



Použití šifrovaných disků v Linuxu

Milan Brož
mbroz@redhat.com

LinuxAlt 2008, 1.listopadu 2008, Brno

Proč šifrovat

... ještě vám nikdy neukradli notebook?

Co zneužitelného může obsahovat disk

pracovní údaje

- účetnictví, smlouvy, plány, ...
- lékařské údaje, právníkové dokumenty, ...
- údaje podléhající zákonu o ochraně osobních dat

soukromá data

- hesla, přístupové kódy, privátní klíče...
 - elektronická pošta
 - ~ zneužitelná data (výpisy z účtu apod.)
-
- cena za bezpečnost (výkon, rychlost, ...)
 - systém je bezpečný jako jeho nejslabší článek
 - ... otevřené řešení nezatížené patenty

Výběr vrstvy pro šifrování...

- **Speciální HW**
- **Přímo v aplikaci**

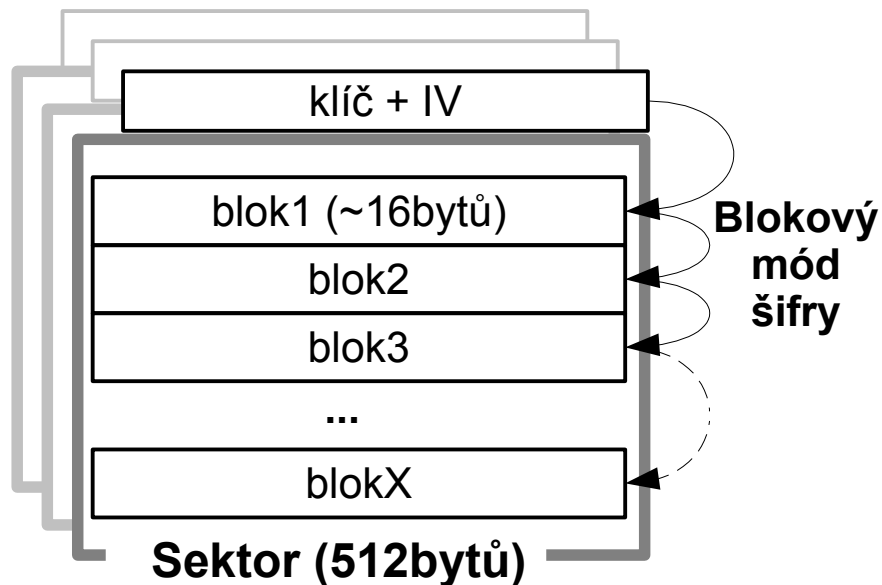
- **Souborový systém: (EncFS, eCryptfs, ...)**
 - šifrování na úrovni filesystemu
 - metadata algoritmu v souboru nebo adresáři, kopírují se s daty
 - některá metadata filesystemu nejsou šifrovaná
 - selektivní výběr, co se šifruje (které soubory nebo adresáře)

- **Blokové zařízení: (dm-crypt, truecrypt, loop-aes, ...)**
 - šifrování na úrovni sektorů
 - zejména ochrana při vypnutém zařízení
 - transparentní pod filesystemem
 - v kombinaci s volume managementem
 - swap partition

Šifrování disku

▪ Blokové zařízení – jednotkou je sektor

- v Linuxu sektor vždy 512 bytů, náhodný přístup
- **sektory jsou šifrovány nezávisle na sobě**
- předpokládá se, že disk před zápisem obsahuje náhodná data
- algoritmus používá **bloky** \leq sektor
 - blok je obvykle 128bitů (16 bytů)
 - poslední blok je stejně veliký jako ostatní (zjednodušení)



- IV – inicializační vektor
 - Pro každý blok zvlášť
 - odvozen od čísla bloku
- Granularita – blok vs sektor

Blokové šifrovací algoritmy (příklady)

aes-cbc-essiv:sha256, aes-xts-plain, ...

- **Algoritmus** (definuje velikost klíče)
AES, twofish, serpent, ...

