

Linux Mint 19.3 Cinnamon

Instalacja + Full Disk Encryption + LVM + LUKS + UEFI boot mode

Tutorial – LM19.3 FDE, Rev. 0.1

Przemysław Fornalkiewicz

e-mail: przemyslaw.fornalkiewicz@gmail.com

Spis treści

Wstęp.....	3
1. Przygotowanie dysku twardego na instalację systemu operacyjnego.....	5
1.1. Sprawdzenie czy komputer działa w trybie UEFI.....	5
1.2. Przygotowanie środowiska pracy.....	5
1.3. Utworzenie partycji.....	6
1.4. Szyfrowanie partycji za pomocą LUKS.....	8
1.5. Tworzenie woluminów LVM.....	11
2. Przygotowanie dodatkowego dysku twardego.....	14
3. Instalacja systemu na przygotowanym dysku twardym.....	16
3.1. Modyfikacja pliku źródłowego w celu uniknięcia błędu GRUB podczas instalacji.....	16
3.2. Uruchomienie instalatora systemu z terminala.....	18
3.3. Kontynuowanie instalacji w trybie graficznym.....	18
4. Konfiguracja systemu po instalacji.....	24
4.1. Zamontowanie wymaganych katalogów systemu w trybie Live CD.....	24
4.2. Instalacja brakujących pakietów GRUB dla systemu UEFI.....	24
4.3. Automatyczne montowanie zaszyfrowanych woluminów przy starcie systemu.....	24
4.4. Aktualizacja konfiguracji GRUB.....	28
4.5. Dodatkowa konfiguracja GRUB.....	29
4.6. Zakończenie konfiguracji systemu.....	30
4.7. Backup nagłówków kontenera LUKS.....	31
5. Montowanie woluminów LVM w katalogu użytkownika.....	32
5.1. Aktualizacja wpisów w fstab.....	32
5.2. Zmiana uprawnień dostępu dla montowanych partycji.....	33
Bibliografia.....	35

Wstęp

Poniższy tutorial dotyczy sposobu instalacji systemu operacyjnego **Linux Mint 19.3 Cinnamon Tricia 64-bit** na komputerze klasy PC, wyposażonym w tryb bootowania **UEFI**. System jest instalowany na całkowicie zaszyfrowanym dysku twardym, włącznie z partycją **/boot**. Jest to instalacja z pełnym szyfrowaniem dysku (**Full Disk Encryption**).

Dodatkowo dysk będzie podzielony na partycje za pomocą **LVM (Logical Volume Manager)**, a wolumin LVM będzie utworzony na jednej partycji, szyfrowanej za pomocą **LUKS (Linux Unified Key Setup)**, w woluminie LVM zostaną utworzone partycje o punkcie montowania „/” oraz „/home”. Jest to instalacja typu **LVM na LUKS**.

W komputerze dostępny jest drugi dysk, na którym zostanie utworzona jedna partycja, zaszyfrowana za pomocą LUKS, a na niej zostanie utworzony drugi wolumin LVM, który następnie zostanie podzielony na partycje. Partycje zostaną podmontowane w katalogu `/home/user` jako lokalne katalogi użytkownika: Dokumenty, Obrazy, Muzyka, Pobrane, Wideo. Aby użytkownik miał dostęp do tak przygotowanych partycji, zmienione zostaną domyślne parametry montowania oraz nadane zostaną odpowiednie prawa dostępu do montowanych woluminów.

Podczas uruchamiania systemu możliwe jest montowanie każdej partycji LUKS niezależnie, wtedy konieczne będzie wpisanie hasła trzykrotnie. 1 – dla partycji boot, 2 – dla partycji LUKS pierwszego dysku, 3 - dla partycji LUKS drugiego dysku.

Druga możliwość to wpisanie hasła tylko jeden raz – dla partycji `/boot`. W takim rozwiązaniu konieczne jest wygenerowanie kluczy do pozostałych partycji LUKS, co umożliwi ich automatyczne montowanie po odszyfrowaniu partycji `/boot`. W poniższym tutorialu, aby przedstawić pełen wachlarz możliwości, opisano obydwie metody. Docelowo konfigurację tą można zmienić, wykonując dla wybranego dysku odpowiednie kroki opisane w dalszej części tekstu.

Zalety i wady instalacji typu LVM na LUKS:

Zalety:

- Proste partycjonowanie ze znajomością LVM.
- Tylko jeden klucz wymagany do odblokowania wszystkich woluminów.
- Układ woluminów nie jest przezroczysty po zablokowaniu.

Wady:

- LVM dodaje dodatkową warstwę odwzorowania i hook.
- Mniej przydatne, jeśli pojedyncza partycja powinna otrzymać oddzielny klucz.

1. Przygotowanie dysku twardego na instalację systemu operacyjnego

Aby rozpocząć instalację systemu, należy uruchomić system Live CD, nagrany wcześniej na płytę DVD lub pendrive. Po przygotowaniu systemu i uruchomieniu go w wersji live, można rozpocząć przygotowanie dysku twardego. W tym celu należy uruchomić terminal i wydać szereg poleceń. Część poleceń dotyczących konfiguracji dysku można wykonać również w sposób graficzny, na przykład za pomocą GParted i Logical Volume Management (`system-config-lvm`) lub KVM (kvm). Jednak w tym tutorialu przedstawiona jest wersja konsolowa konfiguracji dysku.

Aby mieć pewność, że system zostanie uruchomiony w trybie UEFI, należy sprawdzić ustawienia płyty głównej, która może wspierać rozruch zarówno UEFI jak i BIOS. W ustawieniach płyty głównej należy wybrać odpowiednią opcję rozruchu. Po uruchomieniu Live CD można sprawdzić czy system, na pewno został uruchomiony w trybie UEFI.

1.1. Sprawdzenie czy komputer działa w trybie UEFI

```
$ mount | grep efivars
```

```
efivarfs on /sys/firmware/efi/efivars type efivarfs  
(rw,nosuid,nodev,noexec,relatime)
```

Obecność systemu plików **efivarfs** oznacza, że system został uruchomiony w trybie UEFI:

1.2. Przygotowanie środowiska pracy

a) Włączenie wydawania komend jako administrator (root):

```
$ sudo -i
```

b) Ustawienie zmiennych środowiskowych

Aby możliwe było skopiowanie wszystkich poleceń z tutoriala wprost do terminala, ustawione zostaną zmienne systemowe, zawierające identyfikator dysku, który będzie konfigurowany. Konfigurację dysków w systemie można sprawdzić za pomocą poleceń, np. `fdisk -l` lub `lsblk`. Przeważnie będą to identyfikatory:

- Dysk klasyczny (SSD, HDD)
export DEV="/dev/sda"

- Dysk NVME (M.2):
export DEV="/dev/nvme0n1"

Dodatkowo należy ustawić zmienną środowiskową dla mapowania zaszyfrowanych partycji, która pomija część „/dev/”:

export DM="\${DEV##*/}"

1.3. Utworzenie partycji

a) Usunięcie zawartości dysku

Jeżeli na dysku istnieją jakieś partycje i nie są potrzebne, można je skasować poniższym poleceniem, jeżeli dysk nie ma być czyszczony całkowicie, **NIE należy wykonywać** poniższego polecenia:

sgdisk --zap-all \$DEV

GPT data structures destroyed! You may now partition the disk using fdisk or other utilities.

b) Utworzenie nowych partycji

Wykonanie poniższych komend spowoduje utworzenie nagłówka dysku w formacie GPT (zgodnego z UEFI).

Sprawdzenie czy dysk jest pusty

sgdisk --print \$DEV

Creating new GPT entries.

Disk /dev/sda: 62914560 sectors, 30.0 GiB

Model: QEMU HARDDISK Sector size (logical/physical): 512/512 bytes

Disk identifier (GUID): 91ADE06D-BA18-4675-9986-FA8E959664A5

Partition table holds up to 128 entries

Main partition table begins at sector 2 and ends at sector 33

First usable sector is 34, last usable sector is 62914526

Partitions will be aligned on 2048-sector boundaries

Total free space is 62914493 sectors (30.0 GiB)

Number	Start (sector)	End (sector)	Size	Code	Name
--------	----------------	--------------	------	------	------

Utworzone zostaną trzy partycje:

- EFI-ESP (rozruchowa) - o wielkości 300 MB

- `/boot` – o wielkości 1024 MB (1 GB)
- LVM dla LUKS – pozostała dostępna przestrzeń na dysku, dla montowania „/” (root) oraz „/home”, które będą zawarte w szyfrowanym woluminie LVM

Tworzenie nowych partycji

Składnia: `sgdisk --new=<partition_number>:<start>:<end>`
gdzie **start** i **end** mogą być wartościami względnymi, a gdy mają wartość zero (0) przyjmują odpowiednio najniższą lub najwyższą możliwą wartość.

```
# sgdisk --new=1:0:+300M $DEV
# sgdisk --new=2:0:+1024M $DEV
# sgdisk --new=5:0:0 $DEV
```

Ostatnia partycja ma przypisany numer 5. Wynika to z historycznego podziału partycji, gdy na dysku mogły być maksymalnie 4 partycje podstawowe. Aby zwiększyć liczbę partycji na dysku, tworzą partycję rozszerzoną, a w niej partycje logiczne. Taka partycja miała nadawany numer 5, nawet jeżeli na dysku nie istniały 4 podstawowe partycje.

