Win10+: Linux Encrypted USB-SSD

-JMB

Project Features & Objectives:

- A recent experiment …
- In Win10+: access, store & backup files to a locally mounted USB-SSD.
- The SSD is encrypted and only the owner of the computer has the decryption ‘key’.
  - the key (passphrase) can just be memorized, written on paper, etc.
  - can but need not be stored on any electronic device.
  - The ext4 formatted USB-SSD is Linux Unified Key Setup (LUKS) encrypted ***.
- Use free and open source software (FOSS).
- Not reliant on any cloud based storage.
- Use a device that can be:
  - easily attached / detached.
  - physically stored in a secure location of one’s choosing.
  - small and reasonably portable.
  - a capacity that one’s budget will allow (100’s GB / 1+ TB).
  - affordable and incur only an initial cost (no subscription fees).

*** https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup

Need, Limitations, other Options

Need?

- Enable family (Win10+users) to backup data to a device they physically possess.

History, Experience & Track Record:

- Have used LUKS on Linux computers for more than a decade.
- Proven itself as reliable,day in and day out, with multiple devices.

Limitations:

- The backup is only as good as the storage location of the SSD !
- If the password is lost, so is the data
- If the SSD is corrupted by premature disconnection, the data could be lost.
- If the SSD fails, the data could be unrecoverable.

Other Options:

- Win10+ offers BitBucket which is:
  - perhaps incompatible with Linux.
  - watered-down in the ‘Home’ version.
  - linked to a Micro$oft account where the key is stored.

- 3rd party free / $oftware ‘VeraCrypt’, ‘TrueCrypt’, ‘DiskCryptor’, AxCrypt, etc.
  - compatibilty with Linux is non-existent or questionable

- Built-in encryption in the SSD-HD
  - Linux compatibility unknown.
  - not willing to spend money to experiment or risk one’s data.

Hardware:

- SATA SSD and a USB 3.0+ HD enclosure

  or

  USB 3.0+ SSD
- A computer running a fairly recent Linux (Debian) distribution
  - for formatting the USB-SSD as ext4 & LUKS encryption
- A computer running Win 10+ with WSL2 installed
  - the computer that needs to access the USB-SSD
  - with sufficient RAM (8GB min) & disk space (250 GB min)

USB + SATA SSD *:

USB 3.0 Enclosure

- 2.5” 5Gbps HD Case for SATA Drive
- Approx. $4 (incl. S&H - AliExpress)

SATA SSD

- Samsung SSD 870 QVO 2TB
- Samsung online store

* Why not a pre-built USB-SSD?

Connect the SSD inside a desktop PC via SATA cable if needed

Overview: Sections A & B

A: Prepare a USB-SSD (ext4 / LUKS) using a Linux computer.

- Only briefly described …
- The IBM article listed in the last slide “Reference (LUKS)”, is a well written one that shows the command line way to accomplish this.

B: Install, configure a Win10+ computer to use the USB-SSD.

- Installing WSL2
- Installing Ubuntu (guest OS) in WSL2
- Installing necessary software, scripts, shortcuts, icons, etc.

Section A: (Linux computer)

Format an SSD as LUKS & ext4:

- In the Linux (Ubuntu) PC (not the Win10+ PC):

  sudo aptitude install crypt-setup gnome-disk-utility

  If not already installed by default ...

- Prepare the USB-SSD:

  MATE desktop: Menu → All → Disks → Encrypted (LUKS) → Format (ext4; Partitioned: "USB-SSD") → etc.

  *** Generally there will be more than one drive listed.

  Be VERY careful and select the correct USB connected drive !!! ***

- Create a file at the top level of the USB-SSD within its encrypted partition when it is in an unlocked state:

  touch /mnt/USB-SSD/If_you_can_see_this_it_is_unlocked

  (it will be useful later, as you will see …)

- Un-mount and disconnect the USB-SSD.

  (you are done with the Linux computer ...)

Section B: (Win10+ Computer)

Software: Win10+ (Install WSL2)

- In PowerShell as ‘Administrator:

  wsl --install

- Reboot your Windows computer.

  https://learn.microsoft.com/en-us/windows/wsl/install#install-wsl-command

Software: Win10+ (Install Ubuntu)

- cmd: wsl --list --online

  NAME FRIENDLY NAME

  Ubuntu-18.04 Ubuntu 18.04 LTS

  …

  Ubuntu-24.04 Ubuntu 24.04 LTS

  etc.

- cmd: wsl --install Ubuntu-24.04

  Installing: Ubuntu 24.04 LTS

  [=====XX.X%=====]

  …

  Enter new UNIX username:

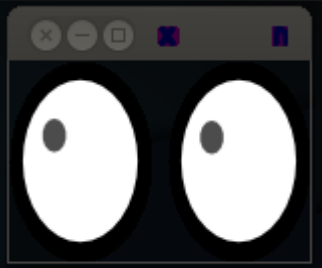
  New password:

  Retype new password:

  etc, etc ...

- cmd: wsl --set-default Ubuntu-24.04
  - If you had installed an older version or different Linux OS
- cmd: wsl --help
  - Shows various options ...

Software: Win11 (Test Ubuntu)



- bash terminal: sudo apt-get install x11-apps
- bash terminal: xeyes &

  This will display 2 eyes that follow the mouse cursor
- bash terminal: echo $DISPLAY

  :0
- Win10+WSL2 (Ubuntu)

  ‘xeyes’ does NOT work!

  Also the $DISPLAY env variable is <blank>

  bash terminal: export $DISPLAY=”:0” does not work !!!

  It seems that GUI output from Ubuntu to the host OS Win10 is not supported,

  or I have not found a solution.

So we need a workaround, more later …

Software: Win10+ computer

- usbipd-win (in the ‘host’ OS)
- Windows Subsystem for Linux (specifically WSL2)
  - Ubuntu 22.04 (‘guest’ OS)
    - linux-tools-X.XX.X-XX-generic
      (choose the right version for your running kernel)
    - hwdata
    - crypt-setup (if not already present)

Install USBIPD-WIN in Win10+:

cd ~/MyFiles/Downloads/
wget -nd https://github.com/dorssel/usbipd-win/releases/download/v4.2.0/usbipd-win_4.2.0.msi

Or download using a browser and save to your favorite folder in M$-Windows ...

Install this tool in Win10+ by double clicking the '.msi' file.
(Allow Win10+ to install from non-Microsoft Store locations!)

