USHBY
TECHNOLOGY ONLY

Our affiliates

[Tutorial, Walkthrough] Linux Kodachi 101 - Part2

Details

Written by: Ushby Technical Team

Category: Tutorials

Published: 04 February 2023

Hits: 719

USHBY 101 ON
LINUX KODACHI
PART2
A full walkthrough on how to protect yourself using the operating system.
With guiding concepts on data protection.

Notes :

- In continuation of *Part 1*.
- For installation, go to the end of this book.
- Solutions here are only Linux and Mac OS compatible, we have the perspective that you are running a Linux OS, if you don't, you can still read through and apply this knowledge. Although, you might not completely apply the last part which is the **Installation**.

Written by ***Ushby Technical Team***

Published for free by ***Ushby Organization***

Credit goes to ***Warith Al Maawali***, ***DIGI77*** and ***Ushby*** - 2023.

For credits, official details and more info, visit **https://ushby.org** or **https://www.digi77.com**

Privacy Vs Anonymity

We are concerned here with two things that aren't consecutive to each other, they don't need and certainly don't complete one another. Mixing Privacy with Anonymity is a heavy burden on both your network and browser. Yes it'll do two jobs at once and harmonize the traffic, but it slows down the network and brings suspicion. One need not apply both these two every time one needs security. There are three models that worth mentioning here, they are classic but they live forever, they are:

1. **Anonymity With Privacy**
2. **Anonymity Without Privacy**
3. **Privacy Without Anonymity**

Anonymity is hiding your face, wearing a mask. No one can identify you or link you to other elements that can describe a bigger picture of who you are, where you come from or any elements that are stuck with you once you put that mask off.

Privacy never says anything about hiding who you are, some actually use VPN in a way that is compromising without them knowing, they tunnel their way out of the network in order to go to a destination point which can tell who they are. Therefore your ISP knows where you went, they just don't know what you did. Yes, even with VPN. VPN only makes it harder for the ISP to know enough about the activity, but they can track you, if they really want to.

A good example is back in my country. We have security checkpoints of armed militias. And, we have tribal gangs that catch people on sight, based on where they come from, to retaliate for some stuff or get ransom.

If you have a license plate, a skin color or an accent that tells where you come from, then taking a different route to that place or entering from a different highway won't delude them about you. And, that's you under VPN, you visit sites while carrying your license plate and skin color. The sites won't care where did you enter from. They know who you are because they can link your characteristics to a certain place, from those, they might even suspect what routes did you take.

Although, with the security check points.. it's different. These armed militias want to know where are you going to ? They worry about that more than the retaliators do. They worry what you might bring back, what you might take out and what influence you could have by reaching certain places. They would like to know particulars about your trip, they want to know more about what you do, not who you are. Your activity is their concern, not you. Therefore having a different road is better, you wouldn't have to run into those guys, you exit the district from different roads and gateways, that are built by other people as a solution for that. That's VPN.

With that being said, there are times when you want to hide everything, you don't want the check points to know where are you going, and neither let the retaliators know where are you coming from. That model is the first one mentioned which is **Anonymity With Privacy.** These cases do occur and sometimes you want to block everything from getting to you. This process is harder than what it seems and requires an interchangeable work. Kodachi does that by building **TorCrypt** over **Tor** Over **DnsCrypt** Over **VPN** over your **ISP**. Of course this will consume the network and require a lot of performance, even though Kodachi script manages to keep the connection stable, time will be always less efficient. One more crucial thing this could bring up, is you attracting suspiciousness, you run into websites, networks and tunnels that can see that you are hiding both your identity and activity, they see what's like a ghost running around. It attracts attention, because it makes people ask the one question of "*Why he needs that much security ? what is he/she hiding ?*"

Since time becomes a problem and attention is a bad thing, you would have to optimize your work. Don't ask for things at a time where you don't need them.

There are two models that can do it for you without the first one mentioned. Those two are **Anonymity Without Privacy & Privacy Without Anonymity.** Sometimes you would like to hide who you are and what you are doing, but your activity isn't linked to your identity, and a stranger will never understand that such activity is done by you, because there is nothing in it that is specific to you. It could be anyone who is interested in such activity. Therefore for that we use **Tor** only. We [illegible] up anonymous to any website we visit, but every web site we visit knows what we do.

In the last one, which is **Privacy Without Anonymity** we don't hide the facts about who we are, but we don't want others to know what we do, we use **VPN** for that. Nevertheless we need to be aware of certain factors here, these factors are:

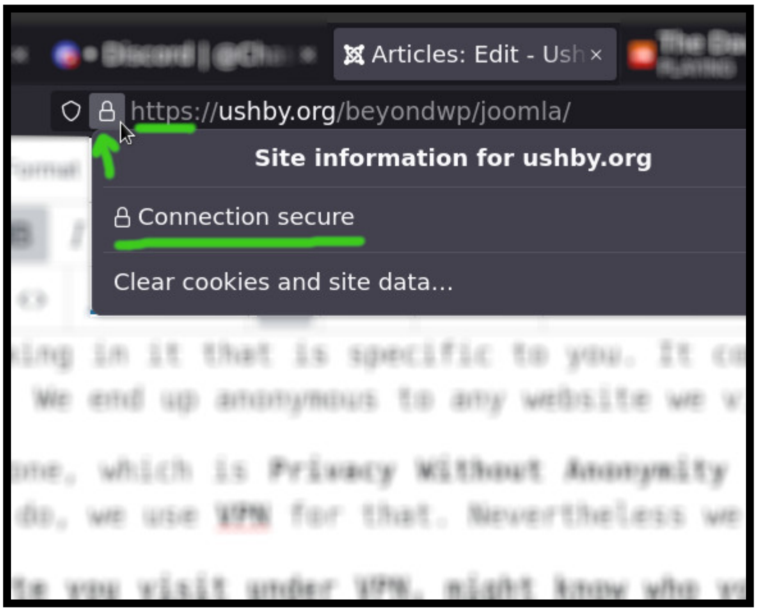
1. **Any website you visit under VPN, might know who you are and what you do.**
2. **Any website you visit under VPN, might know where you come from.**
3. **You can hide from the ISP, you can't necessarily hide from web applications.**
4. **VPN is a way to deal with your ISP, not a way to deal with the internet.**
5. **Any VPN you use, can see the title, and links of the pages you are visiting. As well as searches you are conducting.**

6. **The only thing the VPN provider can't see is the data sent and received, such as files, messages and displayed info.** *[As long as the VPN runs a certificate authority]*

7. **If you use a VPN that has no certificate authority (CA), then you are prone to be spied on and maybe handed false data.**

8. **In case 7, the VPN provider can easily do the well known Man In The Middle Attack.**

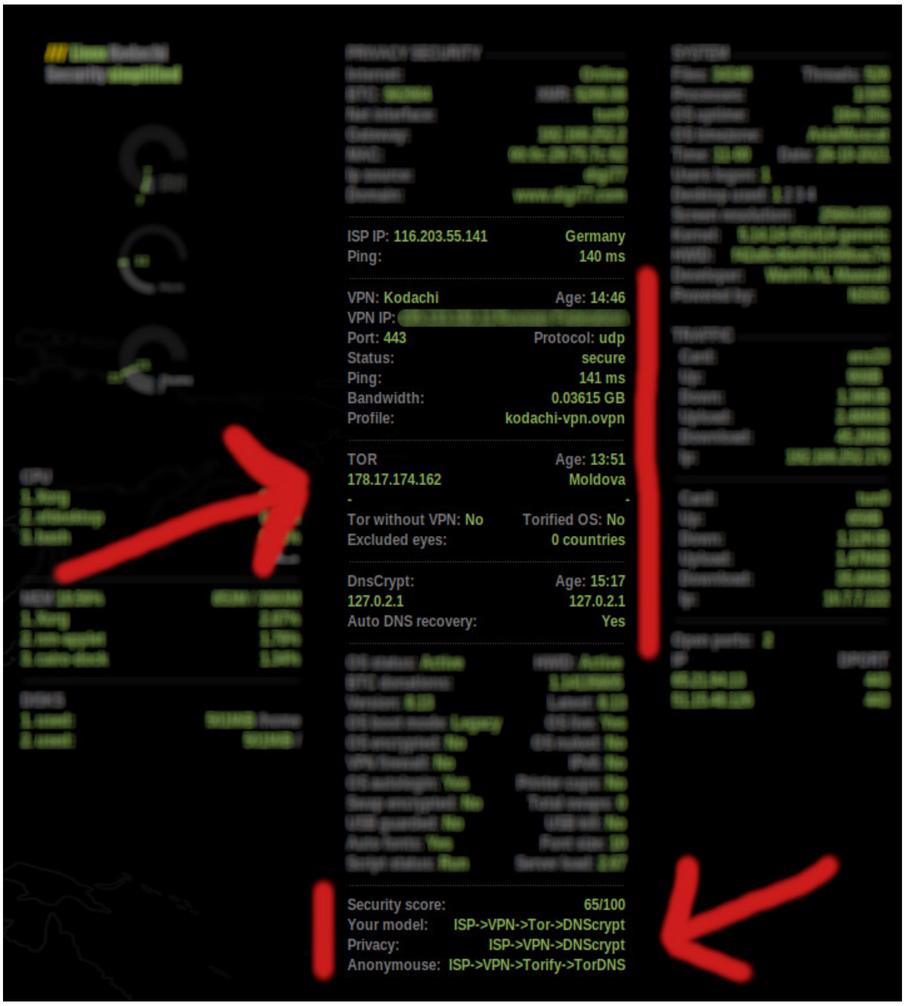
[To verify that your VPN providers has a valid certificate authority, keep aware of your SSL status in your browser, as long as it is green and indicating a valid HTTPS, you are good.]


