VeraCrypt

Położenie: (nie dotyczy)

© 3bird Projects 2016, http://informatyka.3bird.pl

Informacje

Następca *TrueCrypta* (częściowo na licencji *TrueCrypta*, a częściowo na *Apache License*), posiadający wersje na *Windows* / *Linux* / *OS X*. Producentem jest: *http://www.idrix.fr*. Podobnie jak poprzednik, potrafi szyfrować całe dyski, jak i pojedyczne woluminy (kontenery w postaci pliku, które można montować tak jak dyski).

Instalacja

Program ściągamy ze strony: *http://sourceforge.net/projects/veracrypt/*. Wewnątrz spakowanego pliku *.tar.bz2 są wersje instalacyjne na różne systemy. Należy uruchamiać je tak jak skrypty:
./veracrypt-1.18-setup-gui-x64

Do działania potrzebuje bibliotek *FUSE*, oraz narzędzi do „*device mapper*”. W szczególności należy:
1. Uaktywnić składniki jądra:

Device Drivers →
[*] Block devices
  <> Loopback device support* (moduł *loop*; u mnie jest na stałe w jądrze)
[*] Multiple devices driver support (RAID and LVM):
  <M> Device mapper support (moduł *dm_mod*)
    <M> Crypt target support (moduł *dm_crypt*)
File systems →
  <M> The Extended 4 (ext4) filesystem (moduł *ext4*)
  <M> FUSE (Filesystem in Userspace) support (moduł *fuse*)
Cryptographic API →
  <M> XTS support (moduł *xts*)
  oraz inne metody szyfrowania, które chcemy używać (jako moduły)

2. W moim przypadku *udev* nie ładował automatycznie modułu *fuse*, więc dałem wpis w pliku */etc/conf.d/modules* → *fuse* (moduł *dm_crypt* ładowany jest automatycznie, gdy jest potrzebny).

3. Oprócz modułu FUSE w jądrze, należy zainstalować pakiet FUSE:
emerge sys-fs/fuse

4. Zamiast starego „*device-mapper*” („*dmsetup*”) należy zainstalować „*lvm2*” (*Logical Volume Manager*), które zawiera w sobie te narzędzia:
emerge lvm2
rc-update add lvm boot

Tworzenie kontenerów

Uwaga: Na chwilę obecną, całość operacji należy wykonywać na koncie roota, bo inaczej program krzyczy, że nie ma „*sudo*”... a nie ma, bo nie chcę mieć... i tyle! ;-) Być może uda się ten problem rozwiązać w przyszłości.

1. *Create Volume (Never Saved History)...*

2. *Create an encrypted file container...*

3. Decydujemy: „*Standard VeraCrypt Volume*” czy „*Hidden VeraCrypt Volume*” (gdy ktoś zmusi nas do zdradzenia hasła, podajemy mu hasło zapasowe, które otwiera inny fikcyjny zasób; na tym fikcyjnym zasobie umieszczamy pozornie ważne dane). *Hidden Volume* to kontener, który znajduje się wewnątrz głównego kontenera. Nie jest możliwy zapis na ukrytej partycji, dopóki główna partycja (*outer volume*) jest zamontowana.

4. *Volume location* / *Select file...* (wybieramy, a tak naprawdę **tworzymy** pusty kontener... nie wskazujemy na istniejący plik, bo zostanie nadpisany!).

5. Wybieramy metodę szyfrowania kontenera głównego / jawnego (*Outer Volume Encryption Options*): AES-256, SHA-512.

6. *Outer Volume Size* (rozmiar kontenera głównego / jawnego): ustanawiamy maksymalny rozmiar kontenera. Uwaga: Kontener jawny tworzony jest jako VFAT (program nawet o to nie pyta, sam decyduje) w sytuacji, jeśli zadeklarowaliśmy stworzenie także partycji ukrytej. Nie pozwala wtedy także na większy rozmiar całości niż 2048GB (przy czym sformatowanie dysku za pomocą polecenia „*mkreiserfs -b 4096 /dev/sdb*" - nie zmienia sytuacji; prawdopodobnie chodzi o to, że program nie wspiera GPT, lecz jedynie MBR, który ma limit do 2TB). Jeśli jednak tworzymy tylko i wyłącznie partycję jawną – nie ma ograniczeń rozmiaru, a także możemy wybrać inny system plików, np. EXT4.

7. Ustalamy hasło dostępu do kontenera głównego / jawnego (maksymalnie 64 znaki). Możemy także użyć klucza w postaci plików: „*Use keyfiles*". Kluczem może być także zwykły nieszyfrowany plik (ważne, aby nie zmieniał zawartości), np. jpg, mp3, itp. Opcja PIM (*Personal Iterations Multiplier*) umożliwia zdefiniowanie wielokrotnej iteracji i stanowi dodatkowe zabezpieczenie (oprócz hasła).

8. *Outer Volume Contents* – kontener jawny został utworzony i zamontowany w */mnt/veracrypt1* lub */media/veracrypt1*. Kopiujemy na niego pozornie tajne i ważne dane (na takie muszą wyglądać). Następnie program automatycznie ustawi rozmiar ukrytego kontenera na maksymalny możliwy. Uwaga: Po stworzeniu partycji ukrytej, nie zmieniamy już zawartości partycji jawnej, aby nie uszkodzić danych na partycji ukrytej.

9. *Hidden Volume Encryption Option* – ustalany algorytmy szyfrowania dla ukrytego kontenera: AES, SHA-512.

10. *Hidden Volume Size* – ustalamy rozmiar kontenera ukrytego.

11. *Hidden Volume Password* – ustalamy hasło (inne niż w przypadku kontenera jawnego) i klucz.

12. *Format Options* – niestety, nie ma *ReiserFS*... więc wybieramy *EXT4*.

13. *Cross-Platform Support* – wybieramy montowanie tylko na Linuksie.

14. *Hidden Volume Format* – ruszamy myszką w celu zwiększenia chaosu podczas szyfrowania, po czym formatujemy. Formatowanie może trwać nawet kilka godzin.

Montowanie kontenerów

1. W głównym oknie programu montujemy utworzony kontener: wybieramy logiczne oznaczenie dysku, na którym chcemy dokonać montowania (np. 7), a następnie klikamy w „*Select file...*", wybieramy nasz kontener, montujemy go, wprowadzamy hasło i/lub klucze (*keyfiles*), podajemy typ szyfrowania.

2. Zależnie od podanego hasła, zostanie zamontowany albo kontener jawny, albo ukryty (jako */mnt/veracrypt7*). Jeśli kontener zostanie zamontowany w punkcie */media/veracrypt*, należy po odmontowaniu usunąć folder */media*, a od tej pory będzie montował się w */mnt*.

3. Nadajemy zamontowanemu zasobowi pełne prawa do zapisu przez usera:
 # chown -R franek.users /mnt/veracrypt7/

4. W przypadku dużych kontenerów (powyżej 1TB) szybciej jest kopiować zawartość kontenera na inny dysk niż kopiować sam kontener (może to zająć nawet kilkanaście godzin).

Uwagi:

- Pakiet „*device-mapper*” został zastąpiony pakietem „*lvm2*”.
- Plik konfiguracyjny: */home/user/.config/VeraCrypt/Configuration.xml*.

Ostatnia aktualizacja: 17 września 2016.