Nadanie flag i nazw partycjom:

Kody partycji:

ef00 – boot, esp

8300 – partycja Linuksowa

```
# sgdisk --typecode=1:EF00 --typecode=2:8300 --
typecode=5:8300 $DEV
# sgdisk --change-name=1:EFI-SP --change-name=2:/boot --
change-name=5:rootfs $DEV
```

Możliwe jest utworzenie hybrydowego nagłówka partycji, która będzie zawierać metadane w formacie GPT (UEFI) oraz MBR (BIOS). Aby utworzyć taki nagłówek należy wydać polecenie:

```
# sgdisk --hybrid 1:2 $DEV
```

Gdzie 1 i 2 oznaczają partycje, które mają być brane pod uwagę przy rozruchu w trybie MBR. W niniejszym tutorialu to polecenie nie zostało jednak wykonane.

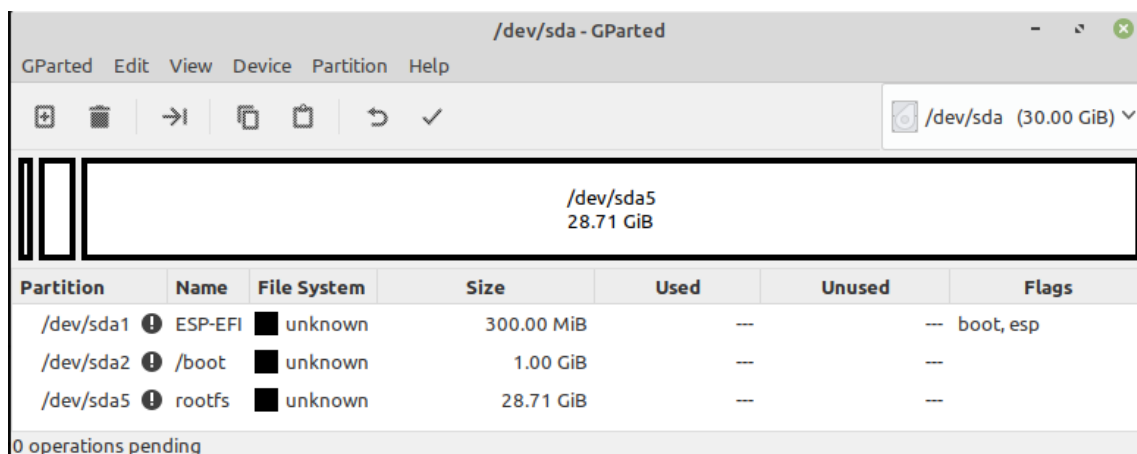
Sprawdzenie tablicy partycji po modyfikacjach:

`sgdisk --print $DEV`

```
Disk /dev/sda: 62914560 sectors, 30.0 GiB
Model: VBOX HARDDISK
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 773070B0-60C8-4512-A912-DEAF28A2708D
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 18874334
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)
```

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	616447	300.0 MiB	EF00	EFI-SP
2	616448	2713599	1024.0 MiB	8300	/boot
5	2713600	62914526	28.7 GiB	8300	rootfs

Można również sprawdzić poprawność wykonanych operacji za pomocą narzędzia GParted



Rys. 1 Widok podziału dysku po utworzeniu partycji

1.4. Szyfrowanie partycji za pomocą LUKS

Domyślny format *LUKS* (*Linux Unified Key Setup*) używany przez narzędzie `cryptsetup` zmienił się na przestrzeni czasu. **Linux Mint 19.3** domyślnie używa

wersji 1 (LUKS1), ale nowsze wersje systemu mogą domyślnie używać wersji 2 (LUKS2). Jednak GRUB obsługuje tylko wersję LUKS1 (w chwili tworzenia tego tutoriala). Więc musimy wyraźnie wyrażać się w poleceniach, których używamy, w przeciwnym razie GRUB nie będzie mógł zainstalować ani odblokować zaszyfrowanego urządzenia.

a) Sprawdzenie domyślnego formatu szyfrowania LUKS

```
# cryptsetup --help
```

```
Default compiled-in device cipher parameters:
```

```
loop-AES: aes, Key 256 bits
```

```
plain: aes-cbc-essiv:sha256, Key 256 bits, Password hashing: ripemd160
```

```
LUKS1: aes-xts-plain64, Key 256, LUKS header hashing: sha256, RNG: /dev/urandom
```

Na powyższym wpisie widać, że w systemie *Linux Mint 19.3*, domyślnie jest wykorzystywany format **LUKS1**. Gdyby tak nie było, konieczne byłoby jawne wykorzystanie tego formatu podczas szyfrowania partycji LUKS, poprzez dodanie opcji `--type=luks1` podczas wykonywania poleceń.

b) Szyfrowanie partycji za pomocą LUKS

Zaszyfrowane zostaną dwie partycje. Jedna dla folderu bootownia `/boot`. Druga, w której utworzony zostanie wolumin LVM dla partycji „/” oraz „/home”. Tylko niewielka partycja rozruchowa EFI-SP zostanie niezaszyfrowana. Na niej będzie się znajdował plik, który umożliwi odszyfrowanie pozostałych partycji, po podaniu hasła, które zostanie utworzone podczas szyfrowaniu wskazanych partycji.

Partycja „/boot”:

```
# cryptsetup luksFormat --type=luks1 ${DEV}2
```

```
WARNING!
```

```
=====
```

```
This will overwrite data on /dev/sda2 irrevocably.
```

```
Are you sure? (Type uppercase yes): YES
```

```
Enter passphrase for /dev/sda2:
```

```
Verify passphrase:
```

Partycja systemowa dla LVM (do montowania „/” oraz „/home”):

```
# cryptsetup luksFormat --type=luks1 ${DEV}5
```

WARNING!

=====

This will overwrite data on /dev/sda5 irrevocably.

Are you sure? (Type uppercase yes): YES

Enter passphrase for /dev/sda5:

Verify passphrase:

c) Montowanie zaszyfrowanych partycji

Nazwy LUKS_BOOT oraz \${DM}5_crypt (co odpowiada np. sda5_crypt) oznaczają nazwy, pod jakimi zaszyfrowane partycje będą dostępne po zamontowaniu. Szyfrowane partycje są montowane w lokalizacji /dev/mapper

Montowanie partycji „/boot”:

```
# cryptsetup open ${DEV}2 LUKS_BOOT
```

Enter passphrase for /dev/sda2:

Montowanie partycji systemowej:

```
# cryptsetup open ${DEV}5 ${DM}5_crypt
```

Enter passphrase for /dev/sda5:

Po zamontowaniu partycje zostaną zmapowane do odpowiednich katalogów, których nazwy zostały podane podczas otwierania zaszyfrowanych partycji:

/dev/mapper/LUKS_BOOT

/dev/mapper/sda5_crypt

d) Formatowanie partycji

WAŻNE!!! Ten krok musi zostać wykonany, w przeciwnym razie instalator wyłączy możliwość zapisu danych na tym urządzeniu.

Formatowanie partycji EFI-SP:

```
# mkfs.vfat -F 32 ${DEV}1
```

Mkfs.fat 4.1 (2017-01-24)

Partycja EFI-SP musi być sformatowana w systemie FAT, np. FAT16, FAT32.

Formatowanie partycji „/boot”:

```
# mkfs.ext4 /dev/mapper/LUKS_BOOT
```

Partycja systemowa zostanie sformatowana w momencie utworzenia woluminu LVM.

1.5. Tworzenie woluminów LVM

Aby utworzyć woluminy LVM, konieczne jest przypisanie fizycznego miejsca na dysku, w którym woluminy będą mogły być utworzone. W niniejszym tutorialu jest to szyfrowana partycja LUKS, zamontowana w katalogu `/dev/sda5_crypt`. Po udostępnieniu miejsca dla LVM, należy utworzyć tzw. grupę LVM, w której tworzone będą logiczne woluminy LVM.

Logiczne woluminy LVM są odpowiednikami partycji logicznych w urządzeniach, w których wykorzystuje się tradycyjne partycjonowanie (bez wykorzystywania LVM). Z kolei grupę LVM można w uproszczeniu określić jako partycję rozszerzoną, na której tworzy się partycje logiczne. Zaletą stosowania LVM jest łatwość oraz duża elastyczność modyfikowania przydziału miejsca na dyskach do odpowiednich partycji, w porównaniu z tradycyjnym podejściem do partycjonowania.

a) Oznaczenie zaszyfrowanej partycji LUKS jako urządzenia dla woluminu LVM

```
# pvcreate /dev/mapper/${DM}5_crypt
```

```
Physical volume "/dev/mapper/sda5_crypt" successfully created.
```

b) Utworzenie grupy , do której będą przypisane tworzone woluminy logiczne

```
# vgcreate systemvg /dev/mapper/${DM}5_crypt
```

```
Volume group "systemvg" successfully created
```

Wyrażenie **systemvg** jest nazwą grupy, jaka zostanie utworzona. Nazwa ta jest później wykorzystywana przy tworzeniu woluminów logicznych i ich montowaniu w systemie. Jest to odpowiednik numerowania partycji w stylu `/dev/sda5_crypt`. Zamiast zapamiętywania numerów partycji, wykorzystywane są nazwy utworzone w LVM.

Przyrostek **vg** pozwala łatwo rozpoznać, że nazwa odnosi się do grupy LVM.

c) Utworzenie woluminów logicznych

```
# lvcreate -n rootlv -L 20G systemvg
```

```
Logical volume "rootlv" created.
```

Pierwszy wpis tworzy wolumin logiczny (partycję) LVM o nazwie **rootlv**, o rozmiarze 20 GB, w grupie LVM o nazwie **systemvg**.

```
# lvcreate -n homelv -l 80%FREE systemvg
```

```
Logical volume "homelv" created.
```

Drugi wpis tworzy wolumin logiczny (partycję) LVM o nazwie **homelv**. Rozmiar partycji będzie zajmował 80% przestrzeni woluminu **systemvg**, jaki pozostał po utworzeniu partycji **rootlv**.