Articles: Edit - Ush ×
https://ushby.org/beyondwp/joomla/
Site information for ushby.org
Connection secure
Clear cookies and site data...

As you can see in the last model, VPN alone is tricky, sensitive and require more planning. Not only they are this sensitive, but you would need to know whether your VPN is actually a VPN. A lot of VPN service provider just tunnel you out, but they don't have a certificate authority that is concrete, they only change your IP but never your DNS.

Another concern is, even if your DNS is not leaking.. how can you be sure that your VPN provider won't give you up ?

One should be educated and be clear with what they want, in order not to end up completely compromised. We shall understand more as we go through this tutorial.



```
ISP IP: 116.203.55.141                 Germany
Ping:                                   140 ms

VPN: Kodachi                        Age: 14:46
VPN IP: [redacted]
Port: 443                       Protocol: udp
Status:                                 secure
Ping:                                   141 ms
Bandwidth:                          0.03615 GB
Profile:                    kodachi-vpn.ovpn

TOR                                 Age: 13:51
178.17.174.162                         Moldova
-                                            -
Tor without VPN: No           Torified OS: No
Excluded eyes:                     0 countries

DnsCrypt:                           Age: 15:17
127.0.2.1                            127.0.2.1
Auto DNS recovery:                         Yes

Security score:                         65/100
Your model:       ISP->VPN->Tor->DNScrypt
Privacy:               ISP->VPN->DNScrypt
Anonymouse:  ISP->VPN->Torify->TorDNS
```

What can you add ?

FOR EDUCATIONAL AND PEN TESTING PURPOSES ONLY

We give such analysis and guides only for educational purposes, we believe that such methods can be included in a penetration tester environment.

1. Kali

As mentioned in ***Part 1***, Kodachi has an opposite approach to Kali Linux, it focuses on protecting, securing and hiding. Kali on the other hand focuses on attacking, destroying, and stealing. Of course people call it a *Pen Testing* OS which is true, but we are technically speaking here.

What happens when you use Kodachi and Kali together is what I like to call the **Bullseye.** Using *dmitry*, *nmap*, *sql injections*, *DDOS* attacks or trying to ruin the server, doesn't go without traces left. When one attacks a network/system and protects themselves at the same time, means they've gotten it right on point. They can go from over here to over there, grab and manipulate or remove what they want, while remaining shielded. Therefore having Kodachi and Kali together, is good for testing forensic abilities of your server. To see whether your server can recognize attacks coming from different exit nodes, and whether you can trace back requests or not.

Kali is best installed inside an **Oracle VirtualBox** when ran inside Kodachi. The network settings in the Virtual machine should be set to *Bridged Mode*. Every scan or tools used in this case are going under VPN, TOR and DNS Encryption. You can run an extensive nmap scan to a web service all you want. If a **SYN** request gets sent and **Ack** one comes back from the server that you're scanning, it gets received through the TOR. You still see the server, but the server can't see you. One of the greatest fears of a Hacker is to get caught running scans, because what would they do that for ?

Word Press for example has the famous *Word-Fence Plugin*, a security system that lets the admins know, who attempted to login, from which IP and what did they enter. Attempts get blocked from one false attempt if the admins chooses so. With a Bullseye, you can switch it in a second and, that won't necessarily be applied. If one is building security plugins for web services, having Kodachi can be the forensic pen tester rather than the security pen tester that is known as Kali.

Because Kodachi helps Torify the entire system, every tool becomes handy for Kali. For someone that is running a new website on a shared or a local server without paying attention to file sizes of his directory. Can end up destroyed by random **wget** and **curl** requests, coming from 3 to 4 Kali OS's installed inside one powerful machine, each will have it's own exit node IP, the server won't understand what is happening. All this with no traces whatsoever.

2. Android Studio (GPS Location Changer)

The Android studio which is mainly made for developers to create and debug android software. Can also run a virtual machine as a mobile inside your Linux. The strong thing is that this Virtual Machine settings allow you to change your location input to any app you are using. This location change isn't an IP reading, it's a GPS reading. Most apps don't suspect the user to be able to change their GPS input, specially from a recognized kernel environment as Google/Android.

3. VMs

Any virtual machine you run is under the Kodachi network. Therefore, VM's can be used as an isolation method. Crypto wallets for example, can be compromised if they get connected from your original network even for one time. To make sure that you continue to use Kodachi separately and, that your wallet is accessed only from a certain IP and Location.. you only run it after establishing the connection first. Everything gets isolated. This same methodology can be implemented for messaging solutions.

Things to never do:

1. **Install *snapd* (Snap Package Manager).**
2. **Configure a new user.**
3. **Create any *systemd* units or services.**
4. **Connect two VPN's at the same time.**
5. **Run commands that starts with** `sudo curl`
6. **Install Kodachi next to another OS without having *a Luks* encryption**
7. **Using the same encryption password of the OS that is living next to Kodachi.**
8. **Install/Run WineHQ or any Windows emulator**

How to Contain and Isolate

- **Personnel Containers**

Linux security specialists claim that containers are not a security feature, that's because they do not focus on isolating, but rather focus on managing the different entities. But nevertheless, this management can serve a security purpose to one, as long as they know what are they out there to protect.

A container concept in the computer world and specially in Linux, have started with Docker. Docker fixes the same problem that landlords today have, which is the common kitchens that cause interference and, an increase in cost if one is to set one for each flat, when the building is of an old structure. Therefore, new buildings now have a different architecture of pipes, electricity and ventilation, to allow any small room to have a small built in kitchen, that serves only what the rentor lessee needs. This concept is what happens with today's Linux, no need for a VM all the time. Just run a new verion of whatever app for whatever new user that comes in, and we get to keep our resources.

[illegible] what does this have to do with Security and Privacy, and let alone Kodachi ? Well, containers never split a thing onto two pieces, they just produce a new introduction of whatever you have, under any claims you want. And since we are here trying to solve the problem of privacy, we are presented with a few aspects:

1. **Accessibility**

   The first idea that would concern someone, who is trying to protect something, is to put it a distance from people or put people at a distance from it. As long as it is not reached, it won't be taken or tampered with. Securing something for oneself, where accessibility is only a privilege of that person that succeeded in securing whatever they want.

Even without IT, accessibility is not always physical. Information can access others through our sensors, that are voice, image and smell. From here you start to wonder, what voice, image and smell have to do with Kodachi and personnel privacy.

Voice travels through the network. Image can travel as well -In the form of text and pictures-. Smell on the other hand is just us being fair to human nature. Let us focus on voice and imagery. The person by default, has both voice and imagery as a one entity, but because he can present them to different people, in different areas, through different fields.. will allow him to have two personalities. The latter opens doors to new presentations. Now, one can change their name, location, age and the third person won't know that there is any false.

With Kodachi, we can create two packages of notes. Notes that specify a Tor exit node to one country for each personality, a different VPN and different emails/accounts. If the features in the dashboard can be mastered correctly, nothing in the network will put the personalities in conjunction with each other.

2. Recognizability

But there are cases where a compromisation have occurred with some of my friends, that applied these tactics. It is because there is a factor of recognition. A person who already seen your image and heard your voice at the same time, can recognize the other when evgen when it's not present, which will stop your claim from achieving its purpose.

