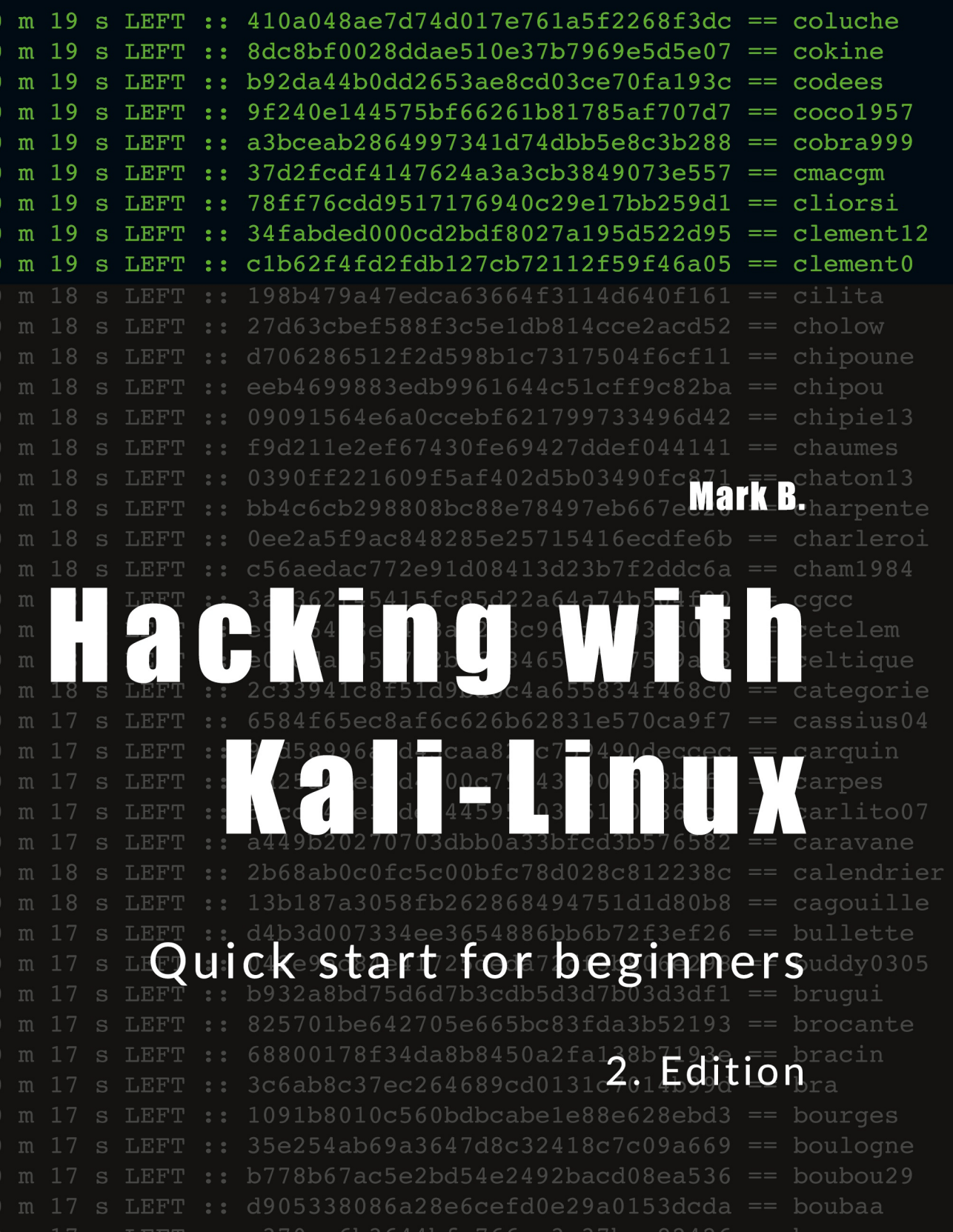


Mark B.

Hacking with Kali-Linux

Quick start for beginners

2. Edition

Acknowledgments and foreword

First of all, I would like to take this opportunity to thank everyone who supported and motivated me during the preparation of this book.

I would especially like to thank my girlfriend, who motivated me throughout the work and didn't take it amiss that I invested so much of our free time together in this project - thank you, honey!

What awaits you in this book is a rough introduction to Linux and the installation of the Linux distribution Kali. Then we will deal with the configuration a little before we will work with various tools.

You will get an insight into the most important sub-areas and learn how to work with ready-made tools, how to approach problems, how to uncover weak points within a system, and in some places, you will even learn how to write small tools yourself.

A word of warning

At this point I want to say very clearly - **anyone who uses what I have shown here against third-party websites, networks, or computers without the consent of the owner is liable to prosecution!** However, whoever uses the tools to test his own IT landscape will be able to increase security enormously by identifying possible attack points to fix them.

Anyone who attacks their own websites should also ask the hosting provider for permission in advance so that the administrators know about and do not immediately send abuse reports to your Internet provider. Besides, it is also advisable to inform your own provider so that he does not block your Internet connection as a precaution as soon as he notices what you are doing.

This book is not intended as a guide to committing criminal offenses, nor as a guide on how to avoid criminal prosecution!

MARK B.

Hacking with Kali-Linux

Quick start for beginners

2. edition

Imprint

Bibliographic information from the German National Library:
The German National Library lists this publication in the German National Bibliography; detailed bibliographic data are available on the Internet at http://dnb.d-nb.de.

Production and publishing:
BoD – Books on Demand, Norderstedt

ISBN:
9783753486642

Inhalt

Why I wrote this book

In my work, I keep coming across networks, websites, etc. with significant security problems. My aim is not to show how to hack other websites, networks, or computers, but to convey to the reader how easy it is to achieve this with various tools. Therefore, in my opinion, anyone who operates a network or a website should know to some extent how various hacker tools work in order to understand how to protect themselves against such threats. Even simple users with their home PCs are popular targets today. It would therefore be advisable for this group of people to familiarize themselves with the topic of IT security.

Given the subject is a very technical one, I will still try to explain the concepts as generally understandable as possible. A degree in computer science is by no means necessary to follow this book. Nevertheless, I don't just want to explain the operation of various tools, but also roughly outline how they work. At least to the point where it becomes clear to you how the tool works and why a certain measure helps against it.

I've noticed a trend for a long time - more and more tutorials and questions about Kali Linux (*formerly Backtrack*) are appearing on the Internet. It seems that hacking is slowly but surely becoming a "national sport". What bothers me about most of the tutorials is that they show how an attack works in a certain situation, but they almost never go into what exactly happens and how the tool works in detail. However, this is exactly the key to understanding the security problem and how to fix it.

These so-called "script-kiddies", who have various tools and can use them in certain situations, do not understand the context behind them and are then usually hopelessly overwhelmed when they have to deviate from the known scheme.

Feedback & criticism

If you want to get rid of criticism, suggestions, or even just praise, please send me an email to `mark.b@post.cz`.

I will try to implement your input in further book projects or in upcoming new editions.

Hackers, crackers, script-kiddies

Since there is no general definition and the terms also have a smooth transition, I will give you my personal definition:

A hacker can be defined as someone who deals with the security of computer systems and looks for weak points in the systems. This can have various reasons, from pastime to thirst for knowledge. If a hacker finds such a vulnerability, he will publish it to make the world aware of the bug. What a hacker will not do, however, is to exploit this vulnerability for their own benefit in order to capitalize on it. This is why these hackers are also known as whitehats.

Crackers are in contrast those hackers who don't follow this moral code and break into systems to cause damage, spy on secrets and then sell them, cripple computer systems and extort money, and so on. These people are usually motivated to capitalize on their skills and earn as much money as possible in the shortest possible time. Many of these crackers are now part of larger organizations and are responsible for billions of Euros damage per year worldwide. This group is also known as blackhats.

Whitehats as well as blackhats are tech-savvy and able to find vulnerabilities in software and develop tools that exploit these vulnerabilities.

Script kiddies don't have these skills. In the best-case scenario, they have knowledge of how to use hacking tools. Often their knowledge is limited to a fraction of the functions of various tools. Furthermore, script-kiddies usually do not really know how exactly the tools they are working with functions and what the technical background of their preferred tools is. Therefore, many of the script-kiddies do not know how to help themselves if the standard procedure does not work. But that doesn't make them any less dangerous. This group includes a good 90-95% of people who carry out attacks on an IT system and this group includes everything - from 14-year-olds who just want to try out what they have found on the Internet to full-time cybercriminals, who wants your account and credit card details.

In the following part of the book, I will use the term "hacker" as an umbrella term for all of the subspecies mentioned here, as most people are used to from common language. At this point, I leave it to you, the reader, to differentiate which type of "hacker" is meant in a particular case.

What is Linux?

After a long back and forth, I decided to start our journey together at the very beginning to give you a smooth start, even without any prior knowledge. Anyone who already has experience with Linux can safely skip this chapter. However, I recommend that those who have Linux experience at least skim the chapter on installing and configuring Kali.

Linux is an operating system such as Windows or Mac OS. Like any operating system, a Linux installation contains a whole bunch of tools. These tools would be a browser, a calculator, an editor, or a player for music and videos, etc. With Windows and Mac OS, this software combination is standardized - depending on the version, the combination of tools can change, but the same tools are always included in every Windows 7 Home Premium. That is perfectly logical since Windows is created by only one company. The same is valid for Mac OS.

Linux is free software. This means that anyone can download the core of Linux and make their distribution out of it. A distribution is a compilation of software. There are currently hundreds of Linux distributions made available by as many different providers. Everything is included - from company-owned distributions created for personal use to hobby projects by enthusiasts to professional distributions that also offer paid support for their product.