Powyższe woluminy (partycje) zostaną następnie wykorzystane jako punkty montowania katalogów „/” oraz „/home”.

Przyrostego **lv** w nazwach, pozwala łatwo rozpoznać, że chodzi o woluminy logiczne LVM.

Podczas tworzenia woluminów, można utworzyć dodatkowe partycje, np. SWAP, wtedy komenda miałaby na przykład postać:

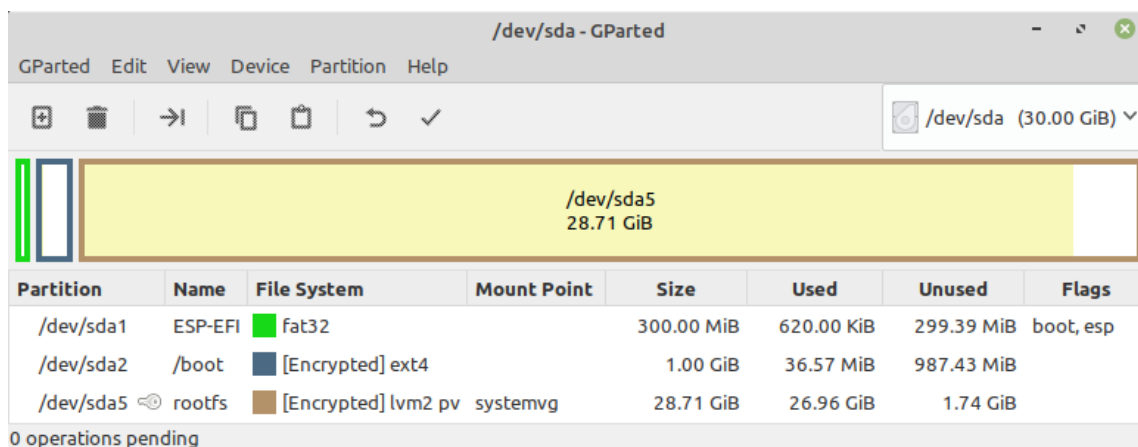
```
# lvcreate -n swap -L 4G systemvg
```

Taką partycję należałoby odpowiednio podmontować w momencie podmontowywania pozostałych partycji, podczas rozpoczęcia instalacji systemu. Komputery PC są obecnie wyposażane w wystarczającą ilość pamięci RAM oraz szybkie dyski SSD, które umożliwiają szybki start komputera. W związku z czym hibernacja systemu niekoniecznie musi być przydatna. Zatem partycja SWAP staje się coraz mniej potrzebna. W aktualnej konfiguracji ta partycja nie zostanie wykorzystana, nie została więc tworzona w tym zestawieniu.

d) Sprawdzenie poprawności wykonanych operacji

Po wykonaniu powyższych operacji można sprawdzić poprawność wykonania

poleceń. W programie GParted można podejrzeć aktualny stan partycji. Jednak widok niewiele powie, ponieważ konfiguracja LVM nie jest „przezroczysta” dla takich narzędzi jak GParted.



Rys. 2 Widoczny podział partycji w GParted po utworzeniu LVM

Aby sprawdzić aktualny podział dysku na partycje można wykonać polecenia:

vgs – informacja o istniejących grupach LVM w systemie

```
VG          #PV #LV #SN Attr   VSize   VFree
systemvg    1   2   0 wz--n- <28.70g 1.74g
```

lvs – informacja o istniejących woluminach logicznych LVM w systemie

```
LV      VG          Attr      LSize   Pool Origin Data%  Meta%  Move Log
homelv  systemvg  -wi-a----- <6.96g
rootlv  systemvg  -wi-a----- 20.00g
```

lsblk \${DEV} – aktualny podział dysku na partycje

```
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0   30G  0 disk
├─sda1                              8:1    0   300M  0 part
├─sda2                              8:2    0     1G  0 part
├─┬─LUKS_BOOT                      253:0    0 1022M  0 crypt
└─┬─sda5                          8:5    0  28.7G  0 part
   └─┬─sda5_crypt                  253:1    0  28.7G  0 crypt
      ├─systemvg-rootlv            253:2    0     20G  0 lvm
      └─systemvg-homelv            253:3    0      7G  0 lvm
```

2. Przygotowanie dodatkowego dysku twardego

Zgodnie z początkowymi założeniami scenariusza, w komputerze dostępny jest drugi dysk, który ma być w całości zaszyfrowany za pomocą LUKS, a utworzone na nim woluminy logiczne LVM będą podpięte jako foldery Dokumenty, Obrazy, Muzyka, Pobrane, Wideo w katalogu użytkownika /home/user.

Poszczególne kroki wykonywane dla drugiego dysku, są analogiczne jak tworzenie woluminów LVM dla dysku pierwszego, zostaną więc przedstawione same komendy, bez szerszych opisów:

Utworzenie zmiennych środowiskowych dla drugiego dysku:

```
# export DEV2="/dev/sdb"  
# export DM2="${DEV2}##*/"
```

Utworzenie partycji (która zajmie całą przestrzeń dysku)

```
# sgdisk --zap-all $DEV2  
# sgdisk --print $DEV2  
# sgdisk --new=1:0:0 $DEV2  
# sgdisk --typecode=1:8300 $DEV2  
# sgdisk --change-name=1:datafs $DEV2  
# sgdisk --print $DEV2
```

Zaszyfrowanie partycji za pomocą LUKS:

```
# cryptsetup luksFormat --type=luks1 ${DEV2}1  
# cryptsetup open ${DEV2}1 ${DM2}1_crypt
```

Podział szyfrowanej partycji na woluminy LVM:

```
# pvcreate /dev/mapper/${DM2}1_crypt  
# vgcreate datavg /dev/mapper/${DM2}1_crypt  
# lvcreate -n documentslv -L 10G datavg  
# lvcreate -n musiclv -L 30G datavg  
# lvcreate -n pictureslv -L 30G datavg  
# lvcreate -n videolv -L 50G datavg  
# lvcreate -n downloadlv -L 30G datavg
```

Sprawdzenie poprawności wykonanych operacji

```
# vgs  
# lvs  
# lsblk ${DEV2}
```

Formatowanie partycji:

```
# mkfs.ext4 /dev/mapper/datavg-documentslv  
# mkfs.ext4 /dev/mapper/datavg-musiclv  
# mkfs.ext4 /dev/mapper/datavg-pictureslv  
# mkfs.ext4 /dev/mapper/datavg-videolv  
# mkfs.ext4 /dev/mapper/datavg-downloadv
```

3. Instalacja systemu na przygotowanym dysku twardym

Po przygotowaniu dysku twardego, można rozpocząć instalację systemu operacyjnego. Standardowo odbywa się to, uruchamiając instalację poprzez uruchomienie instalatora `Install Linux Mint` na pulpicie dystrybucji Live CD. I tak można by postąpić w tym przypadku. Jednak jest duże prawdopodobieństwo, że pod koniec procesu instalacji, pojawi się okno z komunikatem z powodu błędu:

grub-efi-amd64-signed failed to install into /target/

Wystąpienie powyższego błędu spowoduje przerwanie procesu instalacji. W sieci można znaleźć wiele porad, które sugerują ominięcie tego błędu na przykład poprzez wyłączenie instalowania sterowników firm trzecich podczas instalacji lub odłączenie internetu od PC na czas instalacji. Jednak zdaniem autora nie o taką instalację systemu chodzi.

Można spróbować wykonać instalację uruchamiając instalator z pulpitu i liczyć na to, że błąd się nie pojawi lub od razu spróbować zapobiec pojawieniu się powyższego błędu. W tym celu należy wykonać kilka dodatkowych czynności przed rozpoczęciem oraz po zakończeniu instalacji. Poniższy opis zawiera informację jakie dodatkowe czynności należy wykonać.

3.1. Modyfikacja pliku źródłowego w celu uniknięcia błędu GRUB podczas instalacji

Aby uniknąć wspomnianego błędu, należy zmodyfikować jeden plik konfiguracyjny, który – jak można zobaczyć na Rys. 3 – odpowiada za załadowanie zaszyfrowanej partycji.

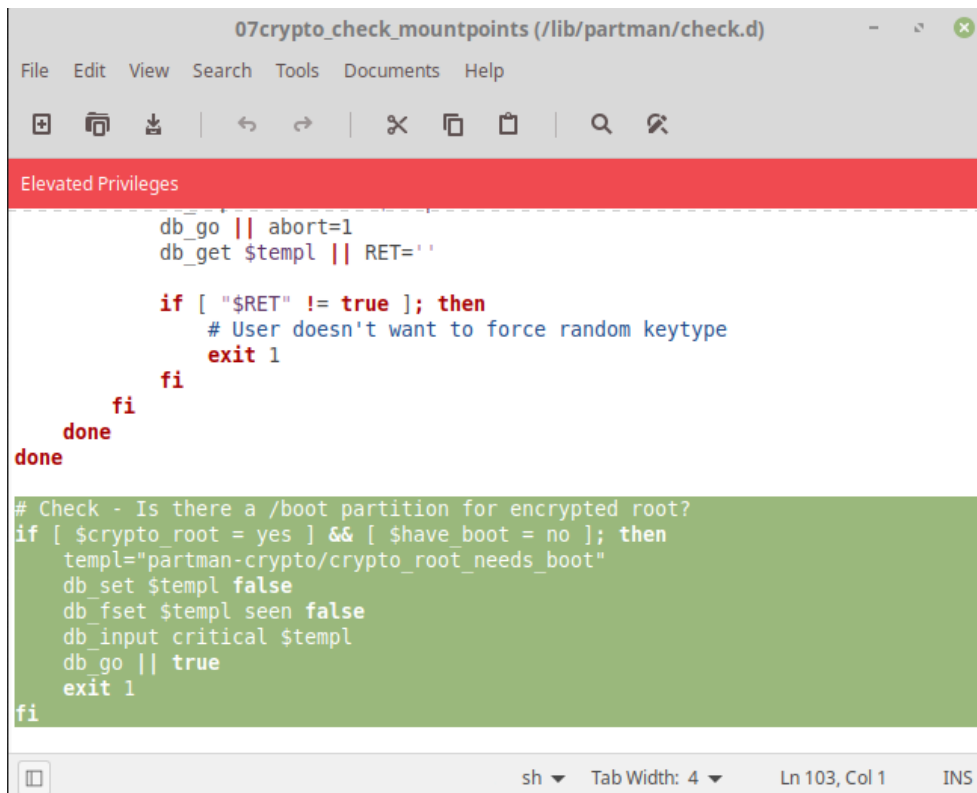
Uruchomienie pliku do edycji:

```
# xed /lib/partman/check.d/07crypto_check_mountpoints
```

Zamiast edytora `xed`, może to być inny dowolny edytor, np. `nano` lub graficzny `vi`.