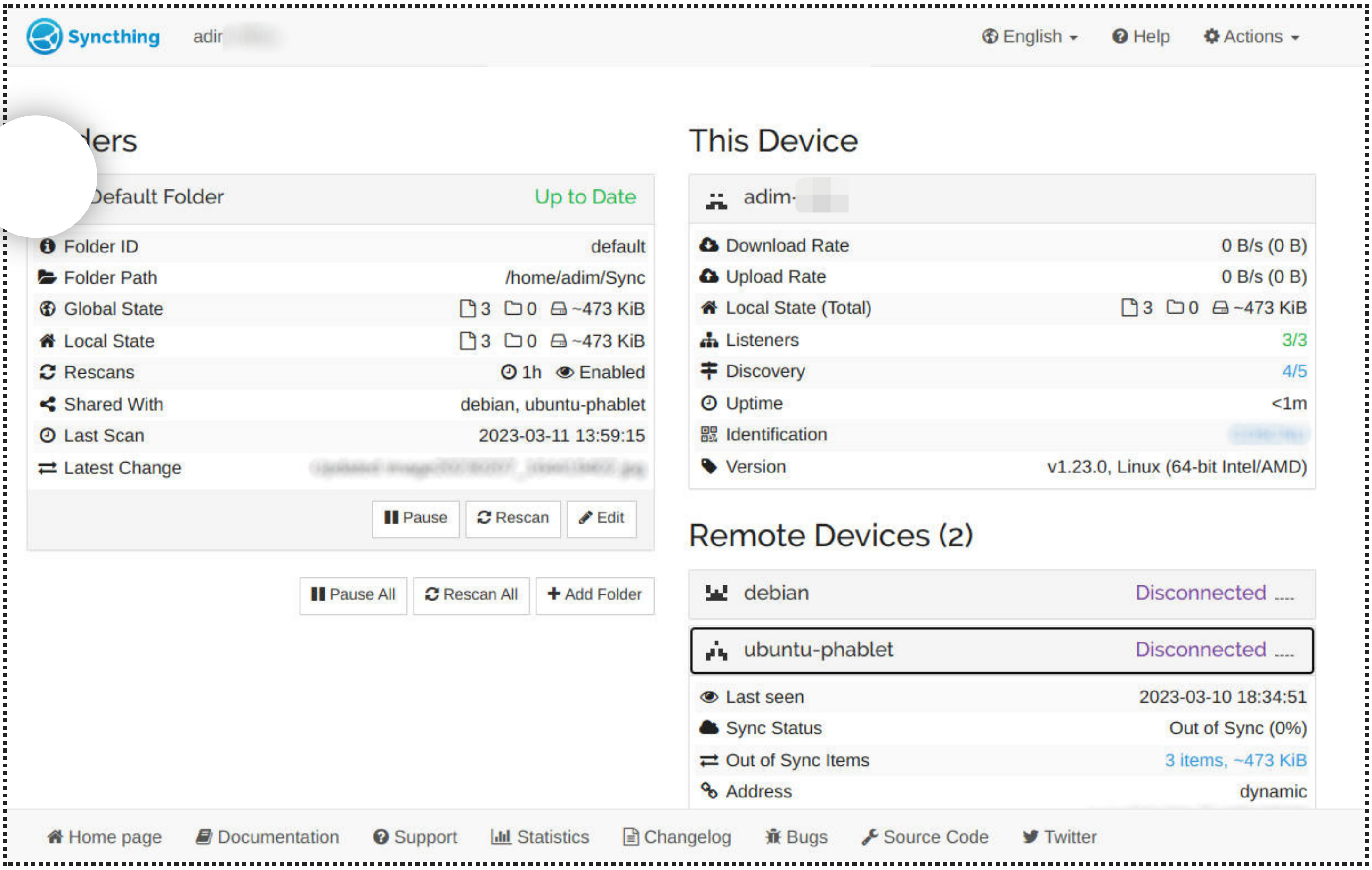
3. Internalization of External Elements

This hard factor of recognition and its surrounding elements, can be better handled if these things were organized, listed and then internalized. In other words, instead of having them outside, we have them inside.

This in itself will create another container, a container of the people or the areas that are like that.

Reflection on The Security Tools

▪ Syncthing



lders

Default Folder — Up to Date

Folder ID	default
Folder Path	/home/adim/Sync
Global State	🗋3 🗀0 🖴~473 KiB
Local State	🗋3 🗀0 🖴~473 KiB
Rescans	1h Enabled
Shared With	debian, ubuntu-phablet
Last Scan	2023-03-11 13:59:15
Latest Change	[redacted]

This Device

adim-[redacted]

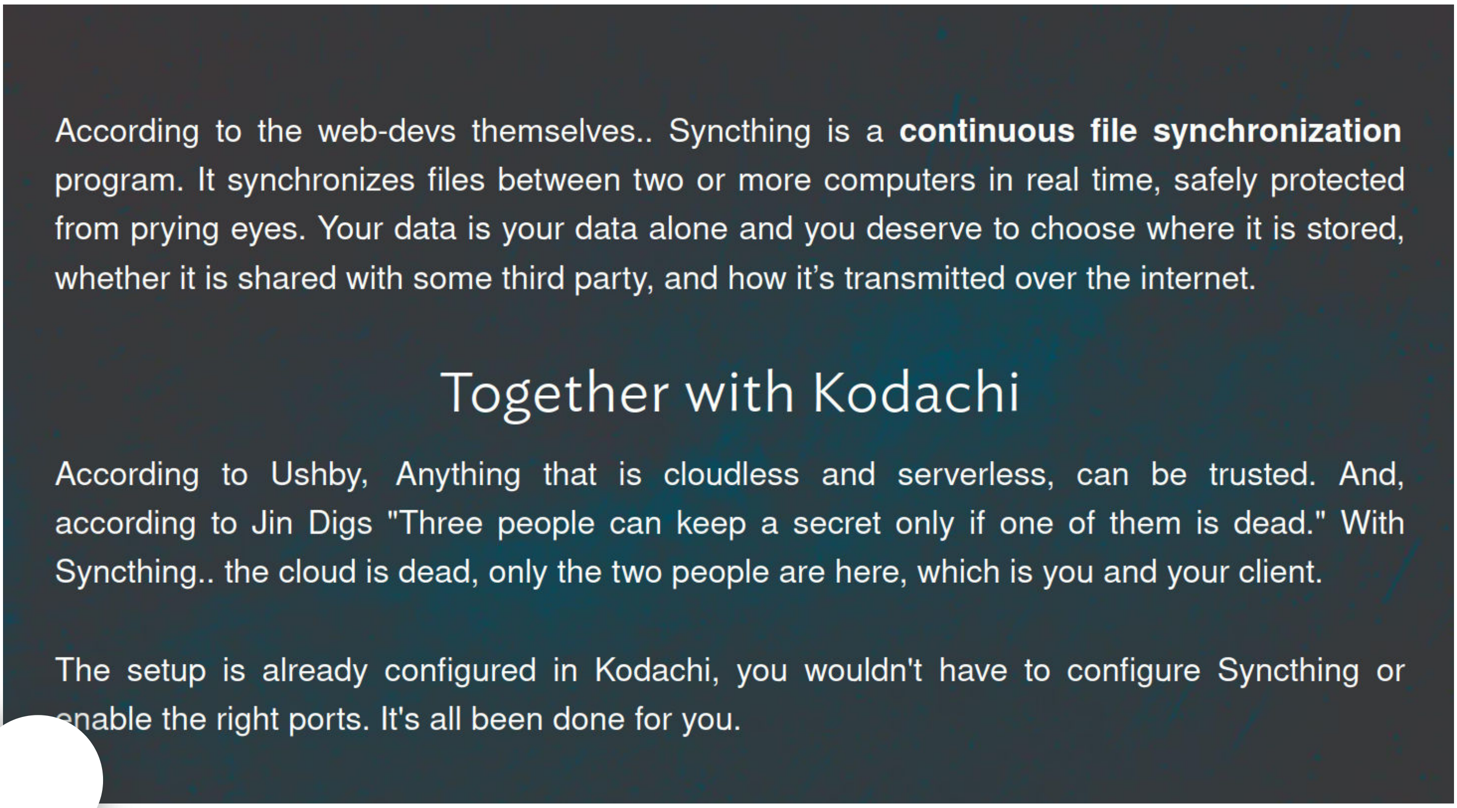
Download Rate	0 B/s (0 B)
Upload Rate	0 B/s (0 B)
Local State (Total)	🗋3 🗀0 🖴~473 KiB
Listeners	3/3
Discovery	4/5
Uptime	<1m
Identification	[redacted]
Version	v1.23.0, Linux (64-bit Intel/AMD)

Remote Devices (2)

debian — Disconnected

ubuntu-phablet — Disconnected

Last seen	2023-03-10 18:34:51
Sync Status	Out of Sync (0%)
Out of Sync Items	3 items, ~473 KiB
Address	dynamic



According to the web-devs themselves.. Syncthing is a **continuous file synchronization** program. It synchronizes files between two or more computers in real time, safely protected from prying eyes. Your data is your data alone and you deserve to choose where it is stored, whether it is shared with some third party, and how it’s transmitted over the internet.

Together with Kodachi

According to Ushby, Anything that is cloudless and serverless, can be trusted. And, according to Jin Digs "Three people can keep a secret only if one of them is dead." With Syncthing.. the cloud is dead, only the two people are here, which is you and your client.

The setup is already configured in Kodachi, you wouldn't have to configure Syncthing or enable the right ports. It's all been done for you.

Note: snap can only be used from the remote device, not the Kodachi host, Kodachi says snap is not safe.

- **VeraCrypt**

A tool which you can never hate on. Depends on what you like, you can go for the CLI or the GUI, I personally prefer the CLI, because it is fast. The bad thing about the GUI version is that, if you install it, you can't use the CLI anymore, because the tool will pop up everytime. So make your choice from the beginning.

The encryption that is offered here should never be underestimated, because it can affect all what's around your system. Common sense will tell you that it encrypts files, when that's not true. It actually encrypts the box that holds them together, even in the logo they put a lock as a symbol. Some other encryption tools do encrypt the files themselves, here no. So when you have such an approach, things get faster. It's not only separate files which you can secure, but you can secure binaries, operating system images, docker containers, and vm disk spaces.

If you put side-links on your file manager to what's inside of the container, and decrypt the main directory with a specific pattern.. then you can work faster. Because, every time you decrypt the links, they will automatically start working. Veracrypt decrypts partitions as (**veracrypt1, veracrypt2, etc**), so the path in the symbolic links can always remember that.

However, errors do occur, so make sure you don't turn off the computer before dismounting the mounted encrypted containers or partitions. If you continue to do that over time, the encryption gets corrupted, and there are cases where you lose everything, so just make sure you follow procedure.

One quick way to dismount everything before shutting down, is to run : **sudo veracrypt -d ,** however, files which are inside the encryption and are still open, will prevent dismount, so you need close them first.