- **Mód blokové šifry**
CBC (cipher block chaining)
LRW (Liskov,Rivest,Wagner), v kernelu od 2.6.20
XTS (XEX-TCB-CTS), v kernelu od 2.6.24, vhodný pro <1 TB dat

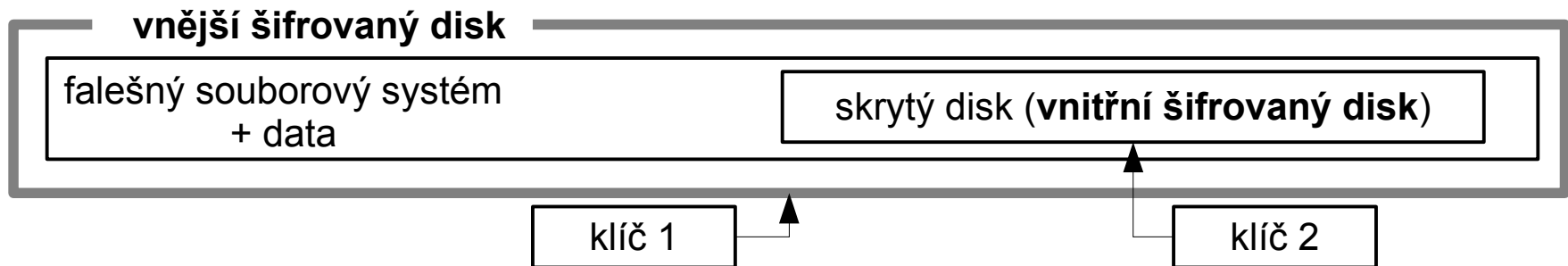
- Wide módy (blok=sektor) se kvůli nutnosti zpracování bloku minimálně ve dvou průchodech nepoužívají. (Navíc standardem doporučovaný EME je patentovaný.)

- **IV – inicializační vektor**
 - **plain** - číslo sektoru (zarovnané na potřebnou velikost)
 - **ESSIV** – Encrypted Salt-Sector, závisí na šifrovaném hashi klíče

Hidden volume (~skrytý disk)

- **plausible deniability**

- schopnost „uvěřitelně“ popřít, že jsou na disku nějaká data
- data jsou ukrytá v „nepoužívaném“ prostoru, ke kterému je nutný další klíč, šifrovaná data nelze rozeznat od „šumu“
- šifrovaná data nemají viditelnou hlavičku
(zda je disk šifrovaný nelze určit jinak, než že se jej systém pokusí dešifrovat a najde signaturu)



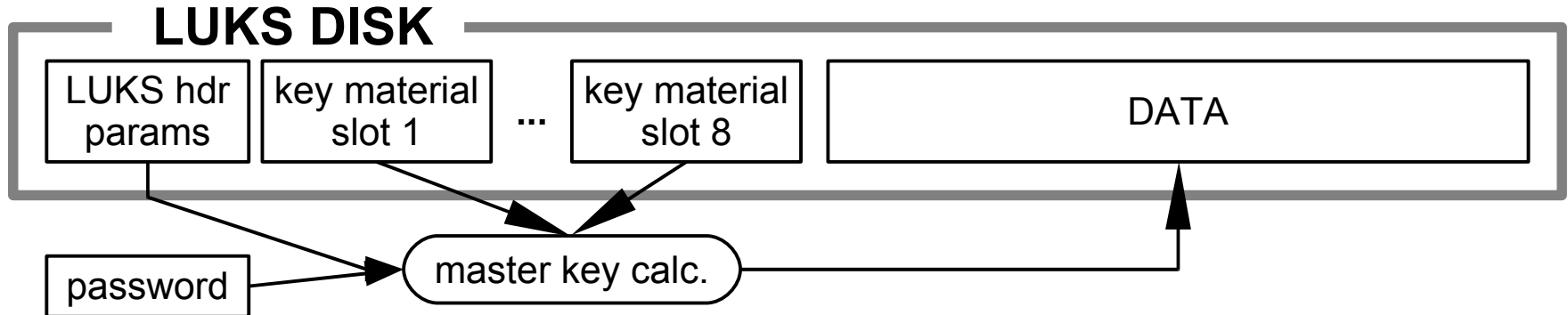
- tento koncept používá například Truecrypt
- pomocí device-mapperu lze vytvořit podobné schéma
- prosakování dat

dm-crypt + cryptsetup

- kernel device-mapper **crypt target**
 - umožňuje vytvořit transparentní šifrované blokové zařízení (a nad ním používat libovolný filesystem)
- **Používá kernel cryptoAPI**
 - **HW support** (VIA Padlock, Geode, ...)
- **cryptsetup[-luks]** – nástroj pro konfiguraci dm-cryptu
 - implementuje **LUKS**
- + grafické utility, integrace do udev, HAL

LUKS (Linux Unified Key Setup)

- de facto standard pro konfiguraci šifrování disku v Linuxu
- přenositelný, podporovaný i jinými OS (FreeOTFE.org)
- **více hesel (passphrases)** odemyká silný **master key**, PBKDF2
- **změna** (zneplatnění) hesla
 - bez nutnosti přešifrovat celý disk



AF-splitter – anti-forenzní ochrana (proti obnovení hesla z realokovaných sektorů)