Edycja pliku:

Należy przewinąć plik do końca i usunąć 9 ostatnich linii tekstu.



```
07crypto_check_mountpoints (/lib/partman/check.d)
File Edit View Search Tools Documents Help
Elevated Privileges

db_go || abort=1
db_get $templ || RET=''

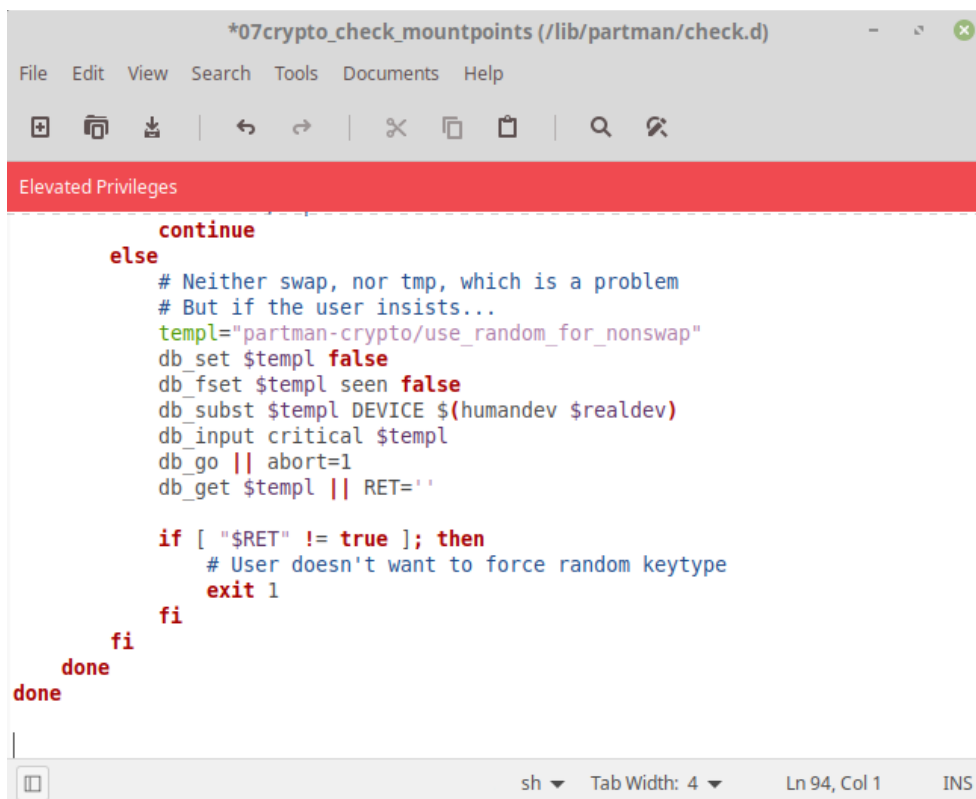
if [ "$RET" != true ]; then
    # User doesn't want to force random keytype
    exit 1
fi
done
done

# Check - Is there a /boot partition for encrypted root?
if [ $crypto_root = yes ] && [ $have_boot = no ]; then
    templ="partman-crypto/crypto_root_needs_boot"
    db_set $templ false
    db_fset $templ seen false
    db_input critical $templ
    db_go || true
    exit 1
fi

sh Tab Width: 4 Ln 103, Col 1 INS
```

Rys. 3 Fragment pliku konfiguracyjnego do usunięcia

Tym samym wyłączamy sprawdzanie, czy instalator Ubiquity nie instaluje dystrybucji, gdy partycja /boot znajduje się w zaszyfrowanym urządzeniu.



```
*07crypto_check_mountpoints (/lib/partman/check.d)
File Edit View Search Tools Documents Help
Elevated Privileges

        continue
    else
        # Neither swap, nor tmp, which is a problem
        # But if the user insists...
        templ="partman-crypto/use_random_for_nonswap"
        db_set $templ false
        db_fset $templ seen false
        db_subst $templ DEVICE $(humandev $realdev)
        db_input critical $templ
        db_go || abort=1
        db_get $templ || RET=''

        if [ "$RET" != true ]; then
            # User doesn't want to force random keytype
            exit 1
        fi
    fi
done
done

sh Tab Width: 4 Ln 94, Col 1 INS
```

Rys. 4 Plik konfiguracyjny po modyfikacji

Tak zmodyfikowany plik należy zapisać i zamknąć.

3.2. Uruchomienie instalatora systemu z terminala

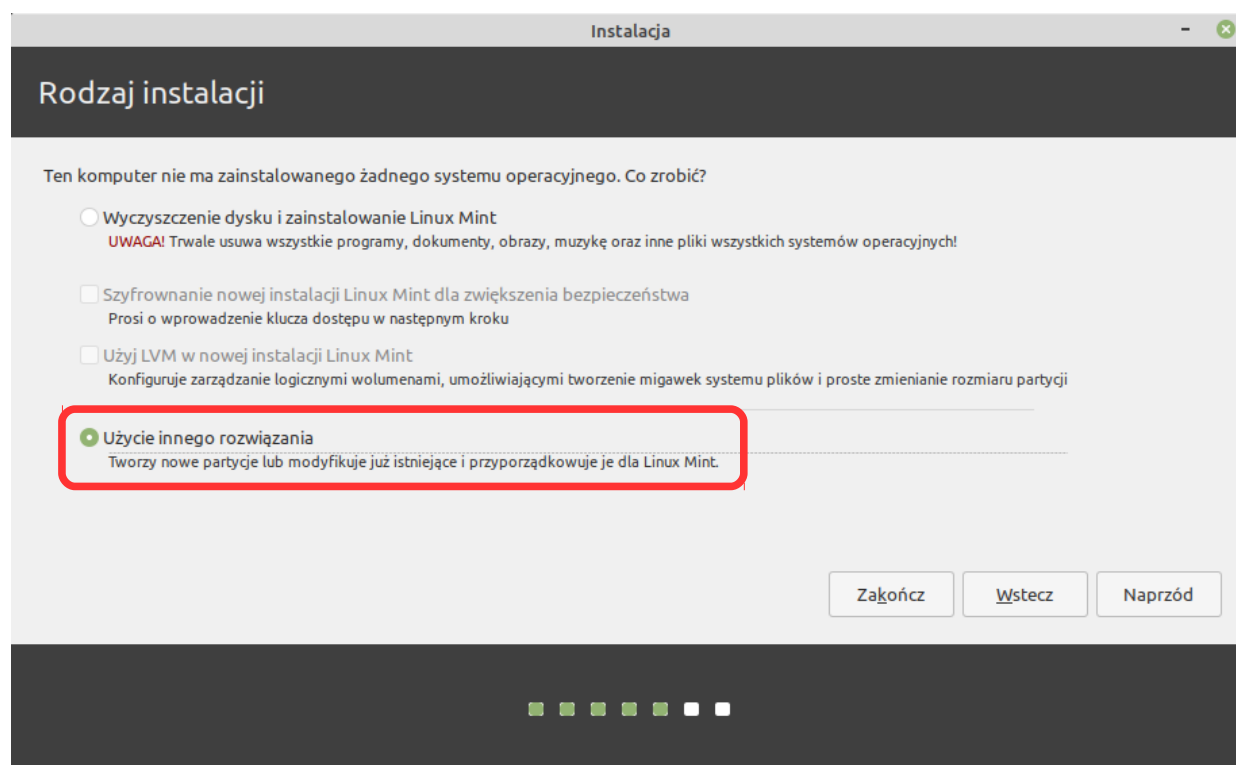
!!!Tą operację wyjątkowo należy wykonać jako zwykły użytkownik, a nie root!!!

```
$ sh -c 'ubiquity -b gtk-ui'&
```

Powyższe polecenie uruchomi instalator Ubiquity pomijając instalację modułu ładującego, a tym samym pozwala zakończyć proces Ubiquity bez błędów.