It's been a long journey with Veracrypt, and I don't know what I would have done without it. If the world sees what I encrypted, they might give me the chair, so I liked this one and liked it quite well.

Computer can't create random inputs. Even with programming language such as JavaScript, random inputs aren't actually random, they are just so many. So for such a reason, VeraCrypt asks you to put mouse movements as inputs of a human, which then gets added into the encryption. If you were to use the CLI, you can input random keyboard strokes. Once I've seen that solution, I knew this tool wasn't messing around.

There also encryption schemes which you need remember or write down on a paper, because if you forget them you could lose access to the files. Below in this book, there is more description of this tool.

ZuluCrypt

The table below should do enough to explain to you that zuluCrypt is totaly different than Veracrypt, I shall only add that the former focuses more on hard-disks, rather than just file containers. So for the binaries, vm disk space and all.. you should stick to Veracrypt.

zuluCrypt
1. Uses Qt GUI toolkit. 2. Is GPLv2+ license, making it compatible with every FOSS repository. 3. Can create and unlock LUKS, plain dm-crypt,TrueCrypt and VeraCrypt volumes. 4. Use of PIM value when creating a VeraCrypt imposes no password requirement, allowing users to create a VeraCrypt volume with a PIM value of "1" and an empty password.

5. Passwords to unlock volume can be stored in kwallet, libsecret and ligcrypt powered-internal-secured-storage-system. In other words,it is better integrated in FOSS desktop environments.
Veracrypt
1. Uses gtk GUI toolkit. 2. Uses custom license and it's forbidden in distributions that cares "too much" about licenses like fedora and debian. 3. Can unlock only TrueCrypt and VeraCrypts, and can create only VeraCrypt volumes. 4. Use of PIM value when creating a VeraCrypt adds restrictions like a password must be atleast certain length. 5. Does not support secure storage of volume passwords.

The table above, was made with the help of the legendary **mhogomchungu**.

jubalh/MAT

Metadata Anonymisation Toolkit

Before you send files, and specially documents that you designed with Libre-Draw, PDF or documents. You should always remove the metadata from them, you can also render them, but that's a different topic. Removing metadata, will prevent anyone who receives the file from knowing when, who and where the files was made.

A lot do so much anonymity work to make a file, but they forget this last step, and everything [illegible]s in vein, or in prison.

- **ExifCleaner**

Similar to MAT, but only for images.

KeePassXC

If you are planning on using any password manager tool, that is either compiled or installs in an non-transparent manner, then you are committing suicide. KeePassXC is however different, it is open source. And, besides KeePassXC, we shall clarify that Open Source from the security perspective, is different that Open Source in the developmental one. Let no one fool you into mixing these two together, because that's been happening a lot, and it works for a lot of scammers to mitigate the lies, which they might throw in your face.

From a security standpoint, Open Source is Open Source, if the code is all available, and the installation scripts are open as well. Open Source in the developmental aspect, is every code which you can develop both technically and legally.

I have participated in a discussion which now I feel bad for, with Ubuntu Touch development team. I nevertheless want to use that unfortunate occurrence in this fortunate one, to benefit you by describing the discussion which happened there.

My claim was that *Ubuntu Touch* is not an Open Source OS. Because first, the Operating System wasn't available in a way that proves it is the same one that was on my phone at the time, Due to the fact that the installer tool, didn't show any scripts running while burning the OS into my phone, it was a compiled one, built with *Electron*. At the same time, there was no official documentation that illustrated a recognized pipeline tool, that could be widely known as *CI/CD*. It could have been said that UT was open source from a developmental aspect, but never from a security standpoint, because one can fork the Gitlab repositories and build on them, even from a legal standpoint, but from a security one, no one knows what code goes into each phone.

Alfred, who is a part of the executive team in *UBports -Ubuntu Touch's Assigned Company-,* refuted my claims, only by describing how nothing is fully Open Source, as if even high level languages such as *JavaScript*, *C++,* etc, will eventually compile into a lower-level language/instructions as well. I got irritated from such a reproach towards security concerns. Because they avoided the main topic and go into one that was unknown to one.

With **KeePassXC**, you need not be concerned about Open Source from any of the two aspects. The code is there, and the installation methods are so as well.

> *KeePassXC is a modern, secure, and open-source password manager that stores and manages your most sensitive information. You can run KeePassXC on Windows, macOS, and Linux systems. KeePassXC is for people with extremely high demands of secure personal data management. It saves many different types of information, such as usernames, passwords, URLs, attachments, and notes in an offline, encrypted file that can be stored in any location, including private and public cloud solutions. For easy identification and management, user-defined titles and icons can be specified for entries. In addition, entries are sorted into customizable groups. An integrated search function allows you to use advanced patterns to easily find any entry in your database. A customizable, fast, and easy-to-use password generator utility allows you to create passwords with any combination of characters or easy to remember passphrases.*

Enhancing Methods

- Increasing the screen scale

To increase the size of your Kodachi. We need to apply more than a few changes, so please follow these steps:

- From the main menu of the system on the lower left; enter a search for *appearance*. Then click the **Appearance** link. From the **Appearance**, go to fonts bar.
- Then, change both the **Default Font** and **Default Monospace Font**. (Make it at about **13** for each, or maybe even **15**).
- Then we need to increase the size of the Panels. So, click on the left panel with a right click, and choose **Panel** > **Panel Preferences**. And, from the **Measurements** section, increase the **Row size** to **65 Pixels**. Before closing the **Panel Preferences**.. click the upper bar, and change it, from **Panel 0** to **Panel 1**. After that, apply the same **Pixel** changes.
- Now, we need to go to the **Window Manager**. So, go to the main menu and search for *window*, you'll see the **Window Manager**. After clicking it, choose **Default-xhdpi**. You can also change the size of the title of the window bar. You'll find an option for that as well.

- System Hardening

- Removing the Grub (Bootloader) background (Taken from [https://makeuseof.com)](https://makeuseof.com)

The Grub configuration file or **grub.cfg** is stored in the **/etc/default** folder. You can edit the file using **gedit**, a command-line tool that lets you edit important system files on your computer with minimal risks.

To change the Grub boot menu background through the terminal:

1. Copy the path to the image file.

Open the **grub.cfg** file located in **/etc/default**.

```
gedit /etc/default/grub.cfg
```

3. Append the following line to the file. Note that you must replace the **/path-to-image** with the path that you have just copied.

```
GRUB_BACKGROUND=/path-to-image
```

4. Save the file and close the editor.
5. Update Grub with the new configuration file.

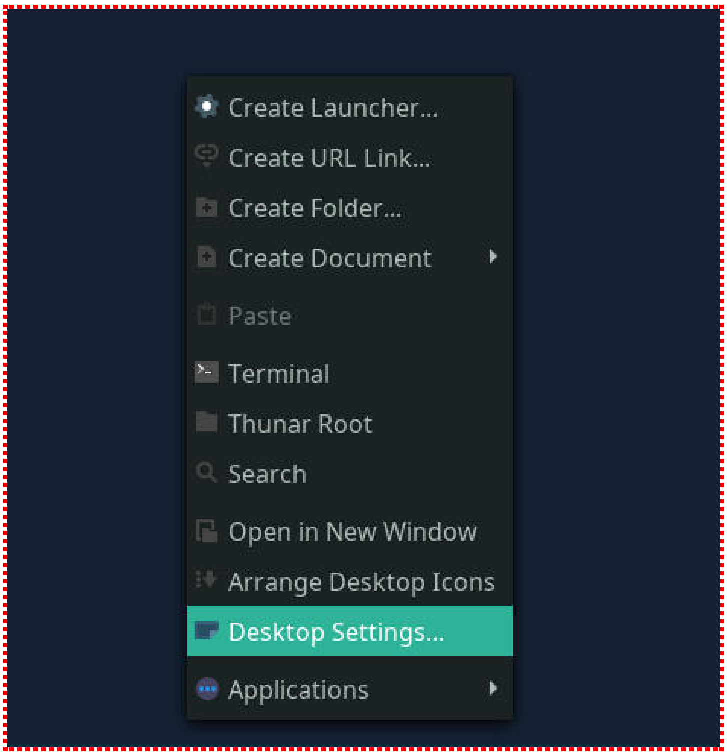
```
sudo update-grub
```

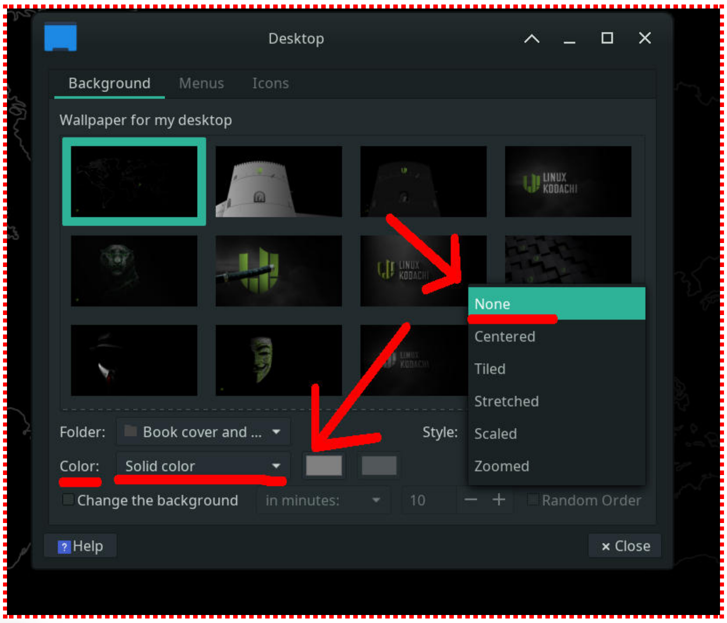
You will see an output that will look something like this. Note that the second line will confirm if Grub has detected the background image or not.

```
Generating grub.cfg &hellip;
Found background image: ~/Pictures/yourpicture.png
Found linux image: /boot/vmlinuz-2.6.39-0-generic
Found initrd image: /boot/initrd.img-2.6.39-0-generic
Found linux image: /boot/vmlinuz-2.6.38-8-generic
Found initrd image: /boot/initrd.img-2.6.38-8-generic
Found memtest86+ image: /boot/memtest86+.bin
done
```

Finally: Reboot your system and check if the background image was successfully changed.

