

PRENUMERATA LINUX MAGAZINE

Nie przegap
takiej okazji!



Najszybszy sposób zamówienia prenumeraty:

<http://www.linux-magazine.pl>

Infolinia: 0801 800 105

BEZPIECZEŃSTWO DANYCH W LINUKSIE – RAID I LVM, CZ. I

System komputerowy rozumiany jest jako jedność dwóch technologii – sprzętu i oprogramowania użytkowego.

Sprzęt się psuje, oprogramowanie zmienia, natomiast dane pozostają i w wielu wypadkach są bezpieczne.

Rozsądna polityka bezpieczeństwa zakłada spójność i dostępność danych danych w każdej chwili.

Czy Linuks jest na to gotowy?

CEZARY GAJDZIŃSKI

Wprowadzenie

Dane w systemach komputerowych (systemach przetwarzania) muszą spełniać wiele kryteriów, aby można je było uznać za bezpieczne. Podstawowe z tych kryteriów, to:

1. Dostępność
2. Poufność
3. Spójność

W literaturze powyższe warunki znane są jako „C.I.A.” (confidentiality, integrity, availability). Zaprzeczeniem tych warunków są:

1. Zniszczenie
2. Ujawnienie
3. Zmiana

Te z kolei znane są jako „D.A.D.” (destruction, alteration, disclosure).

Procesy składające się na zapewnienie poufności danych są ściśle związane z ich dostępnością i spójnością w sensie polityki bezpieczeństwa.

Innym istotnym warunkiem wymaganym od dostępu do danych jest niezawodność (reliability). Niezawodność i dostępność są pojęciami, które się uzupełniają: dostępność jest mierzona empirycznie, niezawodność statystycznie, i w takim sensie użytkownik ocenia, czy dane są „on-line”.

Ponieważ niezawodność zazwyczaj jest możliwa do obliczenia na podstawie paramet-
 rów *MTBF* (mean time between failure) podawanych przez producenta, nie będziemy się nią tutaj zajmować.

Nasza uwaga będzie skupiona na dostępności i spójności danych.

Przedstawimy dwa proste i pozornie podobne rozwiązania dla pamięci masowych, które korzystają, odpowiednio, z narzędzi *mdadm* dla *RAID* i *LVM* dla wolumenów. Rozwiązania te są na tyle proste, że każdy z dostępem do Linuksa może dokonać ich symulacji, oraz na tyle zaawansowane, że pokazują delikatne punkty, w których należy zachować ostrożność. *RAID* zapewni nam dostępność danych, *LVM* plastyczność, natomiast wybrany (odpowiednio) system plików, spójność danych.

Stworzenie dostępnego wolumenu wymaga zrozumienia wszystkich etapów planowania, związanych z jego budową. Są to:

1. Poznanie geometrii dysków fizycznych.
2. Utworzenie partycji (lub slice'ów), o ile to jest konieczne.
3. Utworzenie urządzeń *RAID*, lub urządzeń *LVM*.
4. Utworzenie na zaplanowanej konfiguracji docelowego systemu plików.

Poniżej pokażemy, że każdy z tych etapów planowania jest bardzo istotny i niekoniecznie

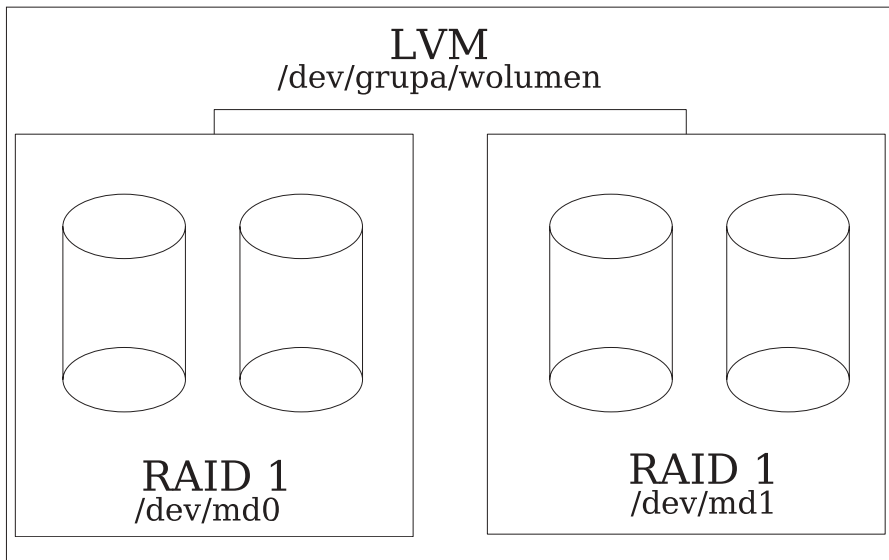
rozpatrywać je należy w podanej kolejności!

Do dyspozycji mamy identyczne technologicznie „dyski twarde” o ustalonej pojemności. Ponieważ nie zawsze są one po ręką, będziemy się posiłkować tzw. pętlą zwrotną (*loopback*). Linux posiada bardzo wygodne narzędzie do tworzenia takich pętli zwrotnych, mianowicie *losetup*. Problemem przy stosowaniu *losetup* jest brak geometrii tworzonych urządzeń, co w niektórych sytuacjach może sprawiać problemy (np. przy analizie systemu plików). Czasami udaje się „oszukać” *losetup* podając w *fdisk* fikcyjną geometrię.