The distributions can also be classified according to their area of application. Some distributions are designed to run as a firewall, others should provide a stable working environment with long-term support, others provide the latest programs and are therefore interesting for developers to test their software, but do not run as stable, etc. Kali-Linux, which is what this book is supposed to be about, is a distribution that comes with an enormous collection of tools for security testing, data forensics, etc.

Kali-Linux is like a system that comes with everything you need to break into computer systems. This is ideal for testing your security, but also a gift for every script kiddie who has a perfect system for hacking.

What is the advantage of Linux?

The main difference is that Linux is open source. This means that everyone can see the source code that makes up the Linux kernel. This source code is a collection of instructions that are then translated into an executable program. Since anyone interested can see how Linux is programmed, security gaps are quickly found, made public, and then closed again. Linux also follows the "*everything is a file*" principle. For example, program configurations are managed legibly in text files and are usually separated for each program. This allows program settings to be easily saved or transferred from one computer to another - copying one or a few text files is sufficient.

Windows is a prime example of closed source - a total black box. Various programs store their settings in a central system-wide registry, in which Windows itself also stores many configuration settings. Besides, these settings are usually stored incomprehensibly and legibly to once again disguise the inner functionality of the individual programs. Of course, the Windows program code is Microsoft's strictest trade secret. But let's compare for yourself how Windows and Linux save program settings:

Using the example of how to switch off SSL 3.0 in IIS:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\Schannel\Protocols\SSL 3.0\Client REG_DWORD 0x00000001
```

This entry can only be found if you open the registry with a special editor. It is hidden with thousands of others in an almost infinite folder and sub-folder structure. Besides, various settings of Windows, system services and user software are mixed in the same registry, which causes two problems:

1) is not exactly clear and
2) various programs are allowed to access it and change things in the registry.

If a program simply changes the settings of a system service to open a back door for the developer, you have a simple but effective Trojan.

On Linux, there would be a file or folder in the `/etc` directory that contains the settings for this system service. It could look like this:

In the "`IIS.conf`" file you would find the line `Client_can_use_SSL3 = Off`. If the configuration were split into different files for client and server, a line with `Can_use_SSL3 = Off` would then be found in the "`IIS_client.conf`" file.

Admittedly, since the IIS is not available for Linux, this is a somewhat theoretical example, but you can see what I mean.

To make this a little clearer, here is an example from an Apache configuration file:

```
<Directory "/var/www/phpMyAdmin">
     order deny,allow
     deny from all
     allow from 127.0.0.1
</Directory>
```

The `<Directory "/var/www/phpMyAdmin">` indicates the folder for which the information applies. Access from anywhere is prohibited (*line 2*) and access from IP `127.0.0.1` is permitted (*line 3*). If you had to apply this configuration to one or more computers, you would only need to copy this file to the relevant computer or paste these lines into the files on these computers.

Since Linux is open source, Linux can also be downloaded, used, and even distributed completely legally and free of charge from the Internet.

With Linux, you have the choice of which window manager you want to use. The window manager is, so to speak, what defines the graphical user interface and comes with the general look and feel as well as the necessary programs for file management, etc. With Kali-Linux you can choose between KDE, Gnome3, Enlightenment, LXDE and XFCE.

The first two window managers are significantly more resource-hungry. Enlightenment, LXDE and XFCE get along well with very modest hardware. To explain the advantages and differences of the individual window managers at this point would be far too much for the scope of this chapter. Therefore, you are free to download the ISO images with the individual WM variants and test them yourself. Kali-Linux offers a so-called live image and can therefore be started and tested immediately from the DVD or USB stick without installation.

"Windows is unsafe!" You read this statement all too often on the Internet and it is partly true. If you take the configuration that you find on 90% of the computers that you buy at the friendly electronics discount around the corner, the statement is true. Home user systems are usually configured to work as an administrator. That is negligent. Every program that I start as an administrator also gets admin rights and if I start a trojan this way, it can operate as it wants on my system. In a company environment, Windows systems are usually configured in such a way that users only have those rights that they need for their work. Linux systems usually require such a configuration and, during installation, force the user to set up a 2nd user besides the administrator (*who is called* `root` *on Linux*) without such extensive privileges. Many systems even go a step further and do not allow you to log into the graphical environment as `root` - at least not before you change some of the configurations.

In this sense, Linux is more secure, but mainly because Linux forces the user to use a more secure configuration. What then remains, in the end, is the issue of viruses, worms, spyware, Trojans, etc. And Windows has more problems there, and for the following reasons:

Windows is very common. This will make it more efficient to write Trojans for Windows. One can assume that many systems are configured insecurely and the masses use Windows and that means he can expect the maximum number of victims.

Windows is "standardized". If I write a program that exploits a security hole in Windows Explorer, then I know 100% that Explorer is installed on every Windows. With Linux, it depends on the distribution and which window manager you are using. KDE users usually have Dolphin as their file manager, Gnome users have Nautilus, and XFCE users use a program called Thunar. A security hole in a program does not necessarily have to affect every Linux, but only those distributions that use this program. If the error affects the system kernel or core components of Linux, the number of potential victims is naturally greater. However, these limitations and the low prevalence make it much less effective to develop such programs for Linux. That doesn't mean, however, that there are no Linux Trojans!

"There are no viruses, spyware, etc. for Linux!" This statement is utter nonsense. There is indeed far less malware out there for Linux. It is also true that the existing malware can usually cause significantly less damage because it lacks the rights in most cases, but you are still not completely safe!

I have to admit that I hardly ever work with Windows systems anymore. But what I still remember today are the frequent system crashes and blue screens. But, I have to say that Linux crashes also occur. If you use the latest program versions, such as Fedora Linux, then you have to struggle with such problems. Those who rely on distributions such as CentOS or Debian, which are designed for stability, have to get along with a smaller selection of software in the repositories but can rest assured that these have been extensively tested and are stable.

I will deal with the installation of drivers and software in the chapter with the installation of the system.

This list of advantages and disadvantages naturally also reflects my personal opinion and in case of doubt, you should decide for yourself what you like better. At this point, I also like to admit that I am a Linux fanboy. However, I have become one through years of positive experience. When I think back to my switch from Windows, there were a few things that seemed unnecessarily complicated, cumbersome and confusing, until I discovered and learned to appreciate the advantages of the Linux approach. It is precisely for this reason that I have written this introduction for all those who are new to Linux or who have little experience with Linux.

Installation and quick start

First of all, of course, the question arises: "Where can I get Kali Linux from?"

Kali is published by Offensive Security and can be downloaded from the official homepage: `https://kali.org/downloads/`

You can choose from ISO files that you can burn on a DVD or flash on a USB stick. It is important not to burn the ISO image as a data DVD or simply copy it to a USB stick. You also have the choice of image files for ARM processors (*e.g. Raspberry Pi*), but I won't go into these here.

In the case of ISO files, there is also a choice between a 32-bit and a 64-bit variant. If you are using a reasonably up-to-date PC, take the 64-bit variant. If you are not sure whether your hardware can handle 64-bit, then my recommendation would be: Try the 64-bit variant and if it doesn't work, then take the 32-bit variant. I choose the 64-bit XFCE version, which I can warmly recommend to you. With Kali 2020.2 this window manager is the standard!

After you've downloaded the image, you'll need to burn it to a DVD. For this, you can use the program **ImgBurn** (`http://imgburn.com`) under Windows. Of course, you can also do this with many other burning programs. ImgBurn is free, specializes in burning image files, and therefore offers little margin for error. Nevertheless, I do not want to withhold quick instructions from you.

When you open the program, select "`Write Image to Disk`" in the overview that you receive after starting the program. In the following dialog, you will find an entry "`Source`" at the top left and next to it a button with an open symbol. Click the icon and select the ISO file that you downloaded. Uncheck "`Test`" and "`Verify`" at the bottom of the burn window and set the burn speed as low as possible. After that, you can click the burn button directly below the checkboxes.

Mac users can use Disk Utility. You can find it in the "`Utilities`" folder within the "`Applications`" folder. The third symbol at the top of the program is a yellow-black striped circle. If you click on this burning symbol, you will see the open dialog. Select the ISO file and click burn. After that, you'll get a confirmation telling you your Mac is ready to burn. Confirm this again by clicking on "burn".

Linux users can easily burn the ISO file from the console. To do this, use this command:

```
wodim -v -dao --eject speed=4 /path/to/file.iso
```

If you do not have a DVD burner available or your Kali Linux PC, like my subnotebook, does not have a DVD drive, you can have the ISO file extracted onto a USB stick. Use **Unetbootin** (`https://unetbootin.github.io`) for this. To do this, select the ISO file by marking the "Disk image" item, select ISO in the drop-down field and click on the ". . ." button to open the ISO file. Then select the USB stick in the drop-down box directly below and click OK. he USB stick must have at least 4GB.

Linux users can solve this with a simple console command:

```
dd if=/path/to/file.iso of=/dev/sd[X] bs=512k
```

The `[X]` must be replaced by the appropriate drive letter for the USB stick. If you are not sure what you are doing, stay away from `dd`! This command is relentless and can render your entire hard drive with all of your data and the operating system unusable. With the appropriate software, some data can certainly still be saved afterward, but that's not funny.

To check what drive letter your USB-stick has you can use the command