- Removing the Desktop backgrounds





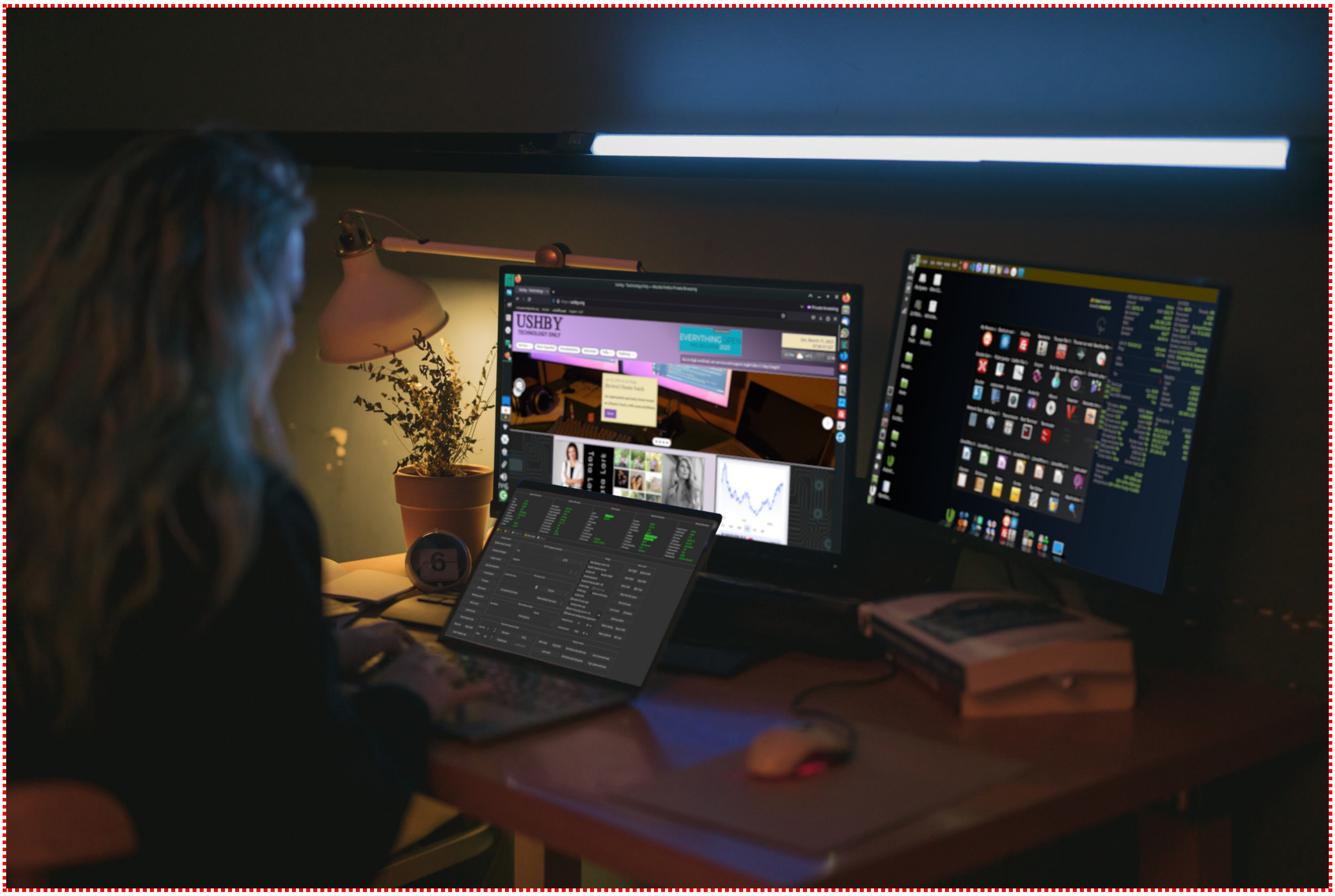
Folder: Book cover and ...

Style: None

Color: Solid color

□ Change the background in minutes: 10 □ Random Order

Arguments



- **HWID**

A lot has been argued in the face of Kodachi together with its HWID pulling technique. A paranoid person will not let his concern be taken away from him, even if they see an evidence. Paranoia takes the best precautions but leads one to nowhere. Kodachi uses HWID identification methods so that it can stop users from bypassing the bandwidth limit, when using Kodachi VPN. If such a method is not applied, none of us will be able to use that feature. But since Kodachi is able to send back such info.. can they really track activity and build an image about the user ? If we answer with either yes or no, your concern will remain in its place. What matters is: what can you do about it, as a skeptic..

The worst case scenario is not just HWID being sent, it is your activity entirely being recorded by Kodachi. But surely you are aware that the developer was honest enough to tell you about the HWID issue. Your job and our job as users can be to do two precautions for such an issue: first, we need to understand how is this sent. Second, we need to learn how to isolate what's within the isolated. Installing an inner VM can help the data be harder for it to be seen, if someone from Kodachi VPN will see it, it will still be distorted. Sometimes encryption can be in principle and not just cryptography.

Now, let's go by flight of imagination to Virtual Box. What if we install Puppy OS inside a VM that runs inside Kodachi and its network.. Would Warith still know the activity ? If so, how the HWID will still be sent ?

The aforementioned hypothesis can conclude that the HWID reporting is actually a part of Kodachi VPN itself, not the Kodachi OS. And for such a case you would be concerned regardless of the HWID being sent or not, because you are using the VPN network in its totality. This brings us to the question of, can VPN know what you are doing ? We did answer this in the beginning of this book. So, for the sake of repeating my self I will give another explanation. Http is data that is sent, while html is data that is viewed. Http, has SSl encryption that works together with an authority, where the vpn itself can't see. Nevertheless html can't be hidden from the VPN. From this fact alone we can assume that a worst case scenario is Warith knowing what you visit with Kodachi, but never what you send. We can look into the *.kbase* directory, and we won't find anything spooky. The final answer is, change the VPN, if the HWID is that much of a problem for you.

I don't find that so striking to have HWID being sent, every system can be a virtual enemy of the consumer. We just have to learn how to protect ourselves from the protector, if we decide not trust them.

- **Tails**

Comparing Tails to Kodachi is like comparing a Mini Cooper to an Airbus 380. Not only Kodachi is highly secured and capable, but I don't even think that Tails claim that they are better than the latter. Tails is simply a Debian that only connects through Tor, nothing less, nothing more. Tails is for Tor browsing, a shallow increase of privacy, it is never meant to be a one and only tool for everything out there. Tor is only a small piece of Kodachi's world, and to put Tails on the other side of the seesaw, is demeaning to Kodachi.

We are concerned here with how close to reality can we get ? For example: are you able with Tails, to run an isolated Crypto Wallet and tunnel it via Tor and VPN ? Are you able to remove every browser trace such as user-agents ? Are you able to restrict all the JavaScript trackers ? or the location givers ? Can you implement a configurable proxy ? Can you change the screen resolution to distract eyes ? Can you create fake traffic ? Can you reduce keyboard frequency to stop key loggers ?

Of course the comparison is only valid for the internet (Network), so we shall only challenge

Tails with those. Still an easy challenge, because does Tails have DNS encryption ? DNS Leak tests ? or TorCrypt ? or perhaps options that close down every traffic based on told conditions ? Tails can't change a timezone, can't spoof neither MAC and neither HWID. But yet we have to hear all these unpleasant reproaches regarding this comparison, and here we are reflecting the questions back.

I shall stop here for now. Maybe I'll come back for [illegible] amends later on.

- **Whoonix**

Our inquiry about Whoonix, won't teach us something that is not of common knowledge. Because if we perceive it as to why Kodachi is better ? we'll be given the golden question. **What is our security goal to begin with ?**

Our personal activity will always remain as a transient structure. It's hard to make it perfect. But, if we can't secure all then at least we can secure some. And because we have to secure different situations, we're gonna need different tools. And, Whonix is just another Tails. With that being said, Tails is just Tor, but Kodachi, and according to The Cyber Gizmo ***"Is a wide system"***. Hence allow me to narrate that Kodachi, covers more practical situations, that necessitate more tools.

Installation

There are two primary methods you can use Kodachi. One is to Install it on your computer, while the other is to boot it live from a USB. The latter means, you wouldn't have to go through the long process of installing the system. Each method has pros and cons. The pros and cons are around security and performance. Since Kodachi is a security OS. When it comes to forensic usage for the different daily implementations, it depends on running being run as *Live*. If one cares mostly about the network communication and internet security, then Installing Kodachi on the computer is always better.