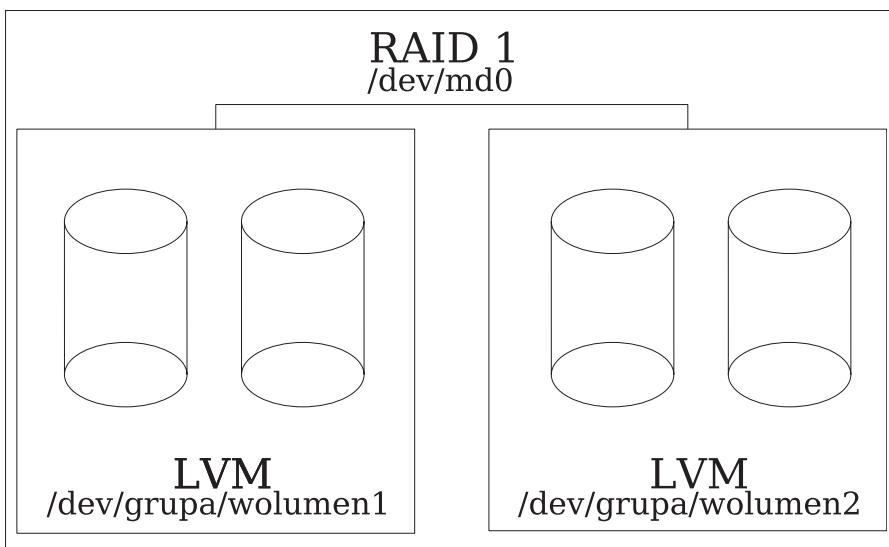
Utworzymy teraz sześć urządzeń blokowych o pojemności 100MB każdy. Będziemy je dalej nazywać *pseudodyskami*.

```
for i in 1 2 3 4 5 6; do dd
if=/dev/zero of=/tmp/disk$i bs=1M
count=100; done
for i in 1 2 3 4 5 6; do losetup
/dev/loop$i /tmp/disk$i; done
```

Dlaczego sześć pseudodysków i dlaczego 100MB? Za chwilę stanie się to jasne (jak również to, że niekoniecznie sześć, i niekoniecznie 100MB). Na pewno uda się wygospodarować 1GB wolnego miejsca na dysku (przy dzisiejszych pojemnościach?).



Rysunek 1. LVM zbudowany na RAID-1



Rysunek 2. RAID-1 zbudowany na LVM

Uważny czytelnik na pewno zauważył, że milcząco założyliśmy istnienie urządzeń `/dev/loop$i`. W razie problemów z ich istnieniem na pewno pomoże rekompilacja jądra systemu i `mknod`.

Konfiguracja I.

Bazę docelowego wolumenu stanowią dwa urządzenia `RAID-1` wykorzystane przez `LVM` do zbudowania wolumenu.

Konfiguracja II.

Bazę docelowego wolumenu stanowią dwa wolumeny `LVM` wykorzystane do zbudowania urządzenia `RAID-1`.

Pojemność końcowych wolumenów jest podobna. Istotne różnice w ich funkcjonal-

ności wynikać będą z kolejności zastosowanych narzędzi: `{mdadm, LVM}` i `{LVM, mdadm}`. Kolejność determinuje w sposób istotny lokalizację informacji o konfiguracji, a to z kolei wpływa na funkcjonalność zdefiniowaną przez wspomniany model C.I.A.

Przejdźmy teraz do zbudowania wolumenów we wskazanych konfiguracjach.

Konfiguracja I.

RAID

```
mdadm --create /dev/md0 --level=1
--raid-devices=2 /dev/loop1/dev/
loop2
mdadm --create /dev/md1 --level=1
--raid-devices=2 /dev/loop3/dev/
loop4
```

LVM

```
pvcreate /dev/md0 /dev/md1
vgcreate -s 1m grupa /dev/md0/dev/md1
lvcreate -i 2 -I 16 -L 190m -n
wolumen grupa
```

Konfiguracja II

LVM

```
pvcreate /dev/loop1 /dev/loop2
/dev/loop3 /dev/loop4
vgcreate -s 1m grupa /dev/loop1
/dev/loop2 /dev/loop3 /dev/loop4
lvcreate -i 2 -I 16 -L 190m -n
wolumen1 grupa /dev/loop1 /dev/
loop2
lvcreate -i 2 -I 16 -L 190m -n
wolumen2 grupa /dev/loop3 /dev/
loop4
```

RAID

```
mdadm --create /dev/md0 -level=1
--raid-devices=2 /dev/grupa/
wolumen1 /dev/grupa/wolumen2
```

Wysoka dostępność jest zdefiniowana poprzez:

1. Możliwość wymiany dysku bez wyłączania wolumenu z produkcji.
2. Rozbudowę pojemności wolumenu bez wyłączania wolumenu z produkcji.
3. Sposób reakcji systemu na zdarzenia krytyczne w wolumenie.
4. Możliwość dokonywania kopii zapasowych bez wyłączania wolumenu z produkcji.

W drugiej części poznamy szczegóły techniczne obu konfiguracji, znaczenie poszczególnych parametrów i przełączników, możliwe symulacje awarii i symulacje wysokiej dostępności. ■

AUTOR

Cezary Gajdziński od wielu lat zajmuje się problemami związanymi z bezpieczeństwem systemów linuxowych i unixowych, oraz bezpieczną wymianą informacji. Kontakt: cezary@amc.homeunix.net.

W Internecie

[1] Strona domowa projektu `LVM`
<http://sourceware.org/lvm2/>

[2] Strona domowa projektu `mdadm`
<http://cgi.cse.unsw.edu.au/~ne-ilib/mdadm/>

[3] Dokumentacja `losetup` <http://linuxreviews.org/man/losetup/>