```
df -h
```

which shows you the free space of all drives.

If you're using `dd` and you're sure which drive you're overwriting, don't be impatient. `dd` takes its time and does not report any progress!

Now that we have a medium to boot the computer with Kali, we can concentrate on the installation. I mentioned earlier that Kali is a live DVD and the system can also run from the DVD or the USB stick. Why I advise against it is simple: Cheap netbooks or notebooks don't cost much today. I bought my Kali netbook for 189 EUR - 2GB RAM, dual-core Atom processor and 500GB hard disk space. As far as hardware is concerned, Linux is very frugal. Why I decided against SSD storage is also easy to explain - cost and storage space. Netbooks often have very small 32GB or 64GB SSD disks, but password lists and dumps from sniffers can quickly become very large and I need the storage space. Sufficiently sized SSD disks cost me far too much for this purpose. The reason I chose a netbook is simply because of its size and weight. Angry tongues would even say: "*And if necessary it fits in the microwave*" ;-)

That is why it would never be an option for me to run the system from a USB stick or even from a DVD, which apart from that is significantly slower than anything else.

Nor would a dual boot with another operating system on my main computer be an option for me. Contrary to what you can see in the usual tutorials, it takes more than 3-5 minutes to crack a password. Word lists for tutorials containing only a few hundred passwords are used for demonstration purposes. In real life, such a dictionary attack can quickly take a few days to a few weeks. Therefore, for me, only an own dedicated computer for Kali makes sense.

Installation of Kali-Linux

After we have booted Kali from the installation medium for the first time, you will see the following screen:

Here I would recommend that you boot the system with the top option before installing to test whether Kali can be started without errors. You confirm the option by pressing the `Enter` or `Return` key. If the computer does not boot from the DVD or USB stick at all, you have to press a key when starting the computer to call up the boot menu. You can find out exactly which one in the manual for your mainboard. Alternatively, you can change the boot sequence in the BIOS so that the computer will always try to boot from the DVD or a USB stick first.

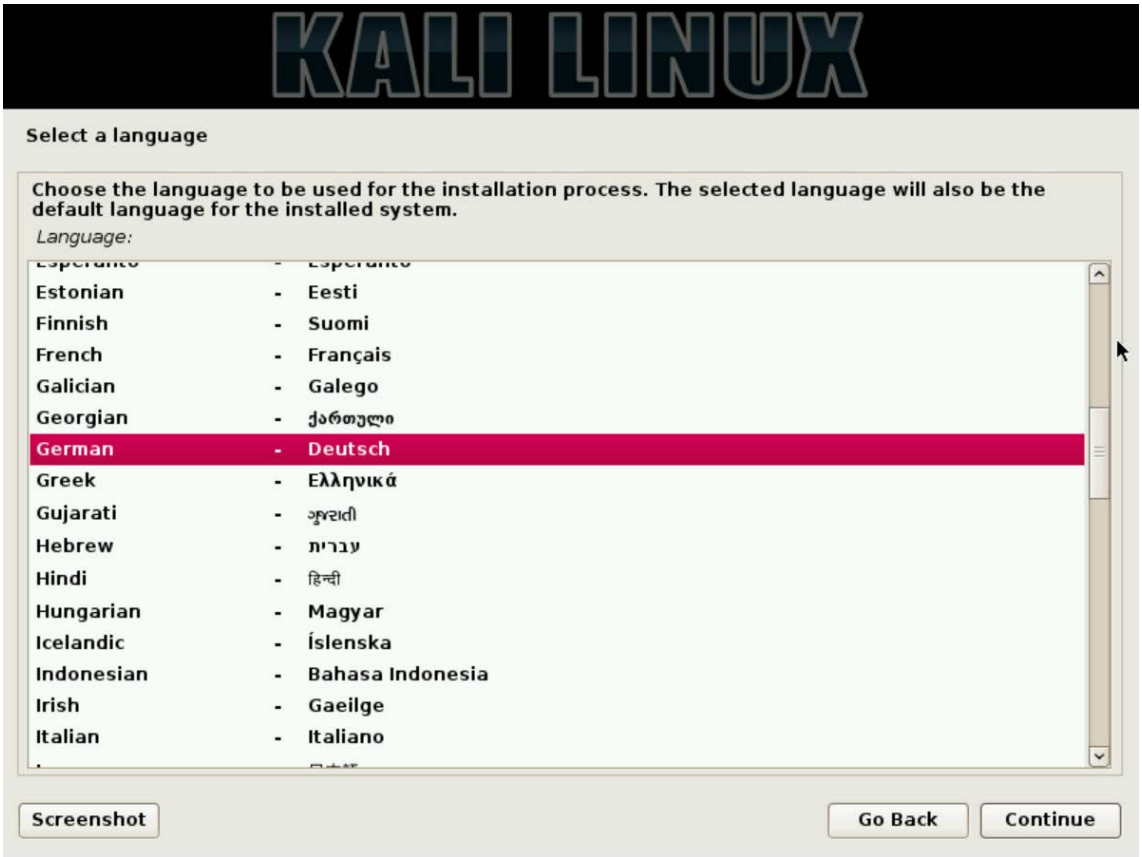
If your PC cannot boot Kali properly, you should check the following

1. Calculate the MD5 sum of the ISO file and check whether it matches the information on the Kali website. If not, the ISO file was corrupted while downloading and you will need to download it again and create a new boot disk.

2. Check your BIOS settings. Sometimes UEFI boot mode has problems. Change the boot mode to "Legacy". The operating instructions for the mainboard will tell you exactly how to call up the BIOS setup. Normally you have to press a certain key when starting the computer to enter the BIOS settings.

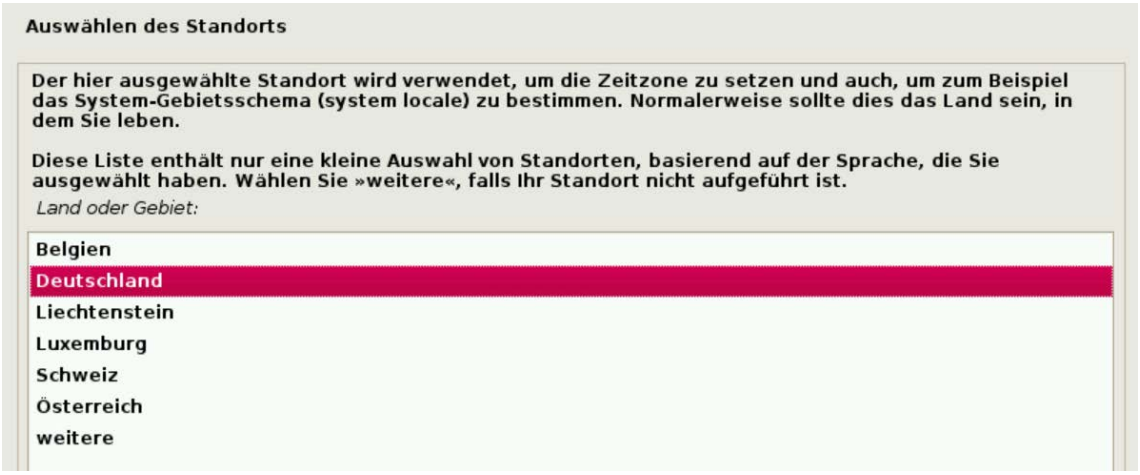
3. If neither of these works, the drive may have a problem reading the DVD. In this case, it is best to try the USB stick variant. If you tried to boot from a flash drive try another one.

Now that Kali has booted without errors and you have tested things like WLAN, etc., you can restart the computer and start the installation. To do this, wait until the Kali boot menu appears again and use the arrow keys to navigate to the line "Graphical install" and confirm it with Enter. Good knowledge of English would not hurt in this context, as many documentation and websites on the subject of hacking and Linux are in English. Nevertheless, Kali is of course also available in other languages like German for example:

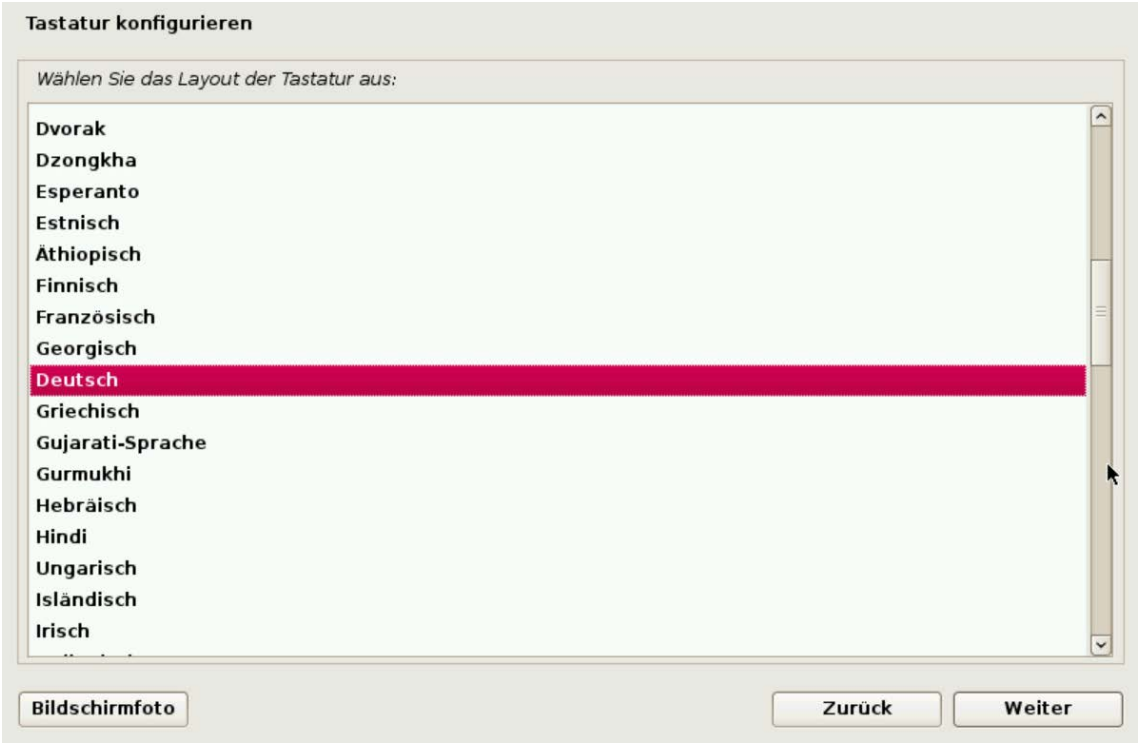


Select your desired language and click "`continue`".

If you change the language confirm in that step the change.



Then select the country and click on `continue`.

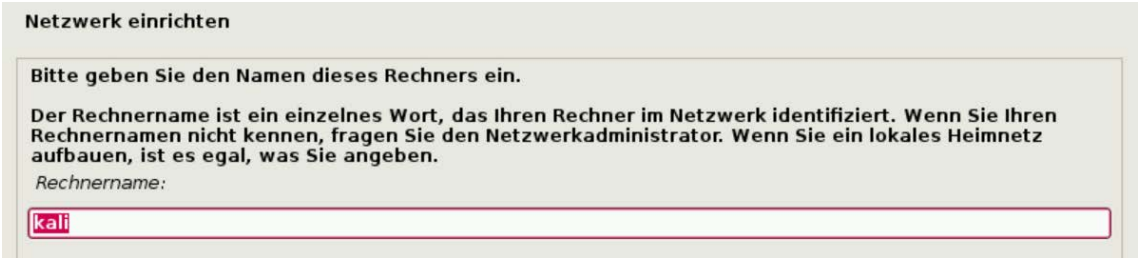


Select the keyboard layout and click on `continue`.

Then Kali-Linux tries to identify your hardware. And load the appropriate drivers. This step can take a few seconds.

An Internet connection is also searched for and an attempt is made to configure it. I would therefore recommend that you carry out the installation while the PC is connected to your router with a network cable.

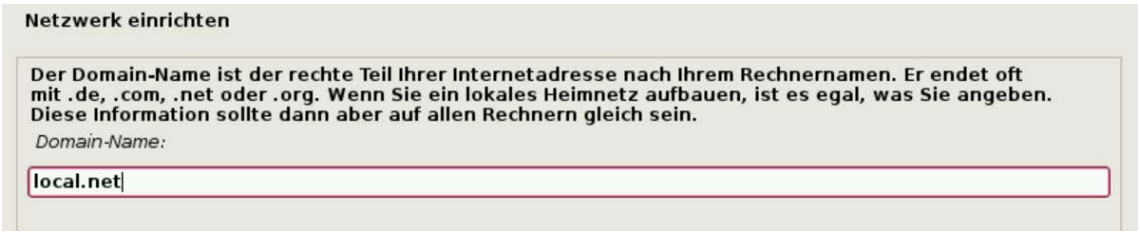
This configuration is easiest for Kali to recognize and Kali will automatically request an IP from the DHCP server on your router and then connect to the Internet.



In the next step, you can assign a name for the computer. I would say that computer names like "`Kali`" or even worse "`MyHackingPC`" or the like will raise alarm bells for every administrator if they appear on the lease list of a DHCP server or something similar. With "`MyHackingPC`" or the like, even the most inexperienced user will be alarmed if such a PC is displayed in the network environment. That's why I usually use a meaningless name here because I don't want to attract attention just by the computer name when testing a network.

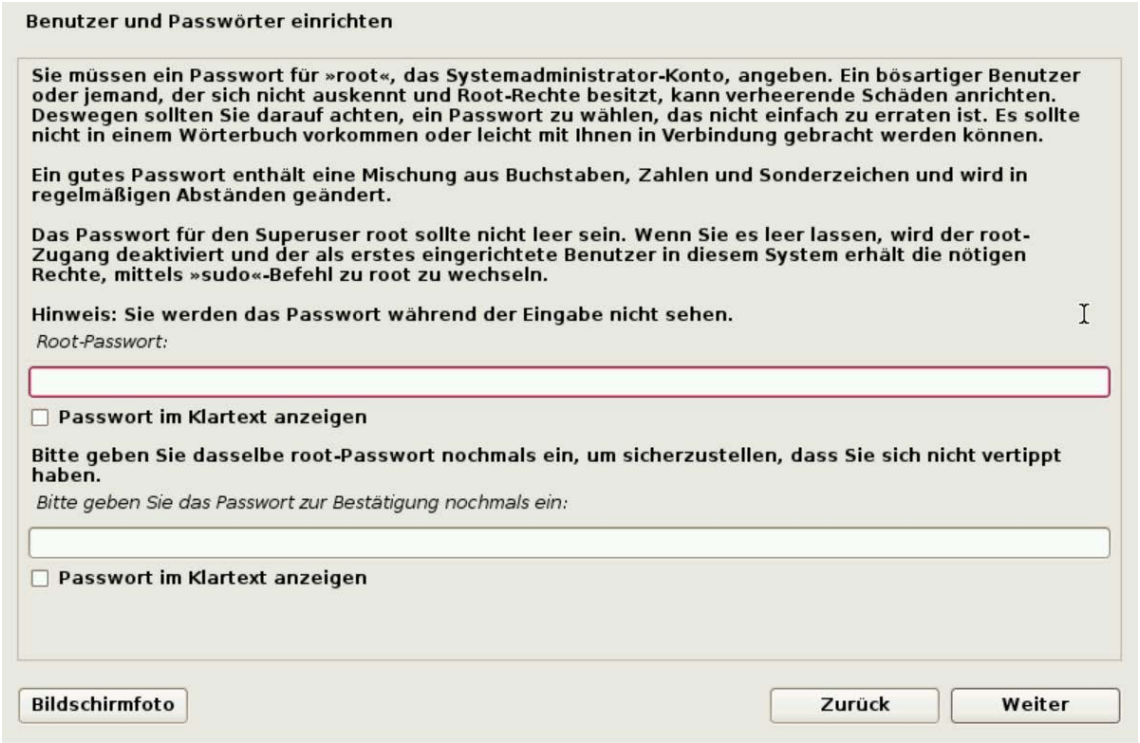
Again at the point - if I test networks then at the customer's request. Everything else is illegal!

For our book, I leave it with the suggested name "`kali`" and click on `continue`.



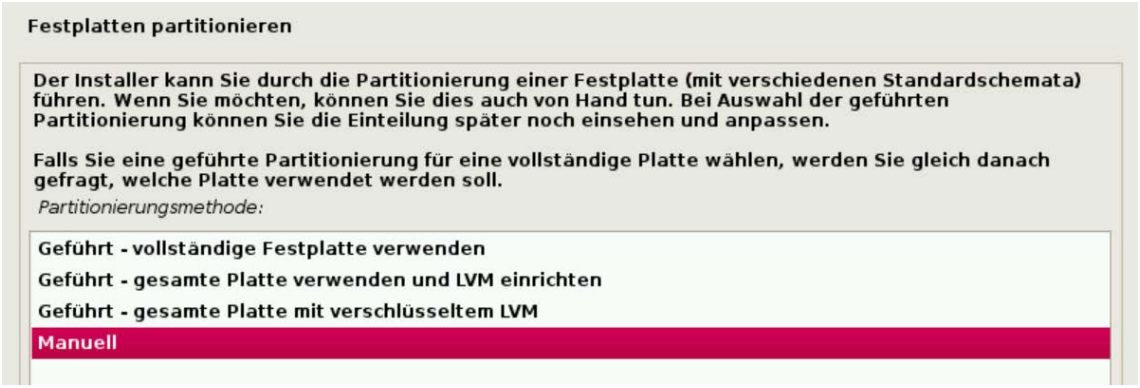
You can choose any domain name. I usually use "`local.net`" here. To accept the entry, we click again on `continue`.

It's not rocket science so far - is it?



Now we need to set a `root` password. The user `root` is the administrator in Linux and his commands are followed without hesitation and sometimes even without a security question. Therefore, you should first be careful what you do as root and secondly, also assign a reasonable password.

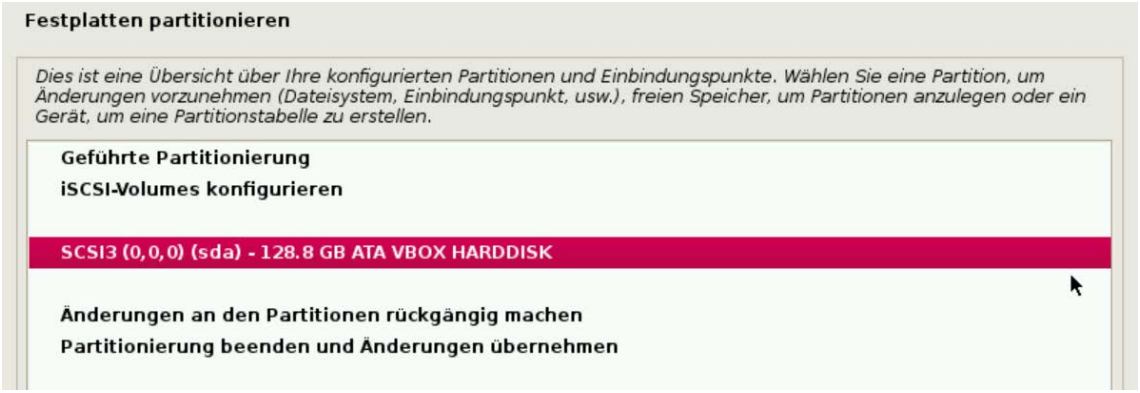
A secure password is at least 10 characters, preferably 12 to 16 characters long, and cannot be found in any dictionary. So your first name or the still very popular password, Password1, 123456, and the like fall out. It is best to use upper and lower case letters with special characters and numbers together. Unless you want to make your computer available to other people who can handle Kali and the tools it contains a little better..



So far everything was child's play - the next step will be a bit more challenging. It's about the partition layout. To put it simply, partitions are subdivisions of a hard disk. This allows you to virtually divide a disk into several hard disks and has the advantage that, for example, the system and data can be separated. If you were to format the system partition to set up the system again, the data would still be contained on the data partition.