In most of the cases, you would have to burn the Kodachi OS into a USB storage device. That is done by copying the ISO files which digi77 gives. The copying isn't as simple copy and paste, but shouldn't be a hard task either. There a ton of tools out there that are designed to that for you, and thanks to Open Source and FOSS, 95% of them are free. The most common ones, that are also recommended here are :

- **Unetbootin**
- **Ventoy**
- **Rufus**
- **gnome-disk-utility**

- Ubuntu's Start Up Disk Creator

These on the list serve one purpose. Nevertheless they serve it differently. Unetbootin can help you burn Kodachi on a specific partition, when then allows you to use the USB device for other purposes at the same time, for cases of USB that are a large HDD/SDD. HDD/SDD usually carry a lot of data and needed for different reasons.

Ventoy's main purpose is to let you run Kodachi anytime you want, but at the same time, run other Operating Systems as images as well. In this sense, Ventoy helps you Install any OS without having to download and burn that OS's ISO every time, this can serve well for service shops who fix and install different OS's for different clients.

Rufus is the old Windows burning tool, it is very plain and simple one, but it has issues which are unfixable without having to run these other tools. Therefore you can try Rufus, but keep in mind that it can let you down at times.

gnome-disk-utility isn't even designed for burning ISO's and OS's, but it does. Because you can use the system restore feature, which does work handy for cases where nothing else works. The downside is that you would burn the OS into the entire disk, which will delete everything else on the device. It does fix scenarios where Kodachi doesn't boot, but data will be gone.

Ubuntu's startup disk is exactly like gnome-disk-utility.

Note : Ventoy also deletes all the disk, but at least you can store files and folders after that, that are regardless to Operating Systems. So at least it gives you that freedom.

A lot of people lost their mind when they can't burn Kodachi, let's admit that. But that's never a problem with Kodachi, it is lack of knowledge. In one sentence; Kodachi installation can goes so smooth for the majority, but it can be tedious for some. In any case, the support group in Discord is designated for helping those that run into Problems.

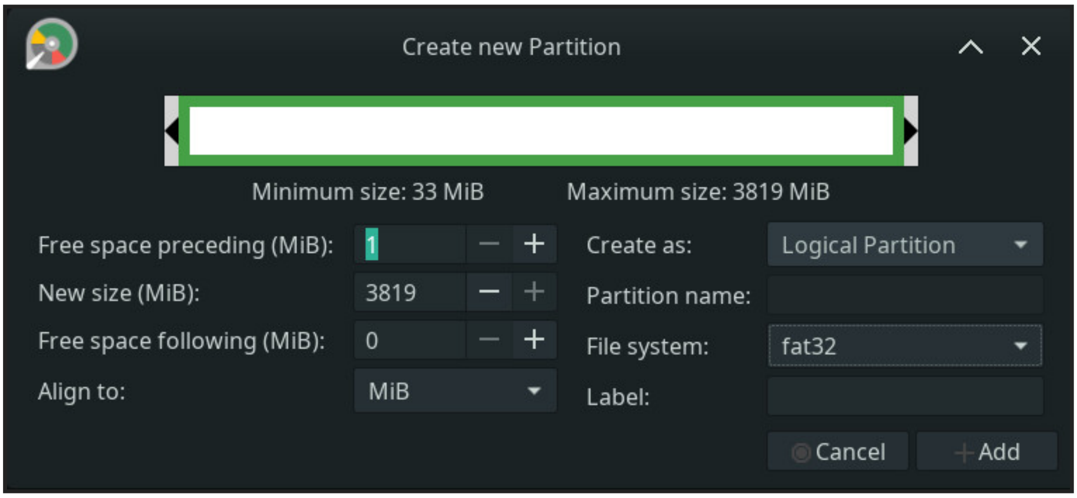
- **How to prepare the USB and actually do it ?**

1. You need to download the Kodachi iso from the official website, that is **https://digi77.com**, after choosing the Kodachi section, it will take you to the Source Forge platform which has the actual file.
2. You need download one of these tools mentioned above. For our case here, we'll use unetbootin,

since it's the most widely used tool.

3. After we download both these tools, we need to set them aside and focus on preparing our usb device. We would have to go through these sub-steps:

    1. Use **Gparted** to create a partition. You can enter the **Create new Partition** option from the main menu in Gparted, at the top. The partition which you need to create for the USB, should always be FAT32. Remember as a golden rule; **all bootable OS USB's should be in FAT32 file-system format**. The size of the partition can vary from one OS to another, but with Kodachi, 6GB should be enough.



Create new Partition

Minimum size: 33 MiB   Maximum size: 3819 MiB

Free space preceding (MiB): 1

New size (MiB): 3819

Free space following (MiB): 0

Align to: MiB

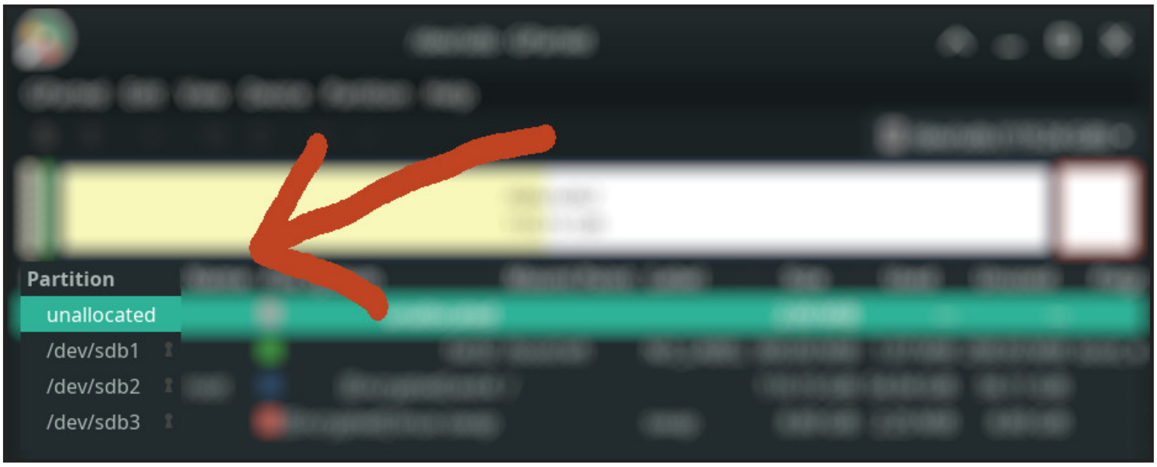
Create as: Logical Partition

Partition name:

File system: fat32

Label:

    1. After we are done partitioning our new bootable partition. We now need to first mount it, and second: burn Kodachi onto it. Therefore we need to know where does our system see this partition, or in other words.. what directory it is being assigned to as.

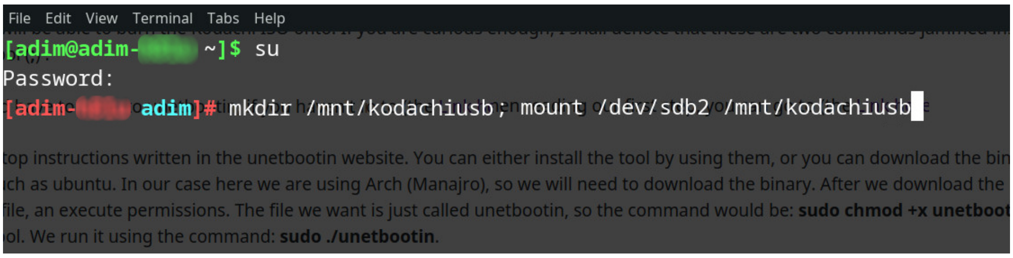


As you see in the picture above, the partitions of our HDD is looked at as /dev/sdb1, /dev/sdb2 and /dev/sdb3

Let's assume that our FAT32 partition that we just made, was designated as **/dev/sdb2**.

3. What we need to do is mount that partition. There are two ways we can do this. We can either mount using disk tools that exist on Linux and Mac OS, or we use the terminal. I believe one need not to depend on nothing but a terminal for such things, therefore ..

4. As you should see in the image below, we first give ourselves root access by running : **su** and then entering the root password.



```
[adim@adim-[redacted] ~]$ su
Password:
[adim-[redacted] adim]# mkdir /mnt/kodachiusb ; mount /dev/sdb2 /mnt/kodachiusb
```

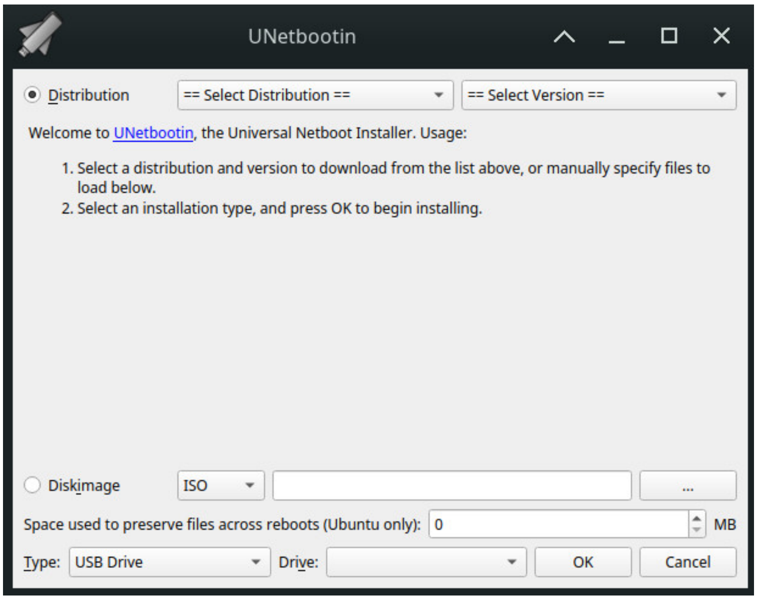
top instructions written in the unetbootin website. You can either install the tool by using them, or you can download the bin
ich as ubuntu. In our case here we are using Arch (Manajro), so we will need to download the binary. After we download the
file, an execute permissions. The file we want is just called unetbootin, so the command would be: **sudo chmod +x unetboot**
ol. We run it using the command: **sudo ./unetbootin**.