3.3. Kontynuowanie instalacji w trybie graficznym

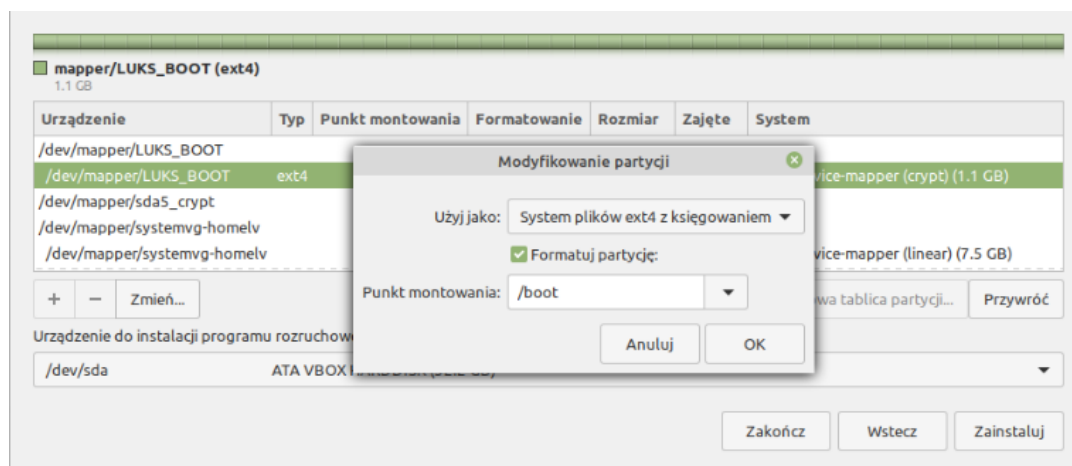
Po wykonaniu powyższego polecenia, zostanie uruchomiona standardowa procedura instalacji systemu w trybie graficznym. Instalacja przebiega standardowo do momentu wyświetlenia okna jak na Rys. 5:



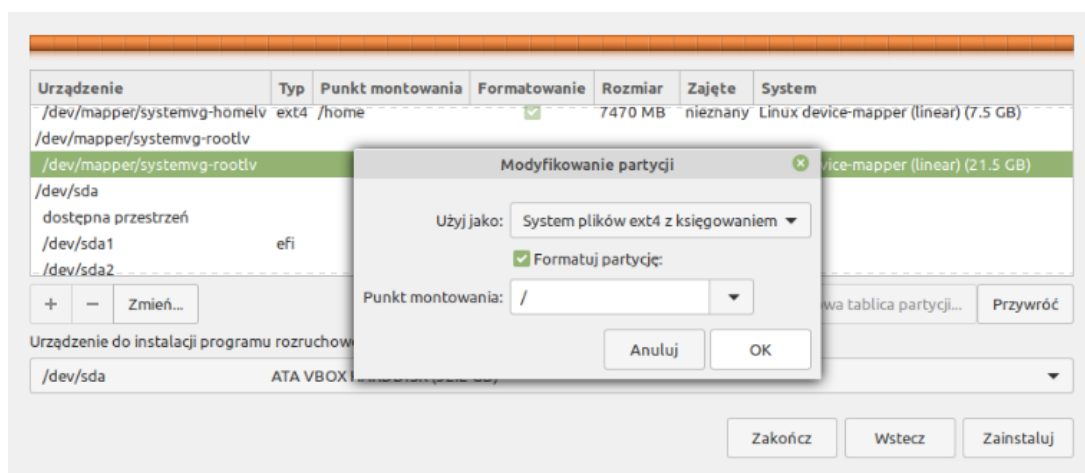
Rys. 5 Okno instalacji z wyborem ręcznego przypisania partycji na dysku

We wskazanym oknie instalatora należy wybrać opcję: Użycie innego rozwiązania, po czym wybrać przycisk Naprzód. Kolejne okno jakie się pojawi, powinno zawierać widok utworzonych wcześniej partycji.

Należy przypisać wcześniej utworzone partycje do odpowiednich punktów montownia, zaznaczając odpowiednią partycję i wybierając przycisk Zmień. Podczas wybierania punktu montownia, należy za każdym razem zaznaczyć opcję formatowania partycji w wybranym systemie plików. Będą to kolejno partycje /boot, /home oraz ”/”.



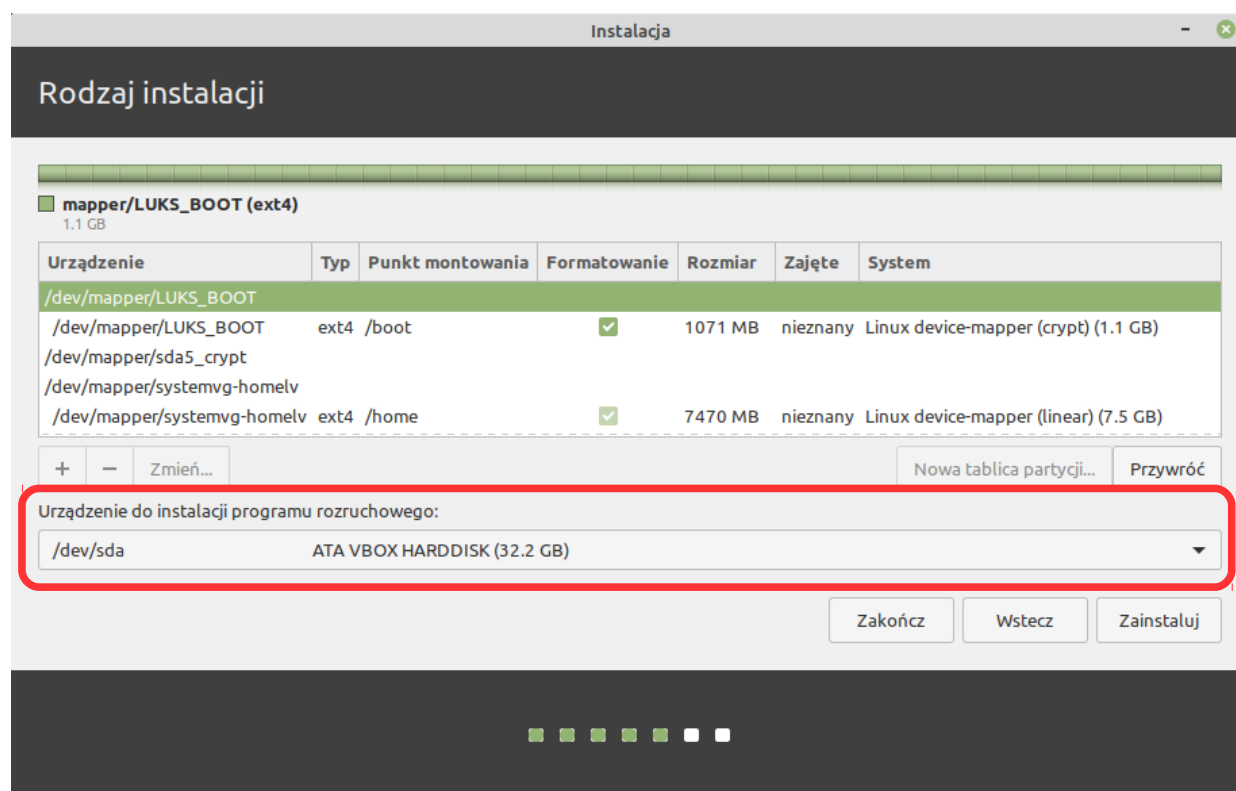
Rys. 6 Przypisanie folderów do odpowiednich partycji - /boot



Rys. 7 Przypisanie folderów do odpowiednich partycji - /root

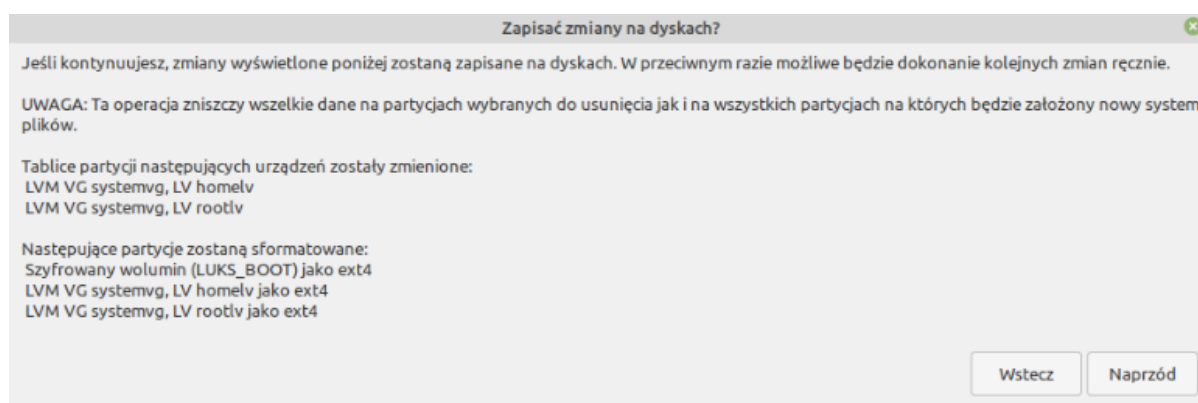
Partycji EFI-SP, nie trzeba wskazywać, instalator automatycznie ją rozpozna, dzięki wcześniej ustawionym flagom: `boot`, `esp`. Jeżeli utworzono wcześniej partycję SWAP, również należy ją podmontować do wybranej lokalizacji w tym kroku instalacji.

Na koniec należy wskazać gdzie instalator ma szukać sektora rozruchowego podczas uruchamiania PC-ta. Należy wskazać główny katalog urządzenia, w tym wypadku `/dev/sda`. Może się zdarzyć, że zakreślone kolorem czerwonym pole wyboru się nie pojawi. Oznacza to, że instalator automatycznie wybrał dysk, na którym instalowany jest system, jako urządzenie rozruchowe.



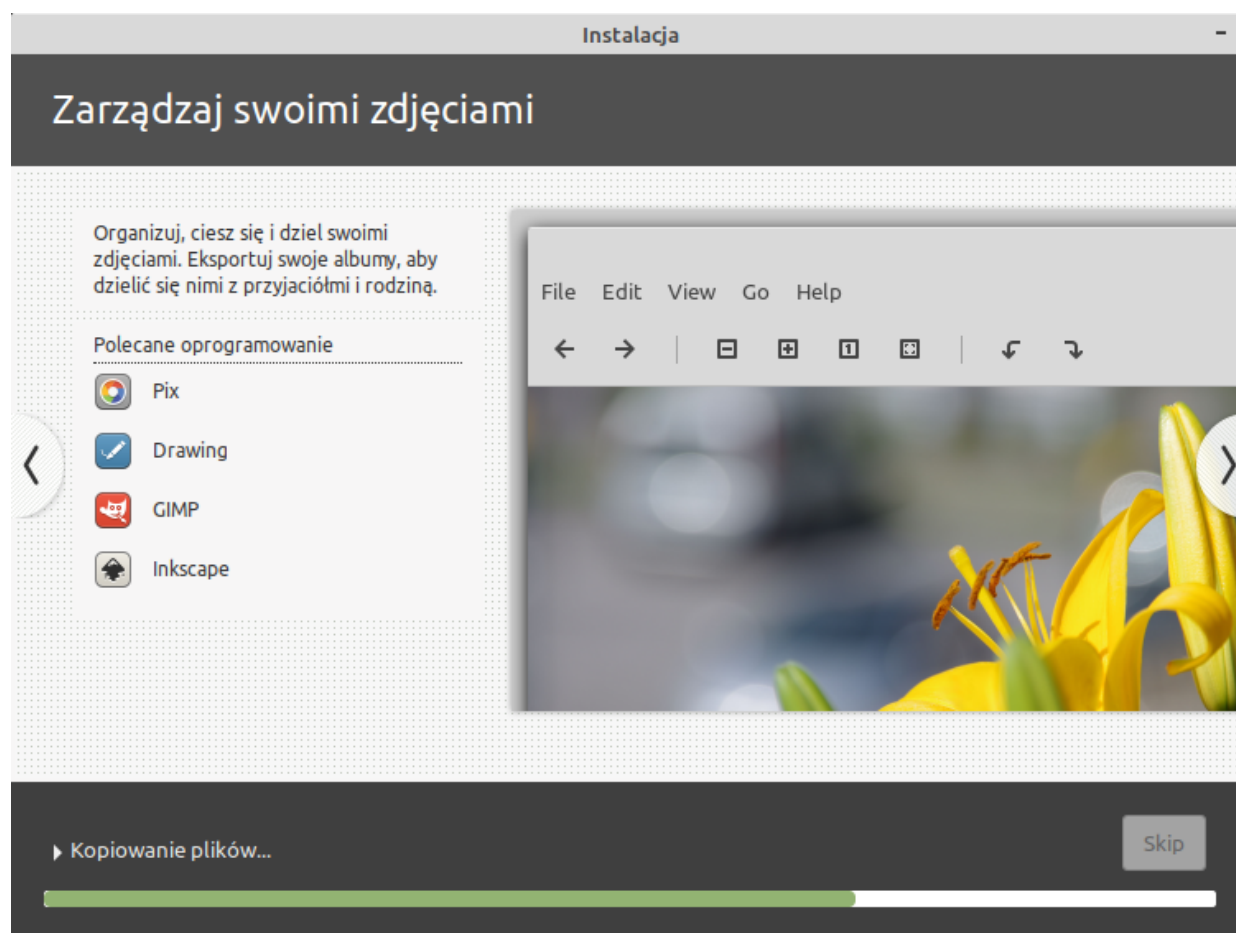
Rys. 8: Wybór miejsca do instalacji programu rozruchowego

Po ustawieniu punktów montowania i wybraniu przycisku Zainstaluj, pojawi się ostrzeżenie:



Rys. 9 Ostrzeżenie o wprowadzanych zmianach

Należy wybrać przycisk Naprzód. Dalsza część instalacji w trybie graficznym przebiega bez zmian.



Rys. 10 Instalacja systemu - kopiowanie plików

!!!WAŻNA UWAGA!!!

Jeżeli rozpoczęto instalację bez modyfikowania pliku konfiguracyjnego według punktu „*Modyfikacja pliku źródłowego w celu uniknięcia błędu GRUB podczas instalacji*” należy wykonać poniższe polecenie. Jeżeli wykonano wcześniej opisany krok, poniższą komendę można pominąć, stosowna modyfikacja zostanie wykonana w późniejszym czasie, już po zakończeniu instalacji systemu.

Po rozpoczęciu kopiowania plików, gdy pasek postępu będzie za połową, ale jeszcze przed ukończeniem kopiowania, należy w terminalu wydać polecenie:

```
# echo "GRUB_ENABLE_CRYPTODISK=y" >> /target/etc/default/grub
```

Polecenie to umożliwi uruchomienie zainstalowanego systemu z zaszyfrowanej partycji GRUB.

Po wykonaniu powyższych czynności system zostanie zainstalowany na dysku twardym komputera. Po zakończeniu instalacji systemu, nie należy wyłączać dystrybucji Live CD. Konieczne jest wykonanie dodatkowej konfiguracji po zakończeniu instalacji systemu.

4. Konfiguracja systemu po instalacji

Dalsza konfiguracja przeprowadzana jest już w docelowym, zainstalowanym systemie. Aby móc pracować na nowo zainstalowanym systemie z poziomu Live CD, należy zamontować wymagane katalogi nowo zainstalowanego systemu i wydać polecenie pracy na tych katalogach za pomocą komendy `chroot` (`change root`).

4.1. Zamontowanie wymaganych katalogów systemu w trybie Live CD

```
# mount /dev/mapper/systemvg-rootlv /target
# mount /dev/mapper/LUKS_BOOT /target/boot
# mount --bind /dev /target/dev
# mount --bind /dev/pts /target/dev/pts
# mount --bind /sys /target/sys
# mount --bind /proc /target/proc
# mount --bind /run /target/run
# mount /dev/sda1 /target/boot/efi
```

Uruchomienie wykonywania poleceń w nowo zainstalowanym systemie

```
# chroot /target
```

Po wykonaniu powyższego polecenia, pozostałe kroki konfiguracyjne są wykonywane na nowo zainstalowanym systemie na dysku twardym.

4.2. Instalacja brakujących pakietów GRUB dla systemu UEFI

Ponieważ podczas instalacji systemu, pominięta została instalacja modułu ładującego, aby uniknąć błędu instalatora, należy brakujący moduł teraz doinstalować.

```
# apt-get update
# apt-get -y install grub-efi
```

4.3. Automatyczne montowanie zaszyfrowanych woluminów przy starcie systemu

Kolejne polecenia są nadal wykonywane w środowisku `chroot`.

System zostanie skonfigurowany w taki sposób, aby wpisanie hasła było konieczne tylko jeden raz, podczas odszyfrowywania partycji `/boot`, zawierającej program ładowania GRUB. Po odblokowaniu partycji `/boot`, pozostałe zaszyfrowane partycje zostaną

automatycznie odblokowane, dzięki zastosowaniu pliku-kłucza odblokowującego, który zostanie wygenerowany przy użyciu poniższych poleceń.

Jeżeli użytkownik nie chce tworzyć plików-kłuczy, może skonfigurować system tak, by odblokowanie pozostałych zaszyfrowanych partycji odbywało się ręcznie. Jednak w takim wypadku konieczne będzie wpisanie hasła tyle razy ile jest zaszyfrowanych partycji. Dla poniższego scenariusza hasło należałoby wpisać trzykrotnie. Dla partycji /boot, oraz dwóch partycji LUKS systemvg oraz datavg.

Poniższa konfiguracja tworzy oddzielny klucz dla zaszyfrowanych partycji LUKS na dysku 1 (dev/sda) i oddzielny dla partycji na dysku 2 (/dev/sdb). Jednak nic nie stoi na przeszkodzie, aby istniał jeden wspólny klucz dla wszystkich dysków.

a) Instalacja pakietów pozwalających na automatyczne odblokowanie zaszyfrowanych woluminów za pomocą pliku-kłucza

```
# apt-get install -y cryptsetup-initramfs
```

W systemie operacyjnym *Linux Mint 19.3* pakiet jest niedostępny, ponieważ jest częścią głównego pakietu `cryptsetup` i jest już zainstalowany.

b) Zapewnienie automatycznego generowania kluczy przy kolejnych aktualizacjach systemu

Poniższe polecenia nakazują systemowi dołączenie plików kluczy (z wzorcem kończącym się na *.keyfile) zawartych w katalogu /etc/luks/ do pliku `initramfs` podczas jego aktualizacji.

```
# echo "KEYFILE_PATTERN=/etc/luks/*.keyfile" >>
/etc/cryptsetup-initramfs/conf-hook
# echo "UMASK=0077" >>
/etc/initramfs-tools/initramfs.conf
```

c) Utworzenie plików kluczy i dodanie ich do systemu

Utworzenie plików kluczy

```
# mkdir /etc/luks
# dd if=/dev/urandom of=/etc/luks/boot_os.keyfile
bs=4096 count=4
# dd if=/dev/urandom of=/etc/luks/boot_data.keyfile
bs=4096 count=4
```

Nadanie praw dostępu do plików kluczy

```
# chmod u=rx,go-rwx /etc/luks
# chmod u=r,go-rwx /etc/luks/boot_os.keyfile
# chmod u=r,go-rwx /etc/luks/boot_data.keyfile
```

Ustawienie zmiennych środowiskowych

Ponieważ aktualnie komendy są wykonywane w innym systemie operacyjnym niż na początku (wcześniej był to Live CD, teraz jest system zainstalowany na dysku twardym; konieczne jest ponowne dodanie zmiennych środowiskowych do nowego systemu. Ważna jest kolejność dodawania zmiennych. Zmienne DM nie mogą być dodane przed DEV, gdyż nie będą działać poprawnie.

```
# export DEV="/dev/nvme0n1" lub # export DEV="/dev/sda"
# export DM="${DEV}##*/"
# export DEV2="/dev/sdb"
# export DM2="${DEV2}##*/"
```

Dodanie kluczy do systemu

```
# cryptsetup luksAddKey ${DEV}2 /etc/luks/boot_os.keyfile
Enter any existing passphrase:
```

```
# cryptsetup luksAddKey ${DEV}5 /etc/luks/boot_os.keyfile
Enter any existing passphrase:
```

```
# cryptsetup luksAddKey ${DEV2}1 /etc/luks/boot_data.keyfile
Enter any existing passphrase:
```

d) Dodanie kluczy do crypttab

```
# echo "LUKS_BOOT UUID=$(blkid -s UUID -o value ${DEV}2)
/etc/luks/boot_os.keyfile luks,discard" >> /etc/crypttab

# echo "${DM}5_crypt UUID=$(blkid -s UUID -o value ${
{DEV}5) /etc/luks/boot_os.keyfile luks,discard" >> /etc/
crypttab

# echo "${DM}1_crypt UUID=$(blkid -s UUID -o value ${
{DEV2}1) /etc/luks/boot_data.keyfile luks,discard" >>
/etc/crypttab
```

Sprawdzenie poprawności wpisów odblokowywanych partycji

Po dodaniu wpisów partycji, które mają być automatycznie odblokowywane podczas startu systemu, warto sprawdzić poprawność wpisów. Szczególnie partycji niezbędnych dla prawidłowego działania systemu operacyjnego

```
# vi /etc/crypttab
```

=====

Jeżeli do zamontowania na przykład dysku drugiego nie chcemy wykorzystywać klucza, tylko wpisywać hasło ręcznie przy każdym uruchomieniu systemu, możemy pominąć kroki tworzenia klucza dla tego dysku. By ręcznie wpisywać hasło, polecenie montowania tego dysku w crypttab, będzie miało postać:

```
# echo "${DM}1_crypt UUID=$(blkid -s UUID -o value ${
{DEV2}1) none luks,discard" >> /etc/crypttab
```

=====

e) Aktualizacja plików ramfs

Na koniec należy zaktualizować początkowe pliki ramfs, aby dodać skrypty odblokowujące cryptsetup i pliki kluczy. Aktualizację ramfs, trzeba wykonać po każdej zmianie dokonanej w pliku /etc/crypttab, aby te zmiany zaczęły funkcjonować.

```
# locale-gen --purge --no-archive
# update-initramfs -u -k all
```

4.4. Aktualizacja konfiguracji GRUB

a) Edycja wpisów w pliku `/etc/default/grub`

Zmian należy dokonywać cały czas jako chroot

Edycji pliku należy dokonać w wybranym przez siebie edytorze tekstowym, na przykład: `vim`, `nano`, `xed`. Wpisy wytłuszczone, są to wpisy, które zostały dodane lub edytowane.

```
#####  
GRUB_DEFAULT=0  
GRUB_TIMEOUT_STYLE=menu  
#GRUB_TIMEOUT_STYLE=hidden  
GRUB_HIDDEN_TIMEOUT=0  
GRUB_HIDDEN_TIMEOUT_QUIET=true  
GRUB_TIMEOUT=5  
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`  
GRUB_ENABLE_CRYPTODISK=y  
GRUB_CMDLINE_LINUX_DEFAULT="cryptdevice=/dev/sda5:sda5_crypt"  
GRUB_CMDLINE_LINUX=""  
#####
```

Znaczenie wyróżnionych opcji:

`GRUB_TIMEOUT_STYLE=menu`

Od tego wpisu zależy czy podczas uruchamiania PC-ta widoczne będzie menu umożliwiające wybranie opcji rozruchu (systemu operacyjnego). Wpis `menu` oznacza, menu wyboru się pojawi. Menu warto wyświetlić jeżeli np. po aktualizacji jądra pojawi się problem z uruchomieniem systemu (wtedy należy zalogować się do systemu z poziomu Live CD, podobnie jak opisano powyżej), ponieważ w menu istnieje wpis, który umożliwia uruchomienie systemu ze starszą wersją jądra, sprzed aktualizacji (o ile nie zostało wcześniej usunięte przez użytkownika).

`GRUB_HIDDEN_TIMEOUT=0`

Jeżeli powyższy wpis istnieje, wówczas menu rozruchu zostanie wyświetlone tylko w przypadku wciśnięcia przycisku na klawiaturze podczas uruchamiania systemu. Wartość po znaku „=” definiuje ile sekund GRUB będzie czekał na wciśnięcie przycisku.

`GRUB_TIMEOUT=5`

Określa przez ile sekund GRUB oczekuje na wybranie systemu operacyjnego jaki ma zostać uruchomiony. Jeżeli w pliku jest wpis `GRUB_TIMEOUT_STYLE=hidden`, GRUB również oczekuje na wybranie opcji rozruchu, jednak menu jest ukryte. Dlatego, przy domyślnych ustawieniach może się wydawać, że system „zawisł” na kilka sekund, po wpisaniu hasła do partycji `/boot`.

`GRUB_ENABLE_CRYPTODISK=y`

Wymagany wpis, aby poprawnie uruchomić system z zaszyfrowaną partycją `/boot`.

`GRUB_CMDLINE_LINUX_DEFAULT="cryptdevice=/dev/sda5:sda5_crypt"`

Konieczne jest dodanie wpisu `cryptdevice=...`, dla partycji LUKS, na jakiej jest montowany katalog `/`.

Usunięcie wpisu `quiet splash`, który domyślnie znajduje się w tej linii spowoduje, że podczas uruchamiania systemu, zamiast logo, widoczne będą wpisy z przebiegu uruchamiania systemu. Pomocne, jeżeli system nie chce się uruchomić lub pojawiają się jakieś błędy podczas uruchamiania.

b) Aktualizacja GRUB

Poniższą komendę trzeba wykonać po każdej zmianie dokonanej w pliku `/etc/default/grub`

```
# update-grub
```

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

4.5. Dodatkowa konfiguracja GRUB**Zmian należy dokonywać cały czas jako `chroot`**

Wykonanie poniższego polecenia może być konieczne, jeżeli system nie będzie chciał się uruchomić, po zrestartowaniu PC-ta [możliwość rozruchu z dysku z zainstalowanym GRUB nie będzie widoczna w BIOS (UEFI)]. Polecenie może też stanowić alternatywę, jeżeli ktoś chce dostosować konfigurację GRUB do swoich

indywidualnych potrzeb i/lub wymagań.

Poniższe polecenia aktualizują GRUB i generują pojedynczy plik efi GRUB, o nazwie `grubx64.efi`, który zawiera moduł ładujący ze wszystkimi wymaganymi plikami i modułami. Wybrany parametr `--bootloader-id=Mint` spowoduje, że GRUB zbuduje katalog o nazwie `/boot/efi/EFI/Mint` i wpis EFI NVRAM o nazwie `Mint`. Modyfikując poniższe polecenie można zbudować własną wersję pliku `*.efi`, zgodną z własnymi preferencjami.

```
# grub-install --target=x86_64-efi
--efi-directory=/boot/efi --bootloader-id=Mint --boot-
directory=/boot --modules="all_video boot btrfs cat chain
configfile crypto cryptodisk disk diskfilter echo
efiwget setup efinet ext2 fat font gettext gcry_arcfour
gcry_blowfish gcry_camellia gcry_cast5 gcry_crc gcry_des
gcry_dsa gcry_idea gcry_md4 gcry_md5 gcry_rfc2268
gcry_rijndael gcry_rmd160 gcry_rsa gcry_seed gcry_serpent
gcry_sha1 gcry_sha256 gcry_sha512 gcry_tiger gcry_twofish
gcry_whirlpool gfxmenu gfxterm gfxterm_background gzio halt
hfsplus iso9660 jpeg keystatus loadenv loopback linux
linuxefi lsefi lsefimmap lsefisystab lssal luks lvm
mdraid09 mdraid1x memdisk minicmd normal part_apple
part_msdos part_gpt password_pbkdf2 png raid5rec raid6rec
reboot search search_fs_uuid search_fs_file search_label
sleep squash4 test true verify video zfs zfscrypt zfsinfo"
-recheck
```

Aby usunąć domyślną konfigurację GRUB, należy wydać polecenie:

```
# rm -r /mnt/boot/efi/EFI/ubuntu
```

4.6. Zakończenie konfiguracji systemu

Po wykonaniu powyższych czynności można odmontować katalogi systemu zainstalowanego na dysku twardym, opuszczając środowisko `chroot`:

```
# exit
```

Spowoduje to powrót do pracy w systemie `Live CD`. Z tego miejsca można

odmontować katalogi nowo zainstalowanego systemu na dysku twardym:

```
# umount /target/boot/efi /target/proc /target/dev/pts  
/target/dev /target/sys /target/boot /target
```

Jeżeli wszystko poszło dobrze, po wykonaniu powyższych czynności system jest gotowy do ponownego uruchomienia.

Po ponownym uruchomieniu systemu, powinien pojawić się komunikat:

```
Attempting to decrypt master key...
```

```
Enter passphrase fot hd0,gpt2 (4aaf9c7e6edd4ca5a8a94b88d6991364)
```

Po wpisaniu hasła system powinien zostać odblokowany i normalnie uruchomiony.