I use a more complex partition layout. One of the reasons is that I can reinstall the system without losing my data, the other reason is that if the system or root partition is full, Linux can no longer boot properly. Therefore I give directories that could cause something like this a separate partition.

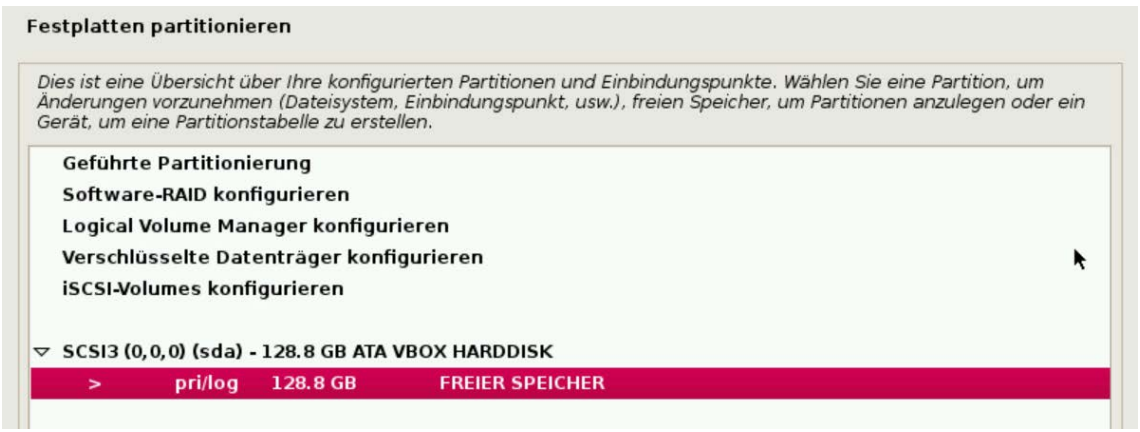
But let's start with the system partition and I'll explain everything else to you. First select "`Manual`" as shown above, then click `continue`.



Then select the internal disk of the PC and not the flash drive (*in case that is your installation medium*). That would work too but I told you why I don't recommend that option.

If your disk has never been used, you will be asked whether you want to create a partition table on the disk. Confirm this with `yes` and click on `continue`. If your hard disk has already been used, you will not see this dialog. To clean it, you have to delete the existing partitions in the next step. Logically all data that were on these partitions will be lost!

Step 1:

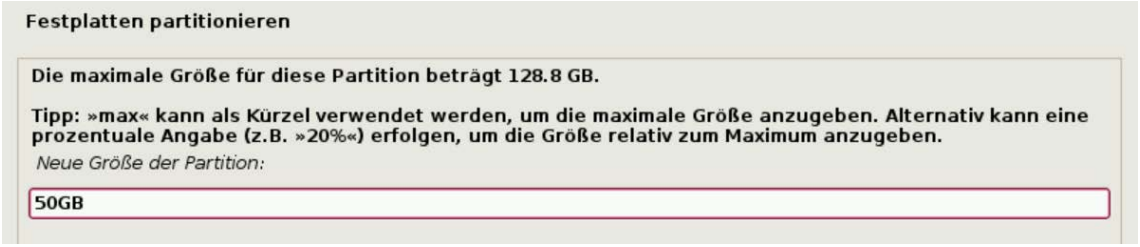


Select the free space and click on `continue`.

Step 2:

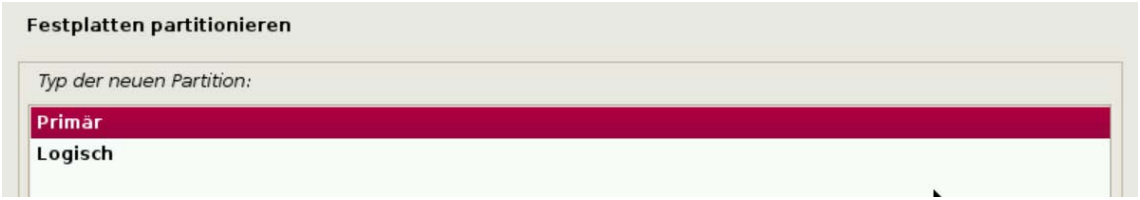
Mark "`Create a new partition`" by clicking on it and click on `continue`.

Step 3:



I usually give my system partition between 50-80 GB. Enter the desired value in the field and confirm your entry by clicking on `continue`.

Step 4:



Select for the system partition "`primary`" and click on `continue`.

Step 5:

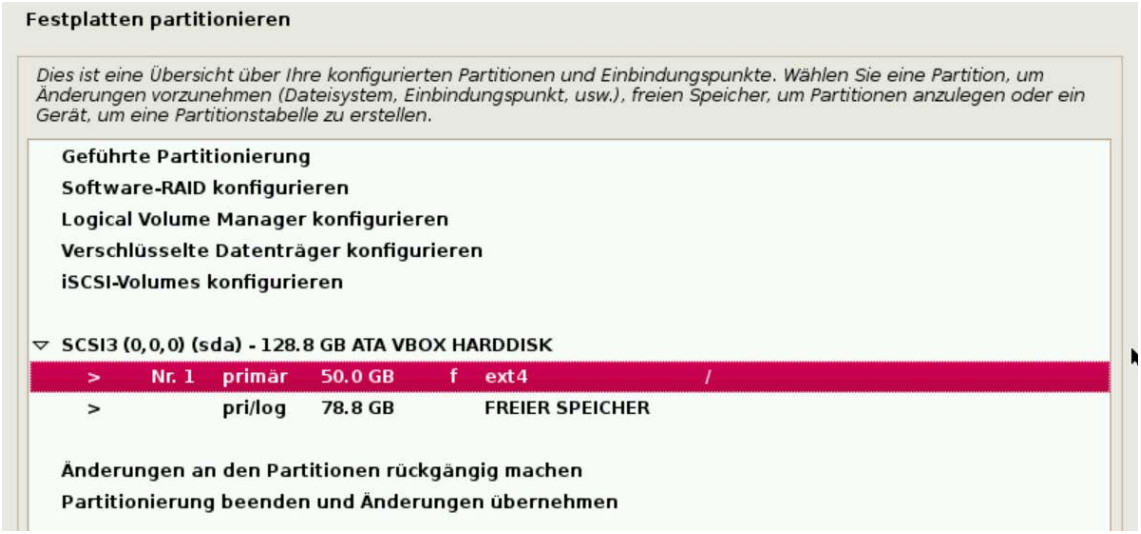
Select "`Begining`" and click on `continue`.

Step 6:

I prefer `Ext4` as the filesystem. If you are not sure what you are doing here, it is best to follow my recommendation and take `Ext4` as well. Unfortunately, I cannot go into the individual file systems in more detail, because that alone would fill an entire book.

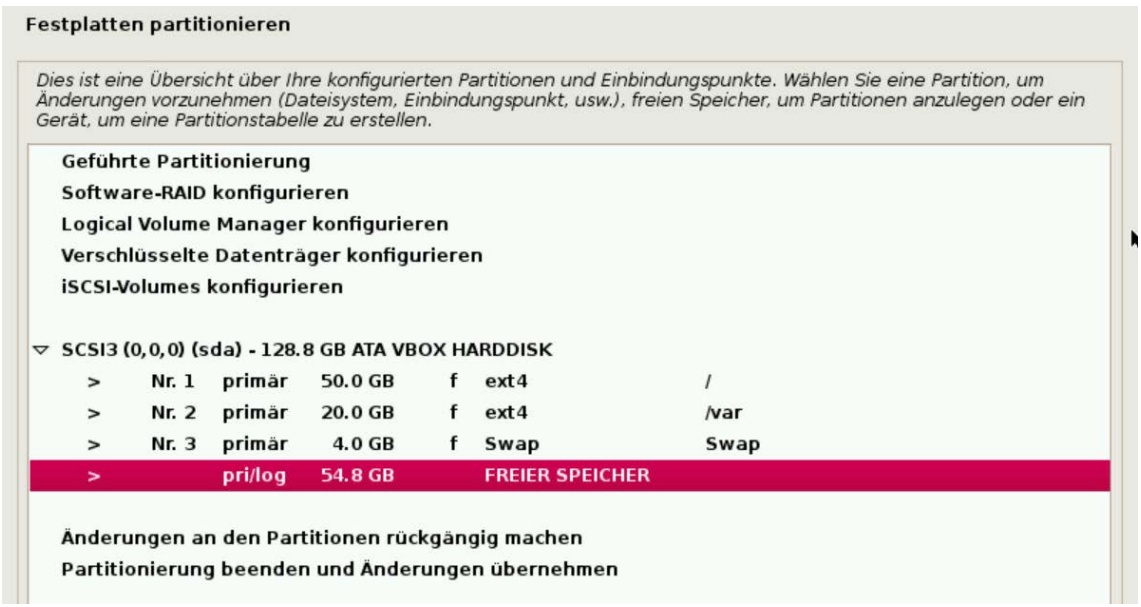
You can make changes here by double-clicking the line. Select `/` as the mount point, which corresponds to the root directory. What exactly that is will be discussed in detail later. You can leave the rest of the default settings.

Now select "`Finish creating the partition`" and click on `continue`. After that you should get the following window:



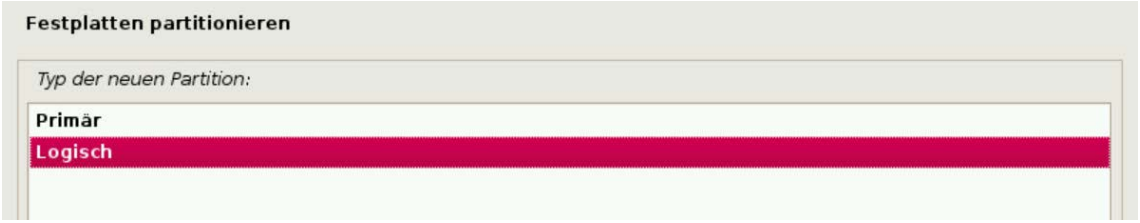
Then repeat steps 1 to 6. The partition size should be around 20-30 GB. Leave Ext4 as the file system and select `/var` as the mount point.

Then repeat steps 1 to 6 again. Now the partition size should be about twice the built-in RAM. This time select the option "`Swap memory`" instead of Ext4. With this, some options disappear and you cannot, for example, select a mount point. That is correct and you finish creating the partition. So far the partitioning should look something like this:



Since a maximum of four primary partitions is allowed, we have to proceed a little differently for the two missing partitions ...

Steps 1 to 3 remain the same. Whereby we allocate half of the available storage space in step 3. Kali saves us the math here and we can enter 50% right away.



In step 4 we now select "`Logical`" and confirm this with `continue`.

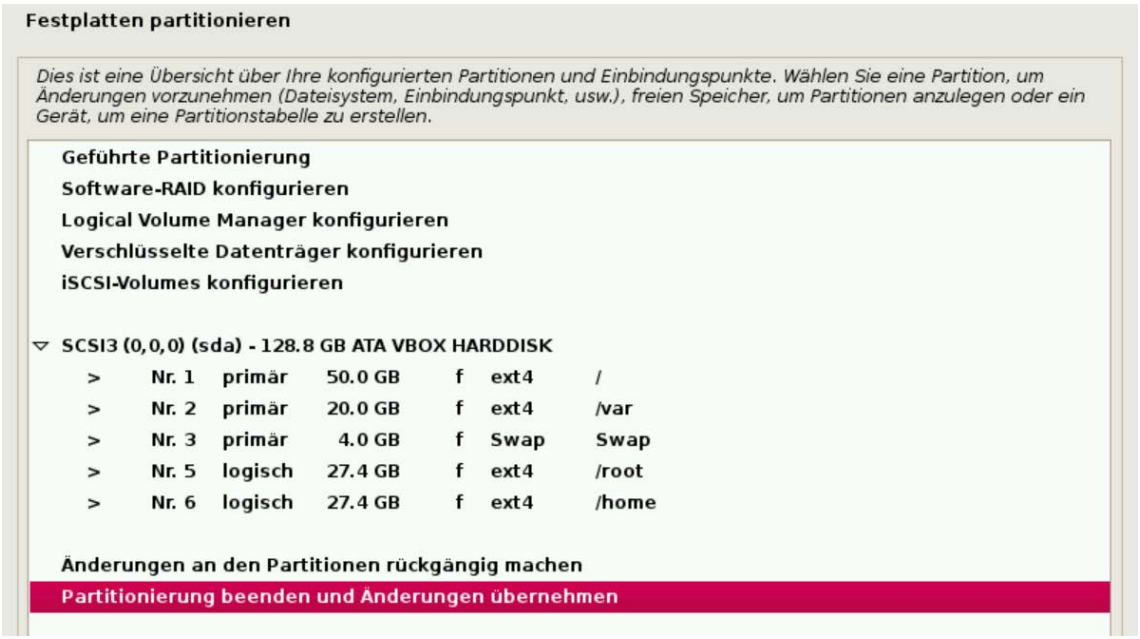
In step 6 we select again `Ext4` and the mount point `/root`. To do this, double-click on the line Mount Point and then select "`Enter manually`" and click on `continue`.

Now we have to enter `/root` in the field and confirm by clicking on next. And then finish creating the partition as usual.

Now we repeat the whole thing one last time and create a new partition. This time we are not asked whether we want to create a logical or a primary partition. Once the first logical partition has been created, only logical partitions can be created.

This time we will allocate all of the remaining space available and in step 6 select Ext4 as the file system and `/home` as the mount point.

The final partitioning should now look like this:

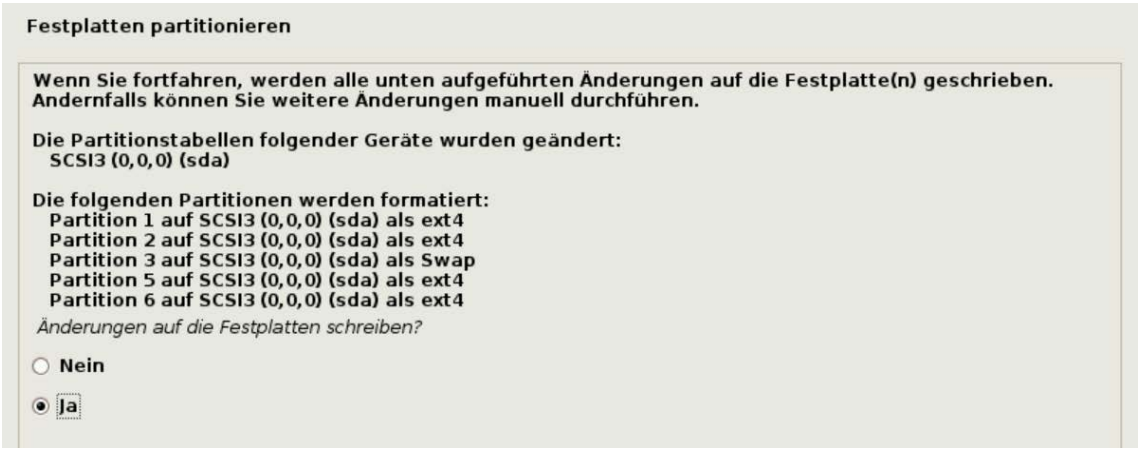


Now we have five partitions:

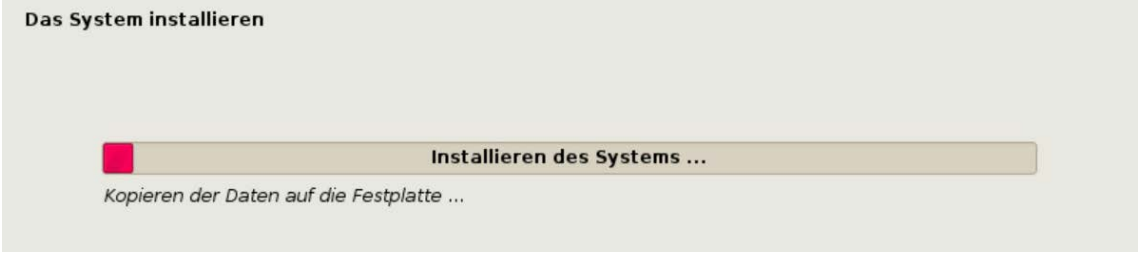
Size	Filesystem	Mountpoint
min. 50 GB	`Ext4`	**`/`**
min. 20 GB	`Ext4`	**`/var`**
2 x RAM	`swap`	**`swap`**
50% of the remaining space	`Ext4`	**`/root`**
50% of the remaining space	`Ext4`	**`/home`**

For a UEFI system, you also need an own EFI partition with up to 1GB!

Check this one last time, mark "`Finish partitioning and apply changes`" and click on `continue`.

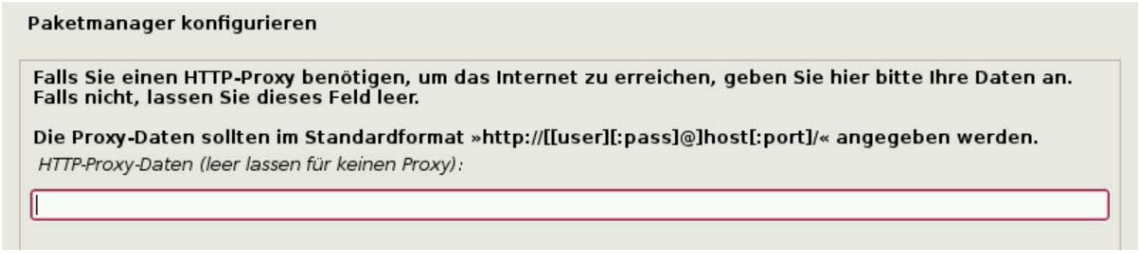


Confirm to write the changes to disk with `Yes` and click on `continue` to start the installation.



You can now see this window and relax a little. The system is installed in a few minutes. We will discuss the reason for the partitioning that has just been carried out in the chapter "Getting started with Linux".

As soon as the installation is complete, we will be asked if we want to install the latest updates right away. So we choose "`Yes`" and click `continue`.



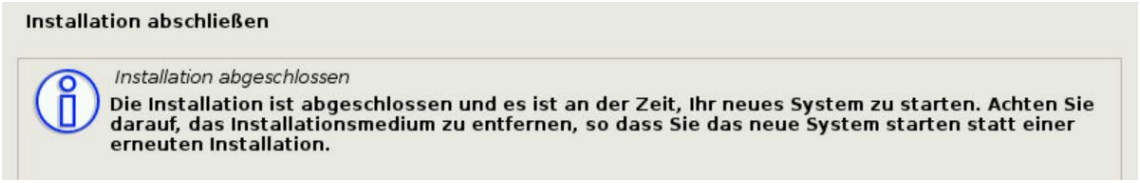
Then we will ask whether we need a proxy server for the internet connection. That will hardly be the case in your network. If so, you have to enter the access data for the proxy as described in the dialog. The entries in `[square brackets]` are optional.

The updates are then downloaded from the Internet and the GRUB boot loader installed. A boot loader is a small program that is located at a certain point on the hard drive and is executed when the PC starts. The bootloader is then responsible for initiating the start of the system.

Now it is asked whether we want to install the GRUB boot loader in the master boot record, `MBR` for short. We also answer this with "`Yes`" and click on `continue`.

Now select the disk on which you have installed Kali-Linux and click on `continue`.

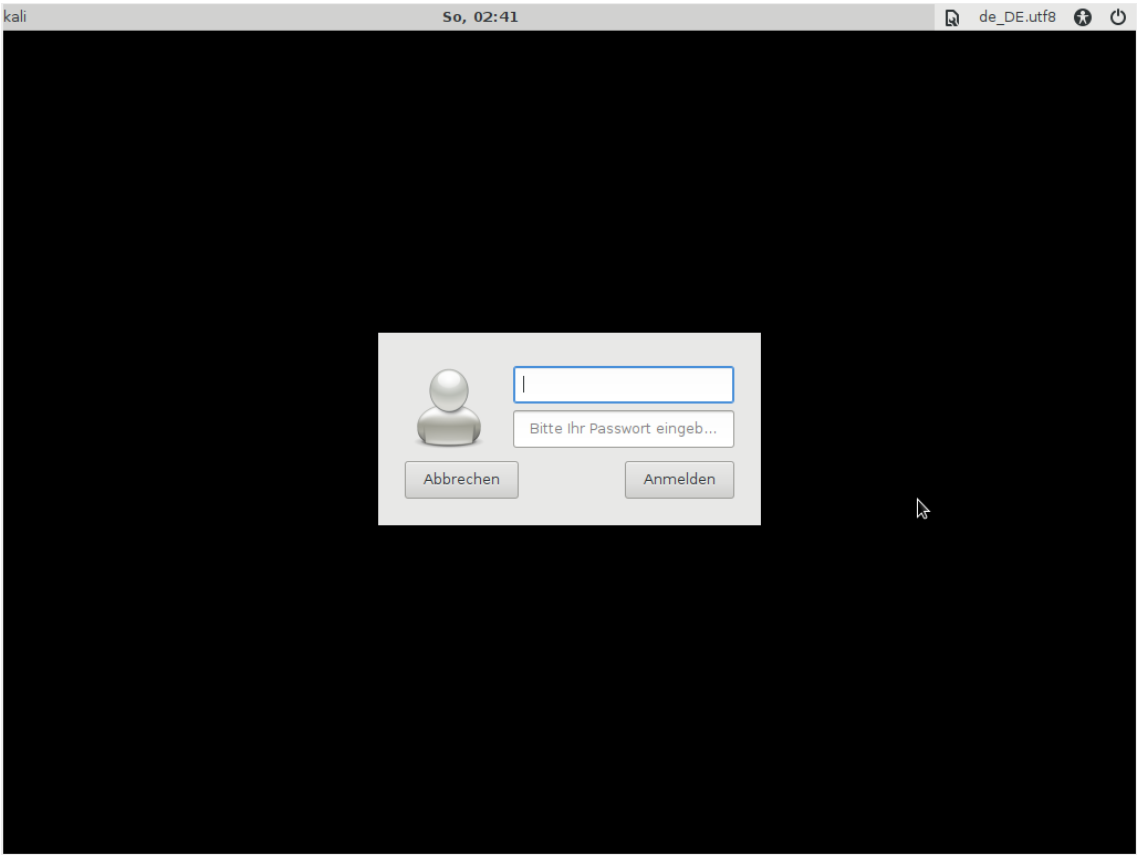
After a friendly pink loading bar has again informed us of the progress of the installation, we get to see the last screen of the installation:



Click on `continue` and make sure to remove the installation medium so that the computer does not reboot into the Kali installation.

Before the restart, we once again see our beloved loading bar, which informs us that the computer is being prepared for the restart and that the data garbage that is no longer required from the installation is removed beforehand.

If everything went smoothly during the installation you should see this screen after rebooting:

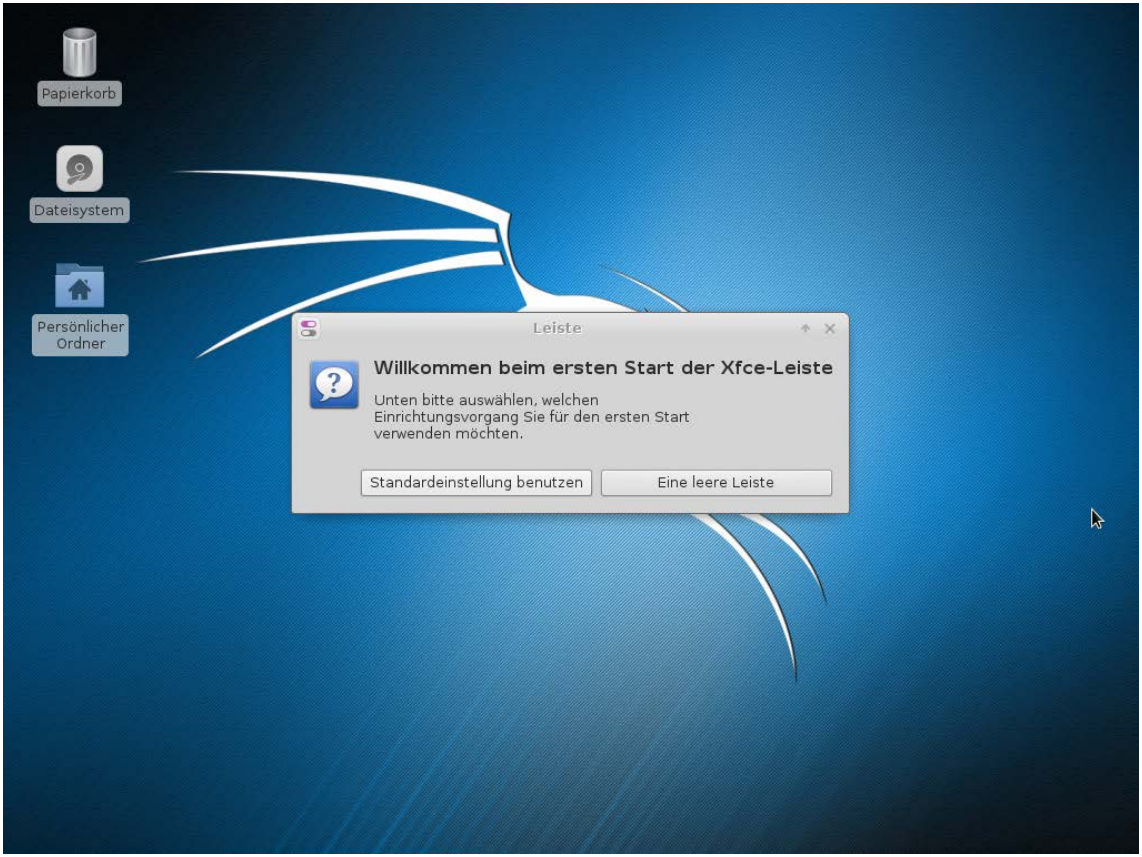


You must enter the user name in the top line. Since we have not yet created any other users, enter `root` here. Be sure to write `root` in lowercase letters. Linux is case-sensitive and therefore `root`, `Root` and `ROOT` are three different user names! As a convention, user names are always lowercase.

In the lower line, you then enter your chosen password and press `Enter` or the `Return` key.

At the first start, you will be greeted with the following message if you have decided on XFCE like me.

With Kali 2020.2 and later you create a normal user during setup. You can then log in as this user.



