```
nmap -p0-65535 10.0.0.36

  Starting Nmap 7.31
   ( https://nmap.org )
  Nmap scan report for 10.0.0.36
  Host is up (0.000086 s latency ).
  Not shown: 65506 closed ports
  PORT STATE SERVICE
  21/tcp open ftp
  22/tcp open ssh
  23/tcp open telnet
  25/tcp open smtp
  53/tcp open domain
```

```
Exploits, Bru
force attacks
Authenticatio
Passwords, Ka
Linux, nmap,
Metasploitabl
hydra, Wiresh
Attack Vector
Phishing, Pen
```

Hacking