Caveat:

Don't know how I managed to change the above setting.
So I do not have the exact steps, menu path, etc.
Somehow it appeared from my fumbling around.

What does it do?

Provides support for connecting USB devices which is not natively available in WSL.

Free and open source: https://github.com/dorssel/usbipd-win/releases

Share & Attach USB Device (Win10+):

Connect the USB SSD to the computer
List all the USB devices connected to Windows in a PowerShell as administrator:

usbipd list

```
Connected:
BUSID  VID:PID    DEVICE                                        STATE
1-5    xxxx:yyyy  USB Attached SCSI (UAS) Mass Storage          DeviceNot shared
```

Start a WSL2 (Ubuntu) terminal & share the device, allowing it to be attached to WSL2:

usbipd bind --busid 1-5

*** IMPORTANT ***
Ensure that this WSL2 command prompt is open in order to keep the WSL2 lightweight VM active.
Once attached to WSL2, the USB device can be used by any distribution (ex. Ubuntu) running as WSL2.
Note that as long as the USB device is attached to WSL2, it cannot be used by Windows.

In a PowerShell. You no longer need to use an elevated administrator prompt.

usbipd attach --wsl --busid 1-5

```
usbipd: info: Using WSL distribution 'Ubuntu' to attach; the device will be available in all WSL 2 distributions.
usbipd: info: Using the IP address xxx.xx.xxx.1 to reach the host.
```

Access USB-SSD in WSL2 (Ubuntu):

- *lsusb*

  Bus 002 Device 002: ID xxxx:xxxx Some Technology Corp SATA 6Gb/s

- *uname -a*

  Linux PC 5.15.146.1-microsoft-standard-WSL2 # 1 SMP Thu Jan 11 04:09:03 UTC 2024 x86_64 ... GNU/Linux

- *sudo apt install linux-tools-5.15.0-97-generic hwdata*

  (bash script to install the correct version -next slide)

- *sudo update-alternatives --install /usr/local/bin/usbip usbip \ /usr/lib/linux-tools/5.15.0-97-generic/usbip 20*

  update-alternatives: using /usr/lib/linux-tools/5.15.0-97-generic/usbip to provide /usr/local/bin/usbip (usbip) in auto mode

- Reboot the windows computer.

Access in WSL2 (Ubuntu): script

```
#!/usr/bin/bash

rel="$(uname -r)"
rel="${rel%%-*}"
rel=(${rel//./ })

function latest_linux_tools {}

sudo apt-get install "$@" "$(latest_linux_tools)"
```

```
function latest_linux_tools {
    apt-cache search linux-tools |
    awk -v cur_ver="${rel[*]}" '
    /^linux-tools([-\.][0-9]+)+-generic\>/ {
        ltg_package=$1
        gsub(/[^0-9]+/," ",$1);
        gsub(/^\s*/,"",$1);
        split($1,ltg_ver,/\s*/);
        split(cur_ver,cmp_ver,/\s*/)
        if (ltg_ver[1]<=cmp_ver[1] && ltg_ver[2]<=cmp_ver[2] &&
ltg_ver[3]<=cmp_ver[3]) {
            print ltg_package;
        }
    }' | sort -nr | head -n 1
}
```

Mount the SSD (Win10+):

Win10+ as administrator in a ‘cmd’ terminal:

C:\Windows\System32>wmic diskdrive list brief

Drive	DeviceID	Model	Partitions	Size
Samsung SSD 870 QVO 2TB SCSI Disk Device	\\.\PHYSICALDRIVE1	Samsung SSD 870 QVO 2TB	1	2000396321280
<Internal Drive>	\\.\PHYSICALDRIVE0	<Internal Drive> …		

(If the USB SSD is not listed try disconnecting and reconnecting it and retry the above command)

C:\Windows\System32>wsl --mount --bare \\.\PHYSICALDRIVE1

The operation completed successfully

Open Encrypted SSD (WSL2/Ubuntu):

- **In a WSL2 (Ubuntu) ‘terminal’:**

df -h

C:\Program Files\usbipd-win\WSL xxxG xxG xxG xx% /run/usbipd-win

lsblk

sdX 8:64 0 xxxG 0 disk

└─sdX1 8:65 0 xxxG 0 part

sudo cryptsetup luksOpen /dev/sdX1 USB-SSD-Encrypted

Enter passphrase for /dev/sdd1:

If you do not type the correct passphrase, your only option will be to format the disk and loose ALL data!

lsblk

sdX 8:64 0 xxxG 0 disk

└─sdX1 8:65 0 xxxG 0 part

└─USB-SSD-Encrypted 252:0 0 xxxG 0 crypt

*sudo mkdir /mnt/*USB-SSD-Encrypt

sudo mount /dev/mapper/USB-SSD-Encrypted /mnt/USB-SSD-Encrypt

The operation completed successfully

df -h

Filesystem Size Used Avail Use% Mounted on

... irrelevant lines not shown ...

/dev/mapper/USB-SSD-Encrypted XXXG 44K XXXG 1% /mnt/USB-SSD-Encrypt

*ls -al /mnt/c/*Users/<user>/

- File 1
- File2 etc …
- (These are the Win10+ files -Yay !!!)

ls -al /mnt/USB-SSD-Encrypt

drwx------ 4 user user 0 Jun 10 14:50 If_you_can_see_this_it_is_unlocked

drwx------ 4 root root 4096 Jun 10 14:58 home

drwx------ 4 root root 16384 Jun 10 14:47 lost+found

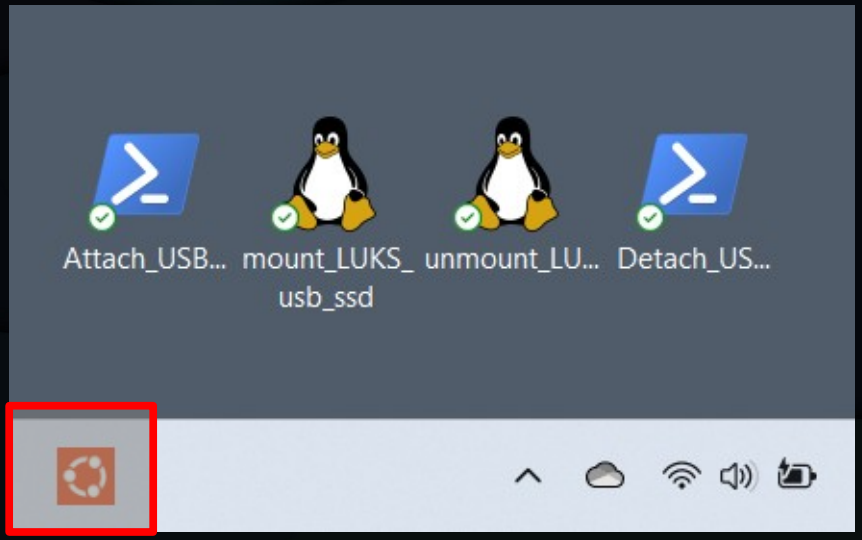
- (and are the Ubuntu files -Yay !!!)

