebook - ***https://www.facebook.com/groups/userandlinux/***

С уважением,
коллектив журнала UserAndLINUX

User And LINUX

UA Linux
open source ■ open future

Адрес журнала в Интернете:
http://ualinux.com/journal

Обсуждение журнала
на форуме:
http://ualinux.com/forum

По вопросам
приобретения журнала:
http://ualinux.com/pay

Адрес редакции:
Украина, 03040,
г. Киев, а/я 56
Email: magazine@ualinux.com

Тип издания:
электронный

Регулярность: ежемесячный
Дата выпуска: 03.04.2014 г.
Тираж: свыше 60 000 загрузок*

*указано среднестатическое ежемесячное суммарное значение, сформированное из полученной статистики загрузок журнала с официального сайта и других известных источников распространения (ftp, http и torrent)

Государственный реестр СМИ
Свид-во: КВ 18270-7070Р от 24.10.2011

Международный стандартный
серийный номер
ISSN: 2223-6988


User
And
LINUX
Больше чем user