5. Then we run the following : **mkdir /mnt/kodachiusb ; mount /dev/sdb2 /mnt/kodachiusb.** The latter is one command, don't

be scared by its length, what it does is first make a directory folder inside of the mount system, called kodachiusb, then it mounts the partition we talked about, inside of that new directory. Now unetbootin, that is our preferred tool of use, will be able to burn the Kodachi ISO onto. If you are curious enough, I shall denote that these are two commands jammed inside one. We can merge commands by adding the symbol (;) .

2. Now would have to jump to unetbootin, if you haven't installed it when reading our first step, you can go to (the **Link here**).

3. There are top instructions written in the unetbootin website. You can either install the tool by using them, or you can download the binary. These instruction or designed for Linux distro that use apt, such as Ubuntu. In our case here we are using Arch (Manajro), so we will need to download the binary. After we download the binary, we need to enter that directory and give the main tool file, an execute permissions. The file we want is just called unetbootin, so the command would be: **sudo chmod +x unetbootin**
After the execute permissions are given, we need to run that tool. We run it using the command: **sudo ./unetbootin*.bin**



● Distribution

Welcome to UNetbootin, the Universal Netboot Installer. Usage:

1. Select a distribution and version to download from the list above, or manually specify files to load below.
2. Select an installation type, and press OK to begin installing.

○ Diskimage ISO

Space used to preserve files across reboots (Ubuntu only): 0 MB

Type: USB Drive

Drive:

1. As you can see in the unetbootin tool, there are [illegible] options to burn Linux, we should disregard all except one, that is the below one. From below we can choose the ISO file of Kodachi, and then choose the option USB. After such, we choose the partition we have, that is **/dev/sdb2**. If you don't that partition, click the disk option and go back to usb, it should show now, it's just a small bug in the tool.

2. Click **Ok**.

3. Now we have our Kodachi burned and ready inside out USB device.

- **How do we boot into Kodachi ?**

Most computers run a system before the operating system. that first system is called either BIOS or UEFI, or both. It talks directly to the hardware.

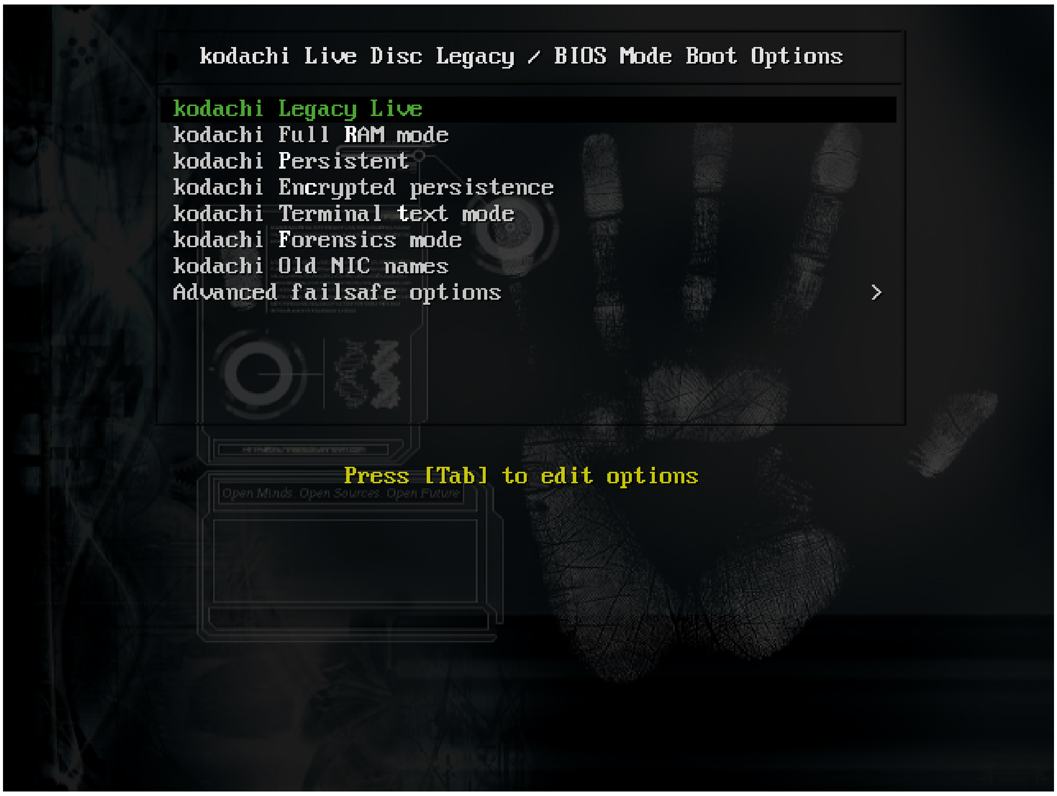
So we make sure our USB first is connected to the computer which we want to run Kodachi on, then we press the power button.

After pressing the power button, we would need to be a little fast. We don't want to wait until the main OS starts, instead, we need to enter the boot menu. Boot menu is usually accessed by pressing either the **F2**, **F9** or **F12** button. If you don't know which button that is, just pay attention to the black screen that runs when your computer is started. If you still don't see it even from the black screen, that means you have fast boot or similar option activated. To fix the last problem, we need to enter the BIOS/UEFI settings, that is done either by **F2**, **Delete**, or **F4**. In that last case, we would have to go to the menu that says Settings, and disable Fast Boot, or any option that stops hints or text to be displayed from the first system menu.

1. Once you figure out the boot menu, you will see any disks on your computer, from them you should see you USB device. You might as well see two of your USB device, chose one of them until you access the Kodachi menu.

2. Boom! Now we are in Kodachi world, we can now either access the Live version by choosing the

first option, or choose **Advanced failsafe options** from which we can install Kdoachi directly.



```
kodachi Live Disc Legacy / BIOS Mode Boot Options

kodachi Legacy Live
kodachi Full RAM mode
kodachi Persistent
kodachi Encrypted persistence
kodachi Terminal text mode
kodachi Forensics mode
kodachi Old NIC names
Advanced failsafe options                          >

Press [Tab] to edit options
```

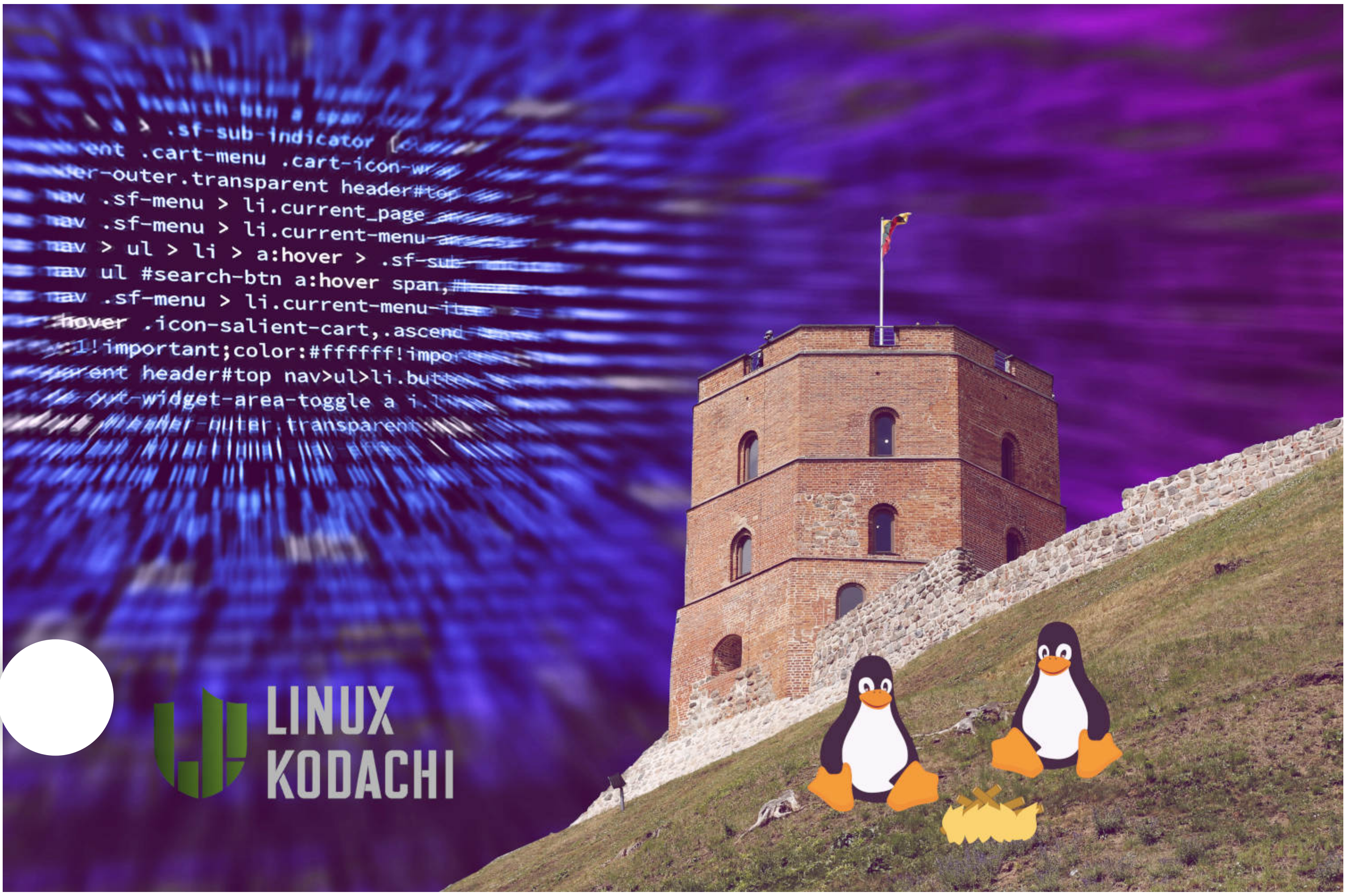
- **Fixing the (No network connection) problem**