- firmware disku provede realokaci sektoru, data mohou být stále na disku ve skryté oblasti
- AF-splitter minimalizuje možnost z této oblasti obnovit citlivá data

cryptsetup

LUKS příkazy: Format, Open, Close, AddKey, KillSlot, Dump
(create, remove, status – přímé nastavení dm-cryptu bez LUKS)

luksFormat – vytvoření hlavičky LUKS

```
cryptsetup [-c serpent-cbc-essiv:sha256 -s 256] luksFormat $DEV
```

luksOpen - zpřístupnění obsahu disku

```
cryptsetup luksOpen $DEV $CRYPT_DEV_NAME
```

luksClose – zrušení mapování

```
cryptsetup luksClose $CRYPT_DEV_NAME
```

luksAddKey, luksKillSlot, (luksRemoveKey) – manipulace s keysloty

luksDump – výpis informací o parametrech, například

```
Cipher name:      serpent
Cipher mode:     cbc-essiv:sha256
Payload offset:  2056
UUID: 09714b0c-9a70-4652-86d2-7300b755eb4f
```

cryptsetup

Příklad nastavení – záleží na distribuci

/etc/crypttab:

```
#<tgt.dev> <src.dev> <key file> <options>
```

- jednoduchý disk, LUKS (není třeba další parametry)

```
$CDISK /dev/sdX none retry=5
```

- swap na LVM oddílu, bez LUKS, master key je náhodný klíč (každý boot jiný)

- pozor na správnou inicializaci RNG při bootu - seed

```
$CSWAP /dev/VG/lv /dev/urandom swap,cipher=aes-cbc-essiv:sha256
```

/etc/fstab:

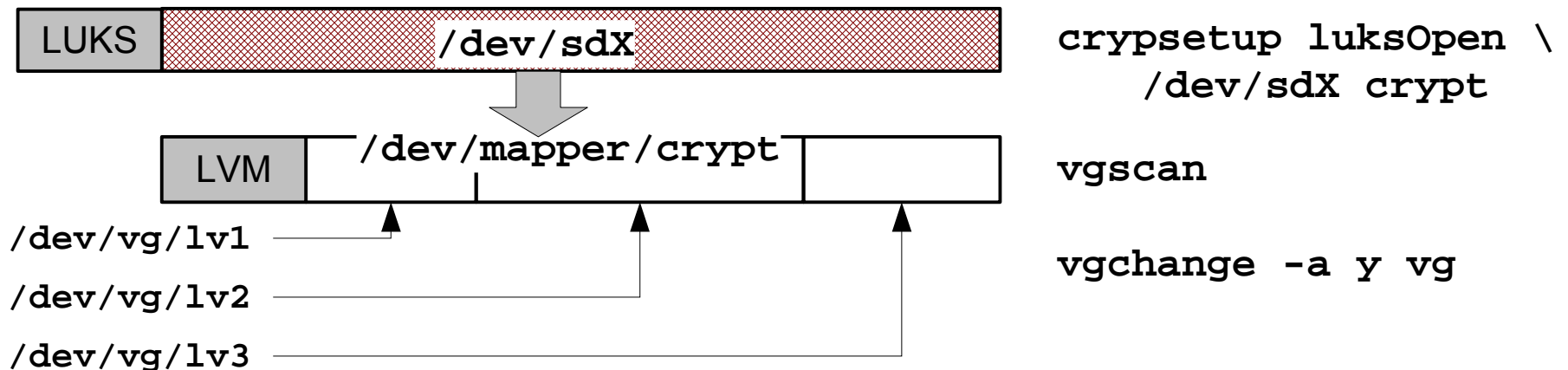
```
/dev/mapper/$CDISK $MNT auto defaults 0 0  
/dev/mapper/$CSWAP swap swap defaults 0 0
```

cryptsetup

- **Zvětšení, zmenšení disku**
 - v hlavičce není zapsána velikost disku,
stačí změnit velikost zařízení
 - ... a správně změnit velikost souborového systému nad ním
 - případně zapsat náhodná data do zvětšené části
 - `cryptsetup resize` – nahraje novou velikost online
- **Změna algoritmu, master key, ...**
 - Zatím asi nejbezpečnější metoda pomocí kopie na jiný disk
- v budoucnu pravděpodobně integrace s LVM s rozšířením funkcí (správa klíčů, integrace do správy systému)

Cryptsetup + LVM (Logical Volume Management)