Done…

next GUI window dressing !!!

GUI window dressing: Look & Feel ...

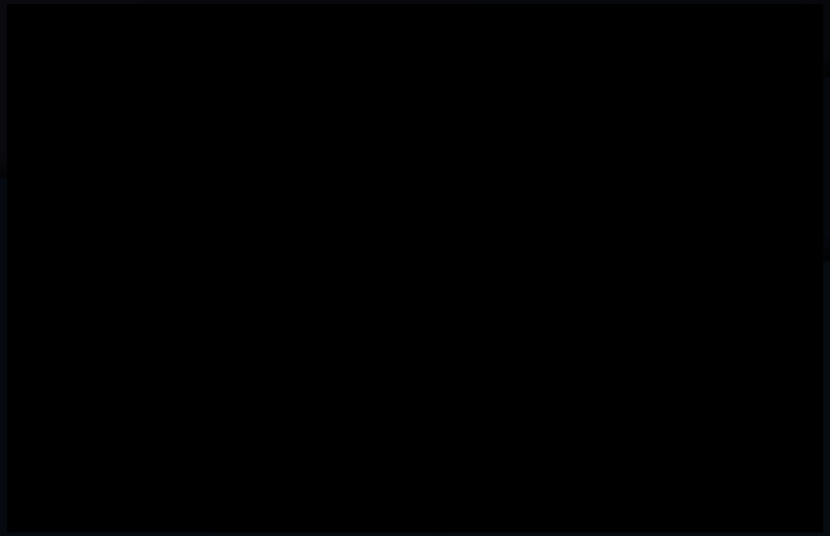
Clickable Desktop shortcuts / icons !



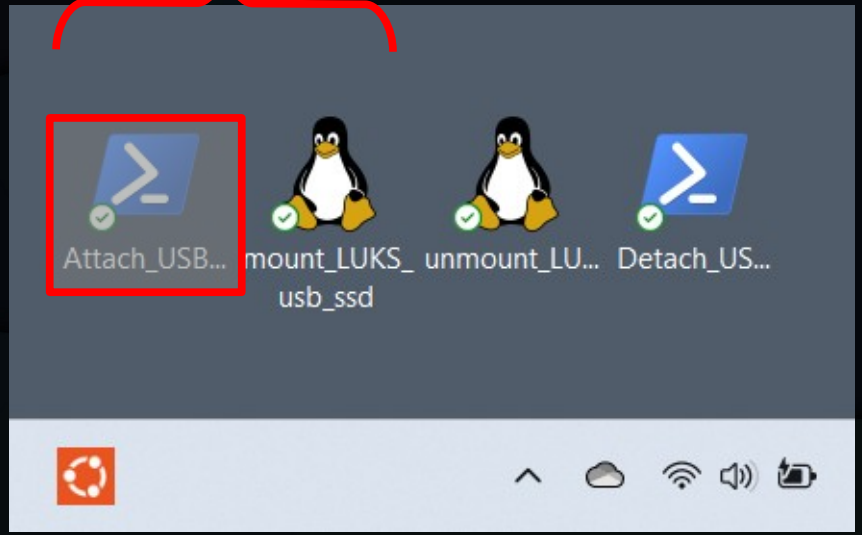
WSL2 / Ubuntu icon →

GUI window dressing: Look & Feel ...

- **Open**
  - **Attach**
    - (permit elevated ‘Adminstrator’ execution -PowerShell)
  - **mount**
    - (type: sudo pswd, LUKS pswd -Ubuntu)

