LUKS good! Станьте Linux на шифрованный раздел и делаем удобной работу с ним

Автор: Сергей Ковалев | Опубликовано: 16.08.2011 | Последнее изменение: 16.08.2011

Содержание статьи

1. Как устроен шифрует LUKS
2. Необходимо
3. Подготовка раздела для шифрования
4. Создание шифрованного раздела
5. Монтирование шифрованного раздела
6. Автоматическое монтирование в Linux Debian
7. Заключение

Сегодня многодисковые конфигурации не в диковинку. В Windows для этого есть RAID-массивы, в Linux — Software RAID. Если для того же или другого раздела нужно шифрование, то в Linux для этого есть LUKS. Сегодня мы расскажем, как с помощью LUKS зашифровать раздел, как его монтировать и как сделать монтирование раздела при загрузке с помощью файлов /etc/crypttab и /etc/fstab (добавляем в /etc/fstab строчку, которая монтирует раздел, а в /etc/crypttab — строчку, которая говорит, что раздел нужно монтировать с паролем, который спрашивается при загрузке).

Шифрование данных для сервера необходимо, если на сервере хранятся конфиденциальные данные, которые не должны быть доступны посторонним. Шифрование также полезно в случае кражи сервера, если на сервере хранятся пароли или ключи к сервисам.

Шифрование данных для сервера необходимо, если на сервере хранятся конфиденциальные данные, которые не должны быть доступны посторонним. Шифрование также полезно в случае кражи сервера, если на сервере хранятся пароли или ключи к сервисам.

DISCLAIMER

Tech Lab: Использование материалов с сайтов на форумах

КАК УСТРОЕН РАЗДЕЛ LUKS

Раздел LUKS имеет следующую формы:

```
LUKS phdr | KM1 | KM2 | ... | bulk data
```

Где заголовок LUKS, phdr, данные на диске следует сразу после заголовка данных (bulk data). KM1, KM2 — это ключевые данные для расшифровки данных, шифрованных паролем.

Для каждого пароля, который используется для создания шифрованного раздела, создается свой ключ. Ключ расшифровывается только при вводе пароля.

Если в LUKS для одного шифрованного раздела прописывается несколько ключей, то что в каждом из ключей шифруется одинаковый ключ. Ключ не хранится на диске в открытом виде.

Заголовок и ключи хранятся сразу после раздела, поэтому при изменении размера раздела, нужно сначала уменьшить раздел, а потом уменьшить файловую систему. Если сначала уменьшить файловую систему, то можно потерять данные.

Для управления шифрованным разделом используется утилита cryptsetup.

Утилита cryptsetup использует:

- device-mapper
- Linux Kernel crypto API
- стандартные утилиты mkfs/fsck
- стандартные утилиты swapon/swapoff
- стандартные утилиты mount/umount

Для работы с cryptsetup требуется только root.

ПОДГОТОВКА

Необходимо иметь свободное место на диске для формирования раздела. Есть возможность использовать для шифрования раздела, который уже содержит данные. Для этого нужно сначала уменьшить раздел, а потом уменьшить файловую систему. Если сначала уменьшить файловую систему, то можно потерять данные.

Рассмотрим случай, когда есть диск с разделами /dev/sda1 и /dev/sda2. Нужно зашифровать раздел /dev/sda2 и сделать его монтирование при загрузке.

Для шифрования раздела нужно:

- Размонтировать раздел
- Уменьшить раздел
- Уменьшить файловую систему
- Зашифровать раздел
- Сделать файловую систему
- Смонтировать раздел

Для работы с cryptsetup требуется пакет cryptsetup-bin.

ШИФРУЕМ

Размонтируем раздел /dev/sda2. Проверяем, что раздел /dev/sda2 не смонтирован:

```
# mount | grep /dev/sda2
```

Если раздел смонтирован, то его размонтируем. Если раздел не смонтирован, то переходим к уменьшению раздела.

Уменьшаем раздел /dev/sda2 на 16 мегабайт (для хранения заголовка LUKS):

```
# fdisk -l /dev/sda
# fdisk /dev/sda
Command (m for help): d
Partition number (1-4): 2
Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (2048-41943039, default 2048): 2048
Last sector, +sectors or +size{K,M,G,T,P} (2048-41943039, default 41943039): +20M
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at the next
reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
# fdisk -l /dev/sda
# partprobe /dev/sda
# fdisk -l /dev/sda
```

Уменьшаем файловую систему:

```
# e2fsck -f /dev/sda2
# resize2fs /dev/sda2 20M
```

Создаем шифрованный раздел:

```
# cryptsetup luks-format /dev/sda2
WARNING!
========
This will overwrite data on /dev/sda2 irrevocably.
Are you sure (Type uppercase yes): YES
Enter passphrase for /dev/sda2:
Verify passphrase:
```

Проверяем, что раздел /dev/sda2 стал шифрованным:

```
# cryptsetup luks-uuid /dev/sda2
# blkid /dev/sda2
/dev/sda2: UUID="ccb0d0d0-0000-0000-0000-000000000000" TYPE="crypto_LUKS"
```

Открываем раздел:

```
# cryptsetup luks-open /dev/sda2 crypt
Enter passphrase for /dev/sda2:
```

После этой операции будет создан в системе поддерживаемый раздел в /dev/mapper:

```
# ls -l /dev/mapper
total 0
crypt -> /dev/dm-0
```

Создаем на разделе файловую систему:

```
# mkfs -t ext2 -L crypt /dev/mapper/crypt
mke2fs 1.41.12 (17-May-2010)
Filesystem label=crypt
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stack size=4096 (log=2)
5120 inodes, 20480 blocks
1024 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=20971520
3 block groups
8192 blocks per group, 8192 fragments per group
1707 inodes per group
Superblock backups stored on blocks:
        8193

Writing inodes: done
Writing superblocks and filesystem accounting information: done
```

Монтируем раздел /mnt/crypt и отмонтируем его (чтобы после сборки не возникало проблем):

```
# mkdir /mnt/crypt
# mount /dev/mapper/crypt /mnt/crypt
# ls -l /mnt/crypt
drwxr-xr-x 2 root root 1024 Jul 31 19:47 lost+found
# umount /mnt/crypt
```

Для корректного монтирования раздела при загрузке нужно прописать в файлах /etc/crypttab и /etc/fstab:

```
# cat /etc/crypttab
crypt /dev/sda2 none luks
# cat /etc/fstab
/dev/mapper/crypt /mnt/crypt ext2 defaults 0 2
```

Проверяем, что раздел монтируется при загрузке:

```
# /sbin/reboot
# mount | grep /mnt/crypt
/dev/mapper/crypt on /mnt/crypt type ext2 (rw)
```

Если раздел не смонтировался, то нужно найти причину. Для этого нужно посмотреть /var/log/syslog:

```
# tail -f /var/log/syslog
Jul 31 19:50:01 ubuntu CRON[2000]: (root) CMD (/usr/lib/sysstat/sa1 1 1)
```