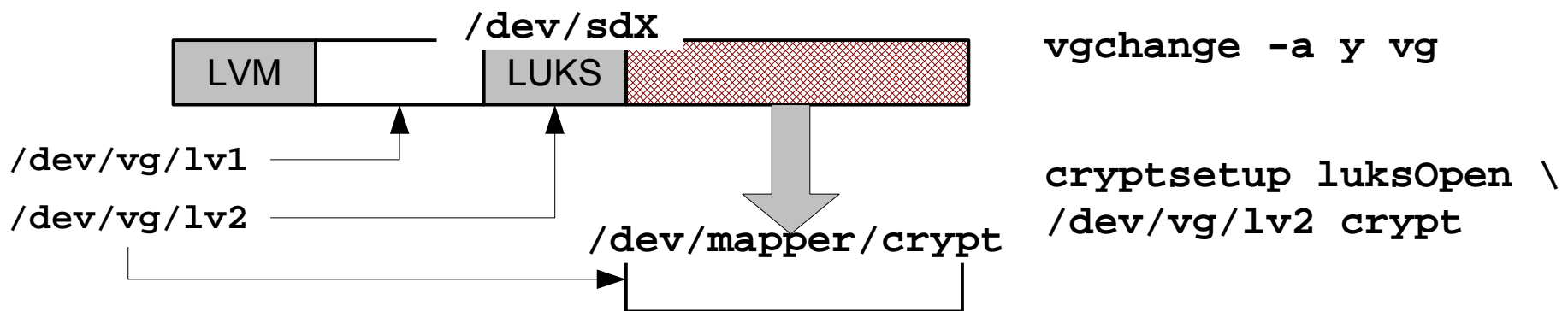
- LVM metadata – redundance, historie změn, archivace
- LUKS metadata – není žádoucí udržovat starou hlavičku
- **LVM nad šifrovaným diskem**
 - PV je šifrovaným diskem
 - Šifrována jsou i LVM metadata



Cryptsetup + LVM (Logical Volume Management)

▪ šifrovaný LVM oddíl

- jen některé logické oddíly mohou být šifrované



- všechny postupy lze použít i pro systémový disk
- nutnost použití ramdisku (initrd)
- musí být dostupné všechny moduly (disk, dm-crypt, crypto)

Truecrypt

- **Jeden z nejznámějších projektů**

- Windows, Linux, MacOS
- verze 6 používá na Linuxu pro šifrování dm-crypt jako backend
- stále závislost na FUSE (skrytý disk, staré kontejnery)
- na windows včetně skrytého OS a bootu
- GUI

- Používá AES, Serpent nebo Twofish v XTS módu

- OpenSource, ale nikoliv GPL

- pokud možno v Linuxu použijte dm-crypt + cryptsetup

Záloha, obnova dat

- **Není klíč nebo LUKS hlavička – ztráta všech dat**
- **Error diffusion**
 - vadný bit v RAM - ztráta minimálně jednoho celého bloku dat
 - chyby HW mají obecně mnohem horší následky
- **Záchrana dat**
 - Řešení je (jako vždy:-) obnova ze zálohy obsahu
 - LUKS
 - Záloha mapovací tabulky, master key
 - `dmsetup table --showkeys`
 - Se znalostí master key není třeba žádné heslo!
 - **Záloha LUKS hlavičky**
 - `dd if=/dev/<dev> of=backup.img bs=512 count=NUM`
 - NUM - počet sektorů v Payload Offset z luksDump
 - Znalost hesla alespoň k jednomu klíči v dané hlavičce

A jak je to rychlé?

- šifrování – mnoho závislostí na podmínkách
 - zatížení procesoru, jaký typ IO, optimalizace
 - IO se šifrují sekvenčně
 - jakýkoliv file sync může čekat i na ostatní data
 - šifrování zpracovává speciální proces, latence
 - multicore/SMP support?
- dnešní procesory jsou rychlé
 - propustnost pro sekvenční čtení bývá podobná jako bez šifrování
 - **HW akcelerace** – do budoucna čím dál více
 - původně pro IPsec
 - transparentní, kernel drivers
 - asynchronní mód

odkazy

- http://en.wikipedia.org/wiki/Disk_encryption_theory
- New methods in hard disk encryption (ale už ne úplně new:-)
<http://clemens.endorphin.org/nmihde/nmihde-A4-os.pdf>
- LRW: <http://grouper.ieee.org/groups/1619/email/pdf00017.pdf>
- XTS: <http://grouper.ieee.org/groups/1619/email/pdf00086.pdf>
- cryptsetup: <http://code.google.com/p/cryptsetup/>
- LUKS: <http://luks.endorphin.org/spec>
- dm-crypt mailing list
 - <http://news.gmane.org/gmane.linux.kernel.device-mapper.dm-crypt>
- Truecrypt: <http://www.truecrypt.org/>