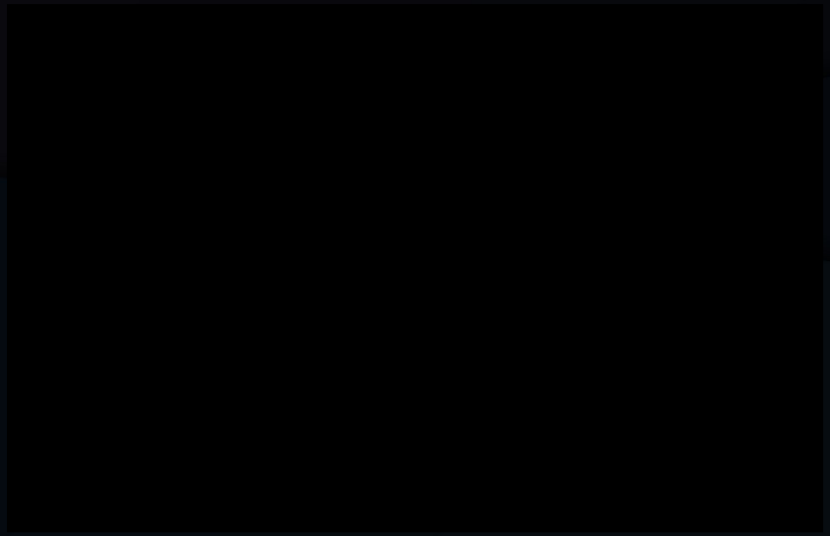


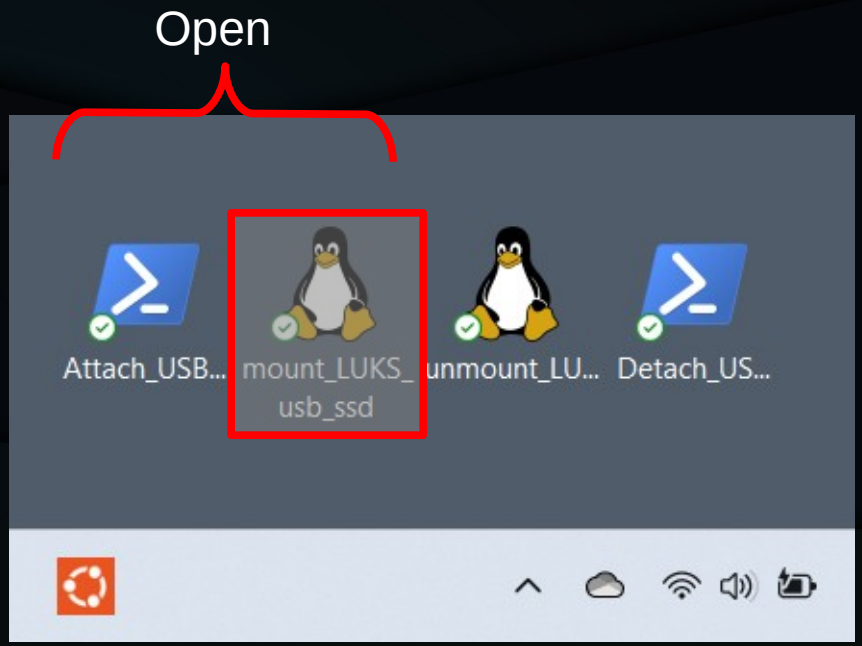
Open



GUI window dressing: Look & Feel ...

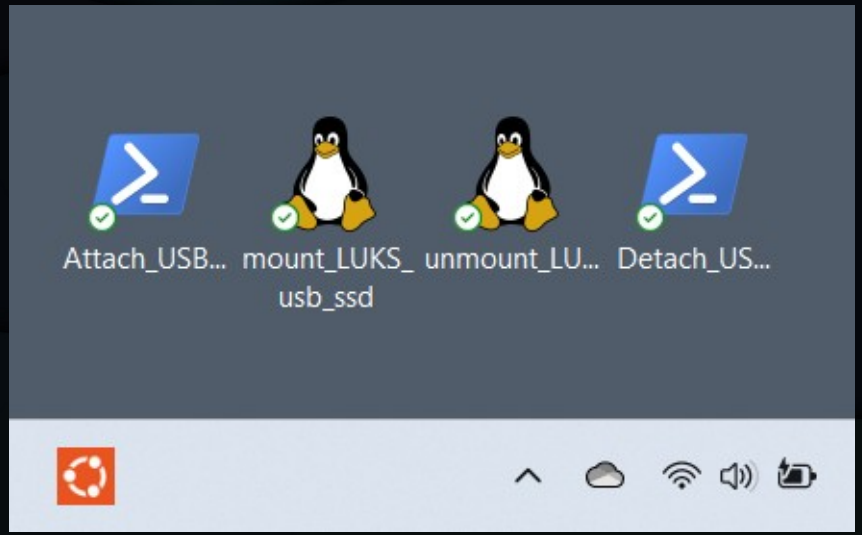
- **Open**
  - **Attach**
    - (permit elevated ‘Adminstrator’ execution -PowerShell)
  - **mount**
    - (type: sudo pswd, LUKS pswd -Ubuntu)



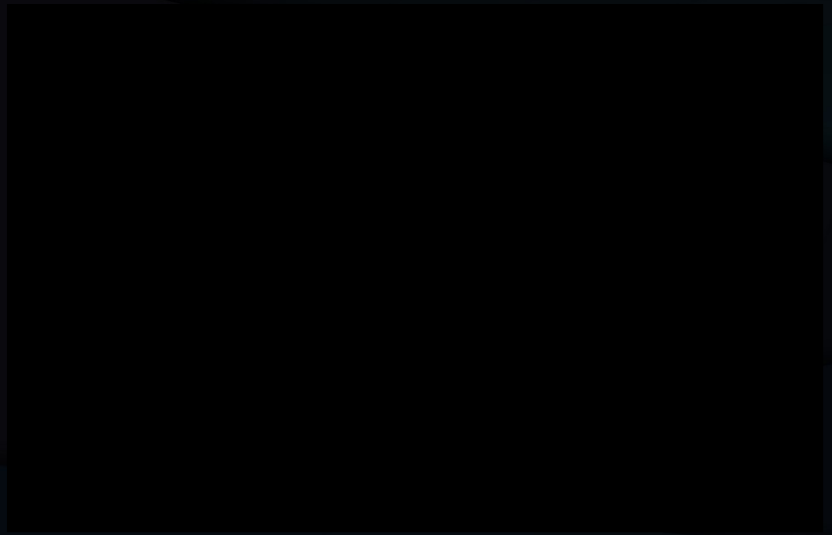


GUI window dressing: Look & Feel ...

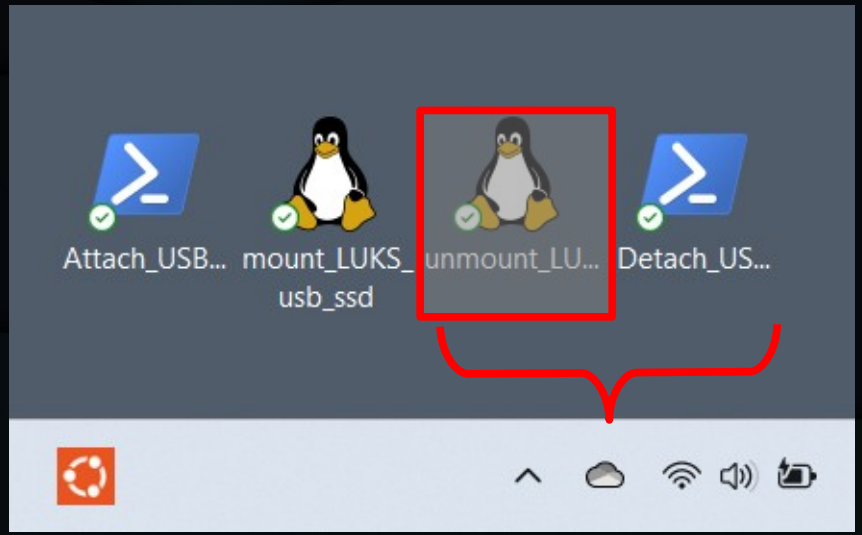
- **Do your stuff ...**
  - create, copy, rsync, delete, rename, etc
  - Linux (subdirectory: /mnt/USB-SSD-Encrypt)
  - Win10+ (look for drive letter such as D:\)



GUI window dressing: Look & Feel ...