1. Choose the repair network option from the Panic Room in the Dashboard.
2. Restart the Network Manager, by running: `sudo systemctl restart NetworkManager`
3. Spoof MAC from the Panic Room in the Dashboard.
4. If nothig happened.. make sure all services are running fine, by looking into: `sudo systemctl enable --now network-manager`
5. If any service is not running, please report it the Kodachi Support Group

oken write access to the ntfs partition.

1. Mount the *ntfs* file-system, using **disks**, or **mount**.
2. Check where this volume is mounted.
3. Then run this : `sudo chown root:root -R /<path to the volume>`
4. If that didn't work, then try `sudo chown kodachi:kodachi -R /<path to the volume>`

What's next ?


.sf-sub-indicator
.cart-menu .cart-icon-wr
-outer.transparent header#top
.sf-menu > li.current_page
.sf-menu > li.current-menu-
> ul > li > a:hover > .sf-sub
ul #search-btn a:hover span,
.sf-menu > li.current-menu-
hover .icon-salient-cart,.ascend
!important;color:#ffffff!impo
header#top nav>ul>li.but
widget-area-toggle a i
LINUX
KODACHI

- Kodachi on Debian 11 and Light version Kodachi (Kilodachi)
- More edits coming soon !
  - It's never late to join the Terminal.
    - *Commands that are favorite to Kodachi (chmod, chown, veracrypt, ls -la, mount, systemctl, ifconfig, nmap).*
  - Luks (un-saved, one time encryption password).
  - Using Audacity to change your voice.
  - Cutting the line between Kodachi and other OS's, to stop them from merging with each other.
  - Securing your USB, HDD and mobile storage from being restored (Overwriting, using the gnome-disk-utility).
  - VPN IP is the same as the ISP IP.

- Tor vs VPN.
- Sphere vs Kodachi Lite Browser.
- Briar (Tor to Tor messaging).
- Wall Street Buddies (Signal and Telegram).
- Normal Procedures.
- Suspicion.
- Affiliations.
- Emergency Procedures (Panic Room).
- Communicating.
- Network-Manager.
- Impersonating & Falsifying.
- Personnel packs.
- Hiding Tor’s exit node.
- VPN over Tor, instead of Tor over VPN.
- Why two VPN's are dangerous together ?.

Given Under Creative Commons License

Attribution 3.0 Unported (CC BY 3.0)

https://creativecommons.org

You are free to:

- Share — copy and redistribute the material in any medium or format.

Next >

Main Menu

Tutorials

Software Analysis Reviews

Analytic Gallery

Informational Resources

Donate now with..
PayPal™
VISA
MasterCard
AMERICAN EXPRESS

bc1q7cznuect9ny396ufsdxdzhwhvcf4g59ujfgvnn

Kodachi Solutions

Forum

- Index
- Recent Topics
- Search

Recent podcasts.

Sorry, that's not currently available.

Luckily, lots of other stuff is.

Sorry, that's not currently available.

Luckily, lots of other stuff is.

Login Form

Username

Password

□ Remember Me

LOGIN WITH FACEBOOK

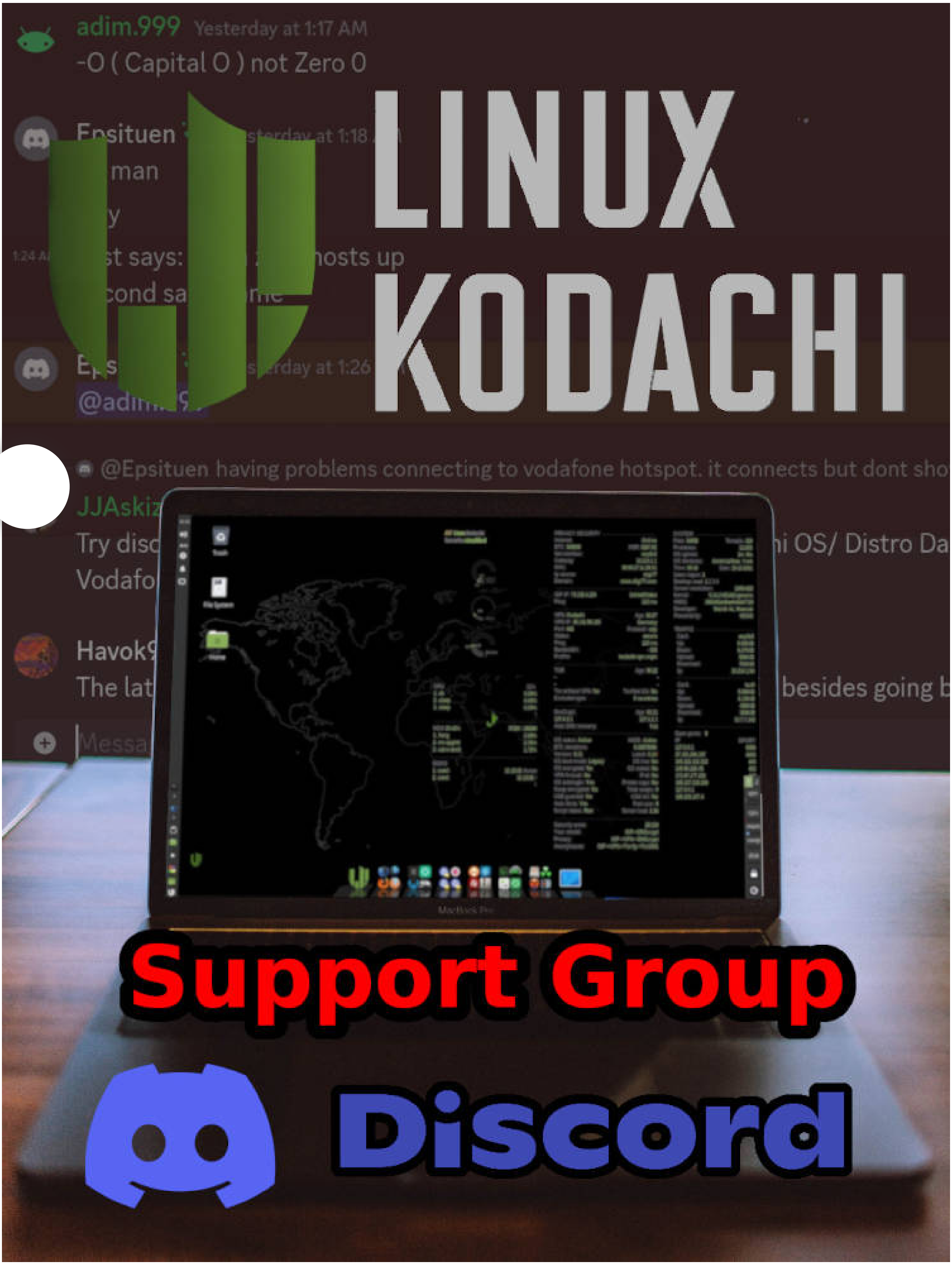
LOGIN WITH GITHUB

SIGN IN WITH GOOGLE

WEB AUTHENTICATION

LOG IN

Forgot your password?
Forgot your username?
Create an account



adim.999 Yesterday at 1:17 AM

-O (Capital O) not Zero 0

Epsituen Yesterday at 1:18 A

man

y

1:24 A st says: hosts up

cond sa me

Ep s sterday at 1:26

@adim 9

@Epsituen having problems connecting to vodafone hotspot. it connects but dont sho

JJAskiz

Try disc hi OS/ Distro Da

Vodafo

Havok9

The lat besides going b

Messa

Credits

Powered by

Operating Systems

Contact us

About us
Support Control Room
Data Policy
echnical Team