Select `Use standard configuration` and you can start using Kali.

Anyone who has ever set up Windows will miss the installation of drivers - this is usually not necessary under Linux. Very little hardware requires its drivers. Almost everything runs without problems with the included standard drivers.

If you want to save yourself unnecessary tinkering with the drivers, you should briefly clarify in advance whether all hardware runs smoothly with Linux. Usually, common hardware is supported and there are only problems with some more exotic components.

Getting started with Linux

Before we start with the configuration, however, I would like to discuss a few basic things about Linux with you.

Windows users are used to dividing their system into drives with certain letters as abbreviations. `C:\` corresponds to the system disk and `D:\` could be the data disk, for example. Then `E:\` would be used for the DVD drive and the attached USB stick would have `F:\` as the drive letter. Network drives from, for example, a NAS could then also be integrated as a drive (e.g. as `N:\`). Everything is "nicely arranged" and each drive can be addressed separately using a unique letter.

Not so with Linux! There is only one root directory `/` and the following folders are located in this:

Well, can you already guess how things are going? Do some of the folders look familiar to you?

Under Linux, the disks or partitions are bound to a mount point. In our partitioning, we have mounted the first partition as `/`, for example. Therefore all data is on this first partition with a few exceptions.

Next, we created a separate partition for `/var`. So all the data in the `/var` folder is on this second partition. The same applies to the `/root` and `/home` folders - each has its partition. All other data are on the partition that is mounted under `/`.

Sounds confusing and complicated at first, but if you think about it, it quickly becomes clear that this organization is much better. Server services store their data in the `/var` directory and the system log files also end up there. If the storage space in `/var` would run out, you could simply mount a second disk within `/var` and expand the storage space by moving the data from `/var/www/` to a disk and then mounting it there. This means that storage space can be expanded flexibly. This can be implemented even more conveniently with LVM or BRTFS, for example. BRTFS also supports snapshots of the file system and some other useful functions for storage servers. Therefore, the choice of the file system and the partitioning are crucial in professional use.

Imagine if a disk gets full in Windows. Then you would only have the possibility to distribute the data on two disks e.g. `D:\` (*data 2011-2017*) and `E:\` (*data 2018-today*). If customer data were now on these two disks, one would have to search for it on both disks to find the data of a customer over all years. And in a year or two the third disk would be added, etc. Of course, there is a workaround again. You simply get a larger disk and copy all the old data to the new disk, which costs time, and then only the remaining free space is available and the old disk collects uselessly in the cupboard. So Linux is much more flexible here - don't you think?

So that you can find your way around a little better, let's go through all the directories:

`/bin/`
Contains binaries. These are executable files (*programs*) of the core functions. For example, you can find the `ping` command here, which is used to check the availability of computers in the network.

`/boot/`
Among other things, it contains the GRUB boot loader and the start file with the name `vmlinuz`. The GRUB configuration files can also be found under `/boot/grub/grub.cnf`.

`/dev/`
Is the directory of the so-called device files. The hardware is addressed via these files during operation. Here we find, for example, `/dev/sda` or `/dev/sda1` etc. I should briefly explain how Linux names hard drives.

The `sd` stands for SCSI controller drive, older IDE disks were called `hd`. Since SATA disks are also seen as SCSI controllers in Linux, the prefix `sd` also applies here. The `a` stands for the first disk or the disk on the first controller. Therefore `/dev/sda` is, for example, the hard disk on the first SATA controller. The number after the drive letter is the partition number. This means that the first partition of this disk can be addressed with `/dev/sda1` and the entire disk with `/dev/sda`. This also includes USB sticks. Assuming there are no other hard disks in your PC, the first USB stick inserted would be `/dev/sdb` and its first partition `/dev/sdb1`. Here you can also find the files `/dev/cdrom` and `/dev/dvd`, which are a link to the actual CD or DVD drive. (*A link is almost like a shortcut in Windows.*)

`/etc/`
Contains configuration files and stands for "editable text configuration". Here you can find the file `/etc/fstab`, in which the mount points of the partitions are configured, or the folder `/etc/apt/`, which contains the configuration of the Kali update and installation mechanism.

`/home/`
Contains the directories of the normal users. There is a directory here for every user excep for `root`. As a rule, the standard configuration provides that a user only has read and write access to his home directory. But more about that when we look at the rights system.

/lib/
This is where the libraries or shared objects can be found. These are not large buildings full of books, but program libraries. These are parts of programs that can be loaded if necessary and they provide standard functions to several programs.

/lib64/
In principle the same as `/lib`, only that these are the 64-bit versions of the program libraries.

/media/
This directory serves as a collection point for subsequently attached removable media. For example, you can find the folder `/media/cdrom/` in which an inserted CD-ROM will be mounted. If you connect a pendrive a folder will be created here and the drive will be mounted in that folder.

/mnt/
This folder is usually empty. `mnt` stands for mount. For example, temporarily required data carriers can be mounted here.

/opt/
Here you can find programs that were installed manually and that bring their libraries. Manually installed programs should find a place in this folder so that there is no overlap with libraries that are always kept up to date by the update mechanism.

/proc/
Is a pseudo-directory. This means that everything that is here only exists in this form during operation. It can be seen as a kind of file and folder-based interface to RAM. Various things can be easily requested here - for example, the kernel version using `/proc/version` or more information about a running program. Each application is given a unique number (`PID`) when it starts, and a subfolder with this PID number is created in the `/proc/` directory. This then contains the information available - e.g.:

```
root@kali:~# cat /proc/1414/staus
Name:          xfce4-terminal
Status:        s (selleping)
Tgid:          1414
...            (Output shortened)
```

As you could see, the name of the program, the status, the PID and much more is provided. It is even possible to observe the memory allocation in the RAM and see what a program is doing exactly.

/root/
Is the home directory of the system administrator, called superuser or `root` under Linux.

/run/
Was introduced with the `systemd`. The `systemd` manages the system services and these store data in `/run/`. By the way, the `d` at the end of the name stands for Daemon, this is how the system services are called in Linux. In general, these files should save the status of the overall system. Older programs still access `/var/run/` for this.

/sbin/
Stands for superuser binaries and contains the executable files of the administration tools.

/srv/
Should contain files from system services. Usually not used at the moment.

/sys/
This directory is also very new and, like `/proc/`, consists of kernel interfaces.

/tmp/
As the name suggests, this is the place for temporary files. If you want to find a file after a restart, you should not put it here. The directory is emptied with every boot process.

/usr/
This abbreviation stands for Unix specific resources. You can find a lot here, from program documentation (`/usr/share/man/`) to various user programs (`/usr/bin/`) to program source code (`/usr/src/`).

/var/
Stands for variable data. Various server services store their data here. For example, you can find the base directory of the Apache webserver (*webroot*), the MySQL databases or the log files of the system, and most of the other server services.

If you need to diagnose a problem it's a good idea to check `/var/log/messages`, `/var/log/dmesg` or `/var/log/syslog` or some of the other files in `/var/log/` dedicated to a specific service!

User rights in Linux

For security reasons, Linux has an authorization system that is divided into the following three categories:

1. User of the file or folder
2. Group(s)
3. Others

To illustrate this, a small example - imagine a company. There is an accounting department. To illustrate this, the `accounting` group was created. There are two users in the `accounting` group - `meier` and `huber`.

Task 1 would be for the users to read each other's data, but not to change it. This is where the second part of the rights comes into play. Each object (*folder, file, link, etc.*) has three permissions. Read (`r`), write (`w`) and execute (`x`)!

The solution for this then looks like this - users `huber` and `meier` are both in the `accounting` group and their user directories have the following permissions:

```
root@kali:~# ls -lh /home
drwxr-x---   2 huber accounting 4,0K Feb  5 02:37 huber
drwxr-x---   3 meier accounting 4,0K Feb  5 02:37 maier
```

Let's examine this output:

`d .............` Directory (*it is a - in case it would be a file*)
`rwx ............` **r**ead, **w**rite, e**x**ecute - rights of the Owner
`r-x ............` **r**ead, e**x**ecute - rights for all members in the same gorup(s) as the owner
`--- ............` no access - rights for all others
`3 ..............` Number of hardlinks
`meier ..........` Username
`buchhaltung ....` Main group
`4,0K ...........` Size
`Feb  5 02:37 ...` Createion date
`maier ..........` File- or folder name

As a second task, the operations manager should now also have read access to the date of all people in the accounting department. This is why the users `huber` and `meier` also join the group `management`. This means that the members of the management can also access the data. However, this would also enable the users `huber` and `meier` to view the management's data. To prevent this from happening, the directory of the users from the management department must not have read rights for the group. The home directory then looks like this:

```
root@kali:~# ls -lh /home
drwxr-x---   2 huber  accounting  4,0K Feb  5 02:37 huber
drwxr-x---   3 meier  accounting  4,0K Feb  5 02:37 maier
drwx------   3 berger management  4,0K Feb  5 02:37 berger
```

It would now even be conceivable to create a folder named `Inbox` in each of the `huber` and `meier` user folders and then to assign write permissions for the group. This allows others to put documents virtually on your desk of a co-worker so that they don't have to permanently send documents internally by email. It would look then like this:

```
root@kali:~# ls -lh /home/huber
drwxrwx---   2 huber  fibu       4,0K Feb  5 02:37 Inbox
drwxr-x---   3 huber  fibu       4,0K Feb  5 02:37 Dokumente
usw.
```

Pretty practical and not even difficult to implement - don't you think so? There is one thing to consider. Besides reading or writing authorization, a folder always needs the authorization to execute. Without this, you cannot open the folder!

Let's go through some special cases using the system:

```
root@kali:~# ls -lh /initrd.img
lrwxrwxrwx 1 root root 33 Feb  5 01:52 /initrd.img -> boot/initrd.img-4.6.0-kali1-amd64
```

The `l` stands for a link. This is made even clearer by the arrow. It can be read so that `/initrd.img` points to `boot/initrd.img-4.6.0-kali1-amd64`. This happens a lot in Linux. In this case, it is the initial ramdisk and the link points to the latest version. If this should contain an error that prevents the system from booting, the administrator could simply change the link to the last working version and the problem would be solved.

```
root@kali:~# ls -lh /bin/su
-rwsr-xr-x 1 root root 40K Nov 12  2015 /bin/su
```

Here an `s` is set instead of the `x` in the authorizations of the owner. This stands for Set-UID and means that every user who runs this program inherits the permissions of the owner of the file. That makes sense in this case because the program `su` is used to switch to another user. You can change also to the root user by `su` without specifying the user name. And for such actions require root privileges because `root` is the only one who is above the Linux rights system and is allowed to do everything.

Every time you open a terminal you will see a line like this in front of the blinking cursor:

```
root@kali:~#.
```