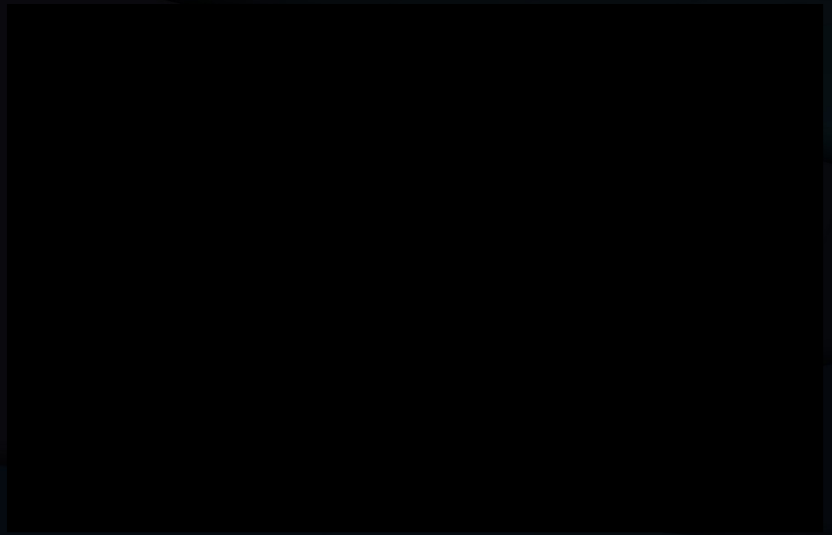


- **Close**
  - **unmount**
    - (type: sudo pswd -Ubuntu)
  - **Detach**
    - (permit elevated ‘Adminstrator’ execution -PowerShell)

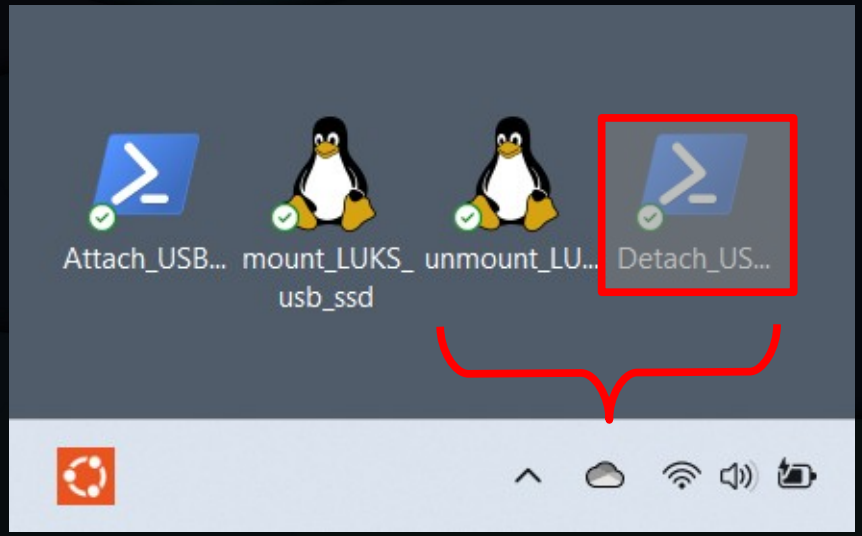


Close

GUI window dressing: Look & Feel ...



- **Close**
  - **unmount**
    - (type: sudo pswd -Ubuntu)
  - **Detach**
    - (permit elevated ‘Adminstrator’ execution -PowerShell)



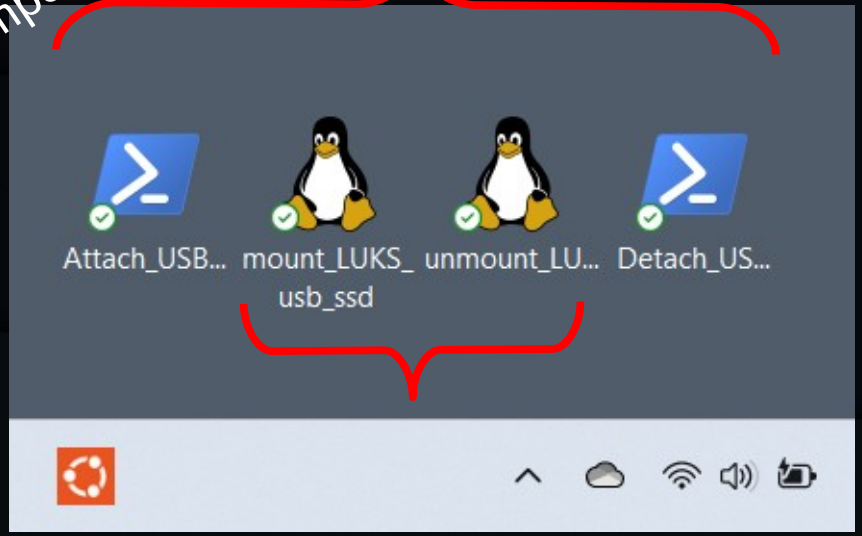
Close

GUI window dressing: Cool !!!

- **Open**
  - **Attach**
    - (permit elevated ‘Administrator’ execution -PowerShell)
  - **mount**
    - (type: sudo pswd, LUKS pswd -Ubuntu)
- **Do your stuff . . .**
  - create, copy, rsync, delete, rename, etc
  - Linux (subdirectory: /mnt/USB-SSD-Encrypt)
  - Win10+ (look for drive letter such as U:\)
- **Close**
  - **unmount**
    - (type: sudo pswd -Ubuntu)
  - **Detach**
    - (permit elevated ‘Administrator’ execution -PowerShell)

The sequence IS important !!!