Instalacja na komputerze z UEFI i HDD z GPT została zakończona

4.7. Backup nagłówków kontenera LUKS

Po uruchomieniu systemu, dobrze jest zrobić sobie backup nagłówków zaszyfrowanych partycji i trzymać te pliki w bezpiecznym miejscu. Backup robimy w poniższy sposób:

```
# cryptsetup luksHeaderBackup /dev/sda2 --header-backup-  
file sda2_header_backup  
# cryptsetup luksHeaderBackup /dev/sda5 --header-backup-  
file sda5_header_backup  
# cryptsetup luksHeaderBackup /dev/sdb1 --header-backup-  
file sdb1_header_backup
```

5. Montowanie woluminów LVM w katalogu użytkownika

Dotychczas wykonane kroki pozwoliły na automatyczne odblokowanie zaszyfrowanych partycji LUKS podczas uruchamiania systemu. Jednak logiczne woluminy LVM utworzone na dysku drugim, nie są jeszcze automatycznie montowane w wybranych katalogach użytkownika systemu. Aby tak było trzeba wykonać dodatkowe czynności.

Podczas instalacji systemu utworzone zostało konto użytkownika o nazwie `test`. Dlatego wszystkie wpisy dotyczące montowania woluminów LVM odnoszą się do katalogu domowego tego użytkownika.

5.1. Aktualizacja wpisów w `fstab`

Za automatyczne montowanie partycji podczas startu systemu odpowiada plik `/etc/fstab`. Aby wcześniej utworzone i zmapowane, zaszyfrowane partycje były montowane automatycznie podczas startu systemu należy dokonać odpowiednich wpisów we wskazanym pliku.

Uruchomienie pliku do edycji za pomocą edytora tekstu, np. `vim`, `nano`, `xed`

```
$ sudo vim /etc/fstab
```

Dodanie niezbędnych wpisów w pliku `/etc/fstab`

```
# Montowanie dysków LVM jako katalogów użytkownika
/dev/datavg/documentslv /home/test/Dokumenty ext4 defaults,x-gvfs-hide 0 2
/dev/datavg/musiclv /home/test/Muzyka ext4 defaults,x-gvfs-hide 0 2
/dev/datavg/pictureslv /home/test/Obrazy ext4 defaults,x-gvfs-hide 0 2
/dev/datavg/videolv /home/test/Wideo ext4 defaults,x-gvfs-hide 0 2
/dev/datavg/downloadlv /home/test/Pobrane ext4 defaults,x-gvfs-hide 0 2
```

Szczegółowe informacje na temat wpisów w pliku `/etc/fstab` i konfiguracji montowanych partycji można bez problemu znaleźć w sieci. Tutaj zostanie jedynie wyjaśniony wpis: `x-gvfs-hide`, który oznacza, że tak zamontowana partycja nie będzie widoczna w panelu urządzeń (po lewej stronie domyślnego menedżera plików) oraz na pulpicie - jako zamontowane urządzenie. Będzie dostępna w katalogu, w którym została zamontowana. Jednak partycje będą już widoczne jako niezależne urządzenia w menadżerze dwu panelowym, np. `Double Commander`. Przy takim wykorzystaniu dodatkowych partycji jak tutaj opisane, jest to wygodne rozwiązanie, jednak finalna

konfiguracja zależy wyłącznie od preferencji użytkownika.

Test montowania partycji

Test montowania partycji można przeprowadzić bez ponownego uruchamiania systemu za pomocą komendy:

```
$ sudo mount -a
```

Komenda `mount` z parametrem `-a` montuje wszystkie partycje, których wpisy widnieją w pliku `/etc/fstab`.

Poniższa komenda pozwala sprawdzić czy montowanie rzeczywiście nastąpiło prawidłowo:

```
$ df -h | grep /home/test  
/dev/mapper/datavg-documentslv      ...    /home/test/Dokumenty  
/dev/mapper/datavg-downloads        ...    /home/test/Pobrane  
/dev/mapper/datavg-videos           ...    /home/test/Wideo  
/dev/mapper/datavg-pictureslv       ...    /home/test/Obrazy  
/dev/mapper/datavg-musiclv          ...    /home/test/Muzyka
```

5.2. Zmiana uprawnień dostępu dla montowanych partycji

Domyślnie po zamontowaniu partycji w powyższy sposób właścicielem plików i katalogów we wskazanych lokalizacjach będzie użytkownik `root`. Również wszystkie uprawnienia dla nowo tworzonych plików i katalogów w tych lokalizacjach będą tworzone z dostępem zgodnym dla użytkownika `root`. Można to sprawdzić za pomocą polecenia:

```
$ ls -l /home/test  
drwxr-xr-x  2 root root      4096 kwi 16 20:15 Dokumenty  
(Pozostałe wpisy dotyczące zamontowanych partycji będą podobne)
```

Aby zmienić właściciela katalogów i tym samym nadać odpowiednie uprawnienia dla nowo tworzonych plików i katalogów na wskazanych partycjach, należy wydać polecenia:

```
$ sudo chown -R test:test /home/test/Dokumenty/  
$ sudo chown -R test:test /home/test/Muzyka/  
$ sudo chown -R test:test /home/test/Obrazy/
```

```
$ sudo chown -R test:test /home/test/Wideo/  
$ sudo chown -R test:test /home/test/Pobrane/
```

Poprawność wykonanych komend można sprawdzić za pomocą polecenia:

```
$ ls -l /home/test  
drwxr-xr-x  2 test test          4096 kwi 16 20:20 Dokumenty  
(Pozostałe wpisy dotyczące zamontowanych partycji będą podobne)
```

Jak widać po rezultacie polecenia zmieniony został właściciel i grupa, do których należą wskazane katalogi. Również wszystkie nowo tworzone pliki i katalogi we wskazanych lokalizacjach będą tworzone jako własność i przynależność do grupy użytkownika `test` i będą tworzone z prawidłowymi uprawnieniami, domyślnymi dla użytkownika.

Oczywiście podczas wykonywania poleceń w docelowym systemie, należy nazwę użytkownika `test` zamienić na właściwą nazwę użytkownika.

Uwagi końcowe

Powyższe polecenie było ostatnim jakie należało wykonać w powyższym tutorialu. Polecenia tutaj zawarte zostały pierwotnie wykonane w wirtualnym systemie instalowanym za pomocą narzędzia VirtualBox, a następnie została przeprowadzona „prawdziwa” instalacja na fizycznym sprzęcie docelowym. Również konfiguracja w docelowym środowisku przebiegła i działa prawidłowo.

Ze względu na fakt, że podczas testowania komend i kopiowania poleceń z pliku *.pdf wprost do terminala, pojawiały się błędy w poleceniach np. w postaci zamiany znaków „--” na „—”, zalecane jest skopiowanie poleceń do pliku tekstowego i dopiero stamtąd kopiowanie ich do terminala. Taki sposób pozwoli uniknąć błędów poleceń, jakie mogą wystąpić przy kopiowaniu do terminala wprost z pliku *.pdf.

Bibliografia

Instalacja Linux Mint 19.3 Full Disk Encryption

- https://help.ubuntu.com/community/Full_Disk_Encryption_Howto_2019 – opis instalacji na podstawie Ubuntu
- <https://community.linuxmint.com/tutorial/view/2061> - opis instalacji dla systemu Linux – rozwiązanie problemu z instalatorem
- <https://nowhere.dk/articles/installing-linux-mint-ubuntu-desktop-edition-with-full-disk-encryption-and-lvm/comment-page-1> - opis instalacji dla systemu Linux – bez szyfrowania partycji „/boot”

Konfiguracja woluminów LVM montowanych w katalogu użytkownika

- https://linuxhint.com/lvm_home_directories/ - Konfiguracja montowania woluminów LVM

Dodatkowe, pomocnicze źródła, przydatne w zrozumieniu niektórych zagadnień

- <http://manpages.ubuntu.com/manpages/xenial/en/man5/fstab.5.html> – Konfiguracja fstab
- <https://community.linuxmint.com/tutorial/view/1513> – Edycja pliku fstab
- <https://morfikov.github.io/post/przejscie-z-truecrypt-na-luks/> - Pliki crypttab oraz fstab
- https://pl.wikipedia.org/wiki/Typ_partycji – Kody typów partycji
- [https://wiki.archlinux.org/index.php/Dm-crypt/Encrypting_an_entire_system_\(Polski\)](https://wiki.archlinux.org/index.php/Dm-crypt/Encrypting_an_entire_system_(Polski)) – Instalacja typu LVM na LUKS
- https://help.ubuntu.com/community/ManualFullSystemEncryption/BasicLVM#The_system_with_both_LUKS_and_LVM – Graficzna prezentacja zagadnienia konfiguracji LVM na LUKS
- <https://www.youtube.com/watch?v=oBq2U-H1QTk> – Tworzenie partycji pod UEFI
- <https://www.erianna.com/adding-an-secondary-encrypted-drive-with-lvm-to-ubuntu-linux/> - Dodatkowy dysk z LVM i LUKS
- https://wiki.opzsgu.pl/index.php/Jak_zaszyfrowa%C4%87_ca%C5%82y_dysk_przez_LUKS_i_montowa%C4%87_automatycznie_podczas_uruchamiania_komputera%3F – Automatyczne montowanie woluminów LUKS
- <https://www.digitalocean.com/community/tutorials/how-to-use-lvm-to-manage-storage-devices-on-ubuntu-18-04> – Zarządzanie wolumenami LVM z poziomu linii komend

- <https://help.ubuntu.com/community/ManualFullSystemEncryption> – Uwagi do instalacji systemu szyfrowanego