Win10+



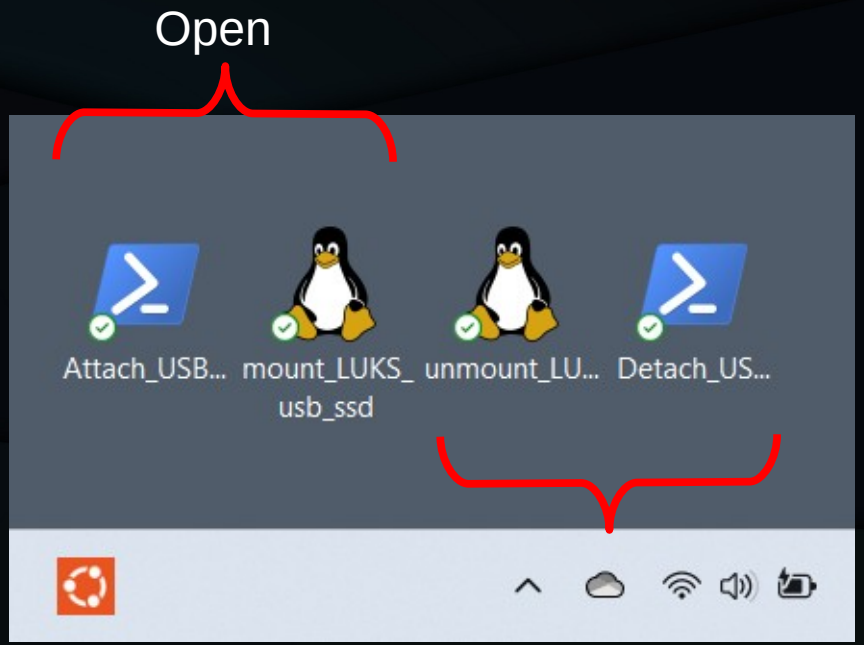
WSL2 / Ubuntu

GUI window dressing… How?

...the 4 scripts behind the curtains

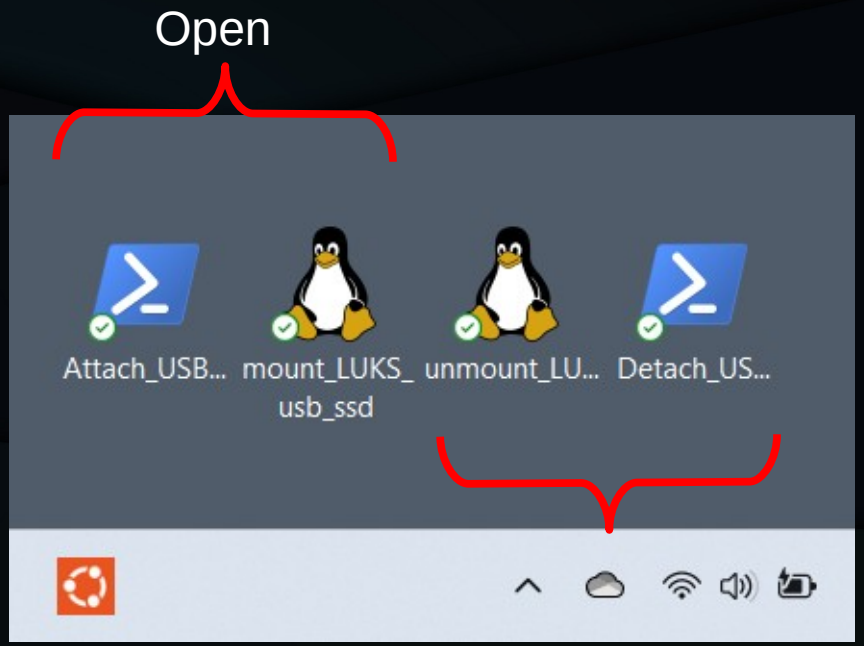
GUI window dressing: How?

- PowerShell & bash scripts (*.ps1 & *.sh files)
  - simple text files
  - create using a Notepad, NotePad++, gedit, etc.
- Windows desktop shortcuts (*.lnk files)
  - binary files
  - create by mouse clicks & the usual M$-Window user contortions (interactions).
  - … your homework !



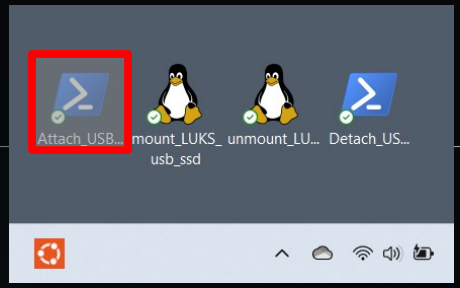
GUI window dressing: Scripts (4) ...

- **PowerShell mount script (.ps1)**
- **Ubuntu luksOpen script (.sh)**



- **Ubuntu luksClose script (.sh)**
- **PowerShell unmount script (.ps1)**

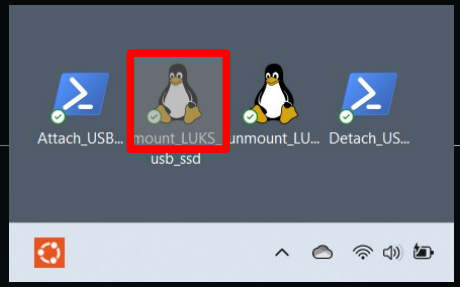
PowerShell mount script:




```
# C:\Users\<user>\MyFiles\ShellScripts\mount_USB-SSD.ps1

# A Power Shell script to connect a USB-SSD partition to a device in WSL2 such as '/dev/sdd1'

# Launch using:
# powershell.exe -ExecutionPolicy Bypass -File C:\Users\caena\mount_USB-SSD.ps1
# otherwise it will be displayed using NotePad, if launched using just the file name.

$DriveFound = (Get-CimInstance Win32_DiskDrive | Where-Object {$_.Caption -like "Samsung*"} | Select-Object -ExpandProperty DeviceID)
# echo "Run the command 'wsl.exe --mount --bare $DriveFound"
wsl.exe --mount --bare $DriveFound
msg * The USB-SSD: $DriveFound will now be accessible in WSL2!
```

```
# To fix the File associativity:
# Choose the method that suits you best. If you want to change the file association so that .ps1 files
#    are executed by PowerShell by default, you can change the default program for .ps1 files to PowerShell. To do this:

# Right-click on a .ps1 script file.
# Choose "Open With" > "Choose Another App."
# Select "More apps" if PowerShell isn't listed.
# Scroll down, select "Look for another app on this PC."
# Navigate to C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe and select it.
# Check the box that says "Always use this app to open .ps1 files."
# Click "OK."
# After doing this, double-clicking a .ps1 file should execute it in PowerShell by default.
```

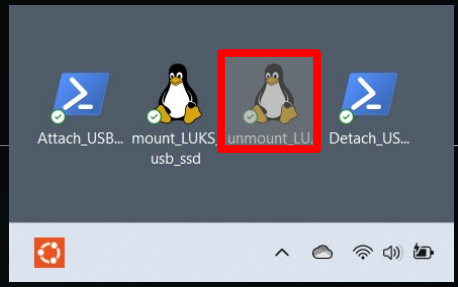
bash luksOpen script:




```
#!/bin/bash

# U:\home\<user>\MyFiles\ShellScripts\mount_luks_usb_ssd.sh
# /home/<user>/MyFiles/ShellScripts/mount_luks_usb_ssd.sh

#Launch using:
#  C:\Windows\System32\wsl.exe bash -c
/home/ks/MyFiles/ShellScripts/mount_luks_usb_ssd.sh

get_sudo_password() {
zenity --password --title="Enter sudo (WSL2) password" --width=400
}

get_LUKS_passphrase() {
zenity --password --title="Enter LUKS Encryption passphrase" --width=400
}

show_message() {
zenity --info --text="USB-SSD-Encrypted drive has been mounted." --title="Mount
Successful"
}
```

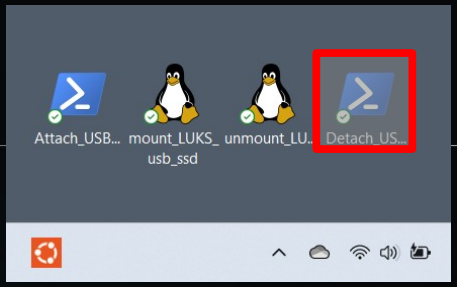
```
# Ask for sudo password
sudo_password=$(get_sudo_password)

# Ask for LUKS passpharse
LUKS_passphrase=$(get_LUKS_passphrase)

# Open LUKS encrypted partition
# echo $sudo_password | sudo -S cryptsetup luksOpen /dev/sdd1 USB-SSD-Encrypted
echo $sudo_password | sudo -S -v
sudo -S cryptsetup luksOpen /dev/sdd1 USB-SSD-Encrypted <<< $LUKS_passphrase
# echo "CryptSetup done ..."

# Mount the decrypted partition
sudo -S mount /dev/mapper/USB-SSD-Encrypted /mnt/USB-SSD-Encrypt
# echo "Mounted Encrypted partition"

# Show success message
show_message
```

bash luksClose script:




```
#!/bin/bash

# U:\home\<user>\MyFiles\ShellScripts\unmount_luks_usb_ssd.sh
# /home/<user>/MyFiles/ShellScripts/unmount_luks_usb_ssd.sh

#Launch using:
#  C:\Windows\System32\wsl.exe bash -c /home/ks/MyFiles/ShellScripts/unmount_luks_usb_ssd.sh

get_sudo_password() {
zenity --password --title="Enter sudo (WSL2) password" --width=400
}

show_message() {
zenity --info --text="USB-SSD-Encrypted drive has been un-mounted." --title="Un-Mount Successful"
}

# Ask for sudo password
sudo_password=$(get_sudo_password)

# Un-Mount the decrypted partition
echo $sudo_password | sudo -S -v
echo $sudo_password | sudo -S umount /mnt/USB-SSD-Encrypt
echo "Un-Mounted Encrypted partition"

# Close LUKS encrypted partition
echo $sudo_password | sudo -S cryptsetup luksClose USB-SSD-Encrypted
echo "CryptSetup un-done ..."

# Show success message
show_message
```

PowerShell unmount script:




```
# C:\Users\<user>\MyFiles\ShellScripts\unmount_USB-SSD.ps1

# A Power Shell script to disconnect a USB-SSD partition from a device in WSL2 such as '/dev/sdd1'

# Launch using:
# powershell.exe -ExecutionPolicy Bypass -File C:\Users\<user>\unmount_USB-SSD.ps1
# otherwise it will be displayed using NotePad, if launched using just the file name.

$DriveFound = (Get-CimInstance Win32_DiskDrive | Where-Object {$_.Caption -like "Samsung*"} | Select-Object -ExpandProperty DeviceID)
wsl.exe --unmount $DriveFound
msg * The USB-SSD: $DriveFound disconnected from WSL2!
```

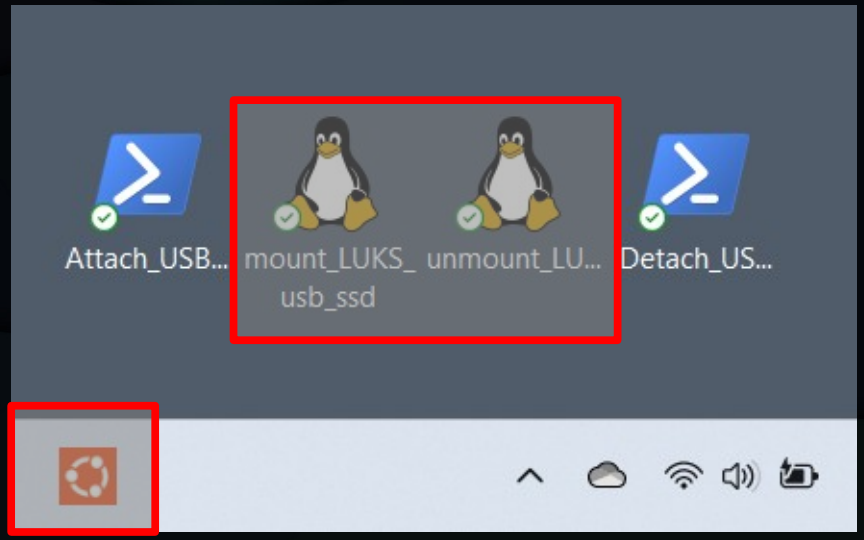
GUI window dressing: Win10

Only Win10 Clickable Desktop shortcuts / icons

Cannot create Ubuntu icons

- Run a bash script in a terminal

This is a limitation & disappointment !



WSL2 / Ubuntu icon →

LUKS Details & Notes

LUKS Details:

LUKS is an encryption layer on a block device, so it operates on a particular block device, and exposes a new block device which is the decrypted version. Access to this device will trigger transparent encryption/decryption while it's in use.

It's typically used on either a disk partition, or a LVM physical volume which would allow multiple partitions in the same encrypted container.

LUKs stores a bunch of metadata at the start of the device. It has slots for multiple passphrases. Each slot has a 256 bit salt that is shown in the clear along with an encrypted message. When entering a passphrase LUKS combines it with each of the salts in turn, hashing the result and tries to use the result as keys to decrypt an encrypted message in each slot. This message consists of some known text, and a copy of the master key. If it works for any one of the slots, because the known text matches, the master key is now known and you can decrypt the entire container. The master key must remain un-encrypted in RAM while the container is in use.

Knowing the master key allows you access to all the data in the container, but doesn't reveal the passwords in the password slots, so one user cannot see the passwords of other users.

The system is not designed for users to need to know or interact with the master key, and this key can't be changed without re-encrypting. The use of password slots means that passwords can be independent of the encryption key: they be changed without re-encrypting the entire container, and there can be multiple password slots.

https://askubuntu.com/questions/805485/how-does-luks-work

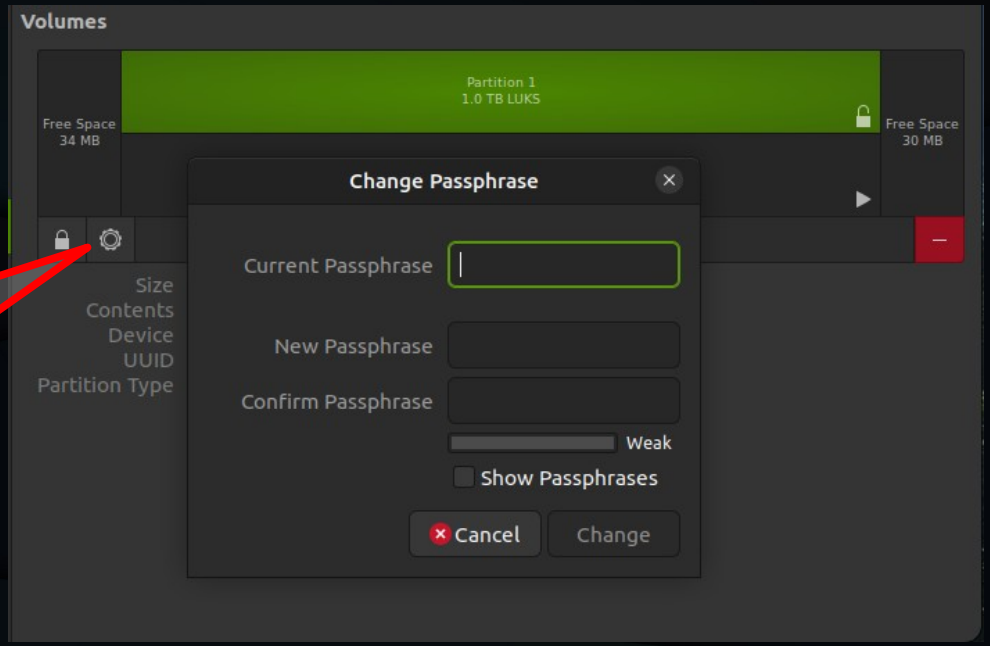
LUKS Change Passphrase:

- Menu → Preferences → Disks

  or

- gnome-disks &

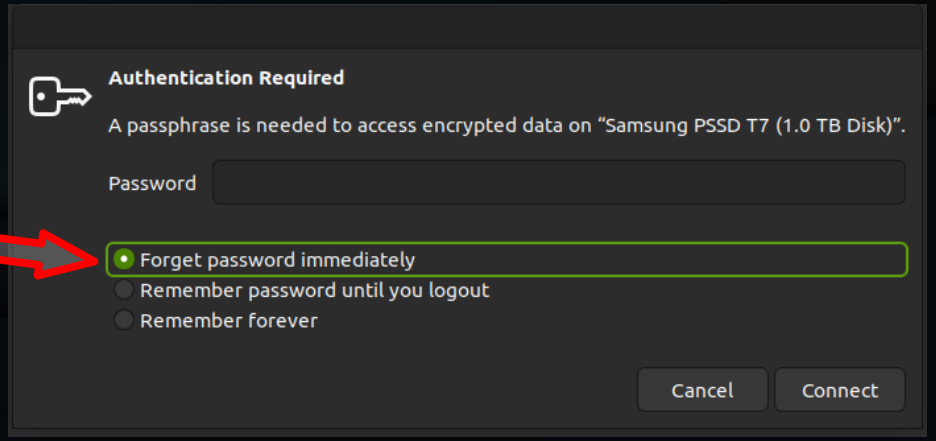
  - Select the appropriate disk
  - Select the LUKS partition
  - Click the ‘gear’ icon
  - Select ‘Change Passphrase’ from the pop-window
  - Type the current and new passphrases



LUKS Passphrase Retention: Options

- When the LUKS passphrase prompt appears, select:
  - ‘Forget password immediately’
  - for the most security.
- If you select:
  - ‘Remember password until you logout’
  - it stays in RAM and is less secure.
- Never select:
  - ‘Remember forever’
  - since it is stored in the hard drive of your computer, and is the least SECURE !!!
  - File: ~/.gnome2/keyrings/login.keyring

    https://askubuntu.com/questions/615408/how-to-disable-remember-forever-option-in-mounting-encrypted-disks



- gnome-keyring-daemon -r

  (resets the passphrase prompt)

LUKS Forget Passphrase:

gnome-keyring-daemon -r

** Message: 12:00:00.000: Replacing daemon, using directory: /run/user/1000/keyring

GNOME_KEYRING_CONTROL=/run/user/1000/keyring

SSH_AUTH_SOCK=/run/user/1000/keyring/ssh

References:

  - https://github.com/dorssel/usbipd-win/releases
  - https://devblogs.microsoft.com/commandline/connecting-usb-devices-to-wsl/
  - https://learn.microsoft.com/en-us/windows/wsl/connect-usb# attach-a-usb-device
  - https://search.brave.com/search?q=Win11+allow+non-Microsoft+store+apps+from+Github+to+be+installed&source=web&summary=1&summary_og=593139f15f9688e9692e8f
  - https://github.com/jovton/USB-Storage-on-WSL2
  - https://trendoceans.com/mount-luks-encrypted-drive-partition-in-linux/
  - https://superuser.com/questions/524822/awk-equivalent-functionality-on-windows
- Complicated method, which involves compiling a custom kernel:
  - https://github.com/jovton/USB-Storage-on-WSL2
- Simpler method:
  - https://devblogs.microsoft.com/commandline/connecting-usb-devices-to-wsl/

References (LUKS):

- https://www.ibm.com/docs/en/order-management-sw/10.0?topic=considerations-encrypting-data-partitions-using-luks
- https://blog.elcomsoft.com/2020/08/breaking-luks-encryption/
- https://security.stackexchange.com/questions/251176/is-luks-still-an-effective-option-for-consumer-fde-considering-elcomsoft-can-bre
- https://askubuntu.com/questions/95137/how-to-change-luks-passphrase

Questions,comments or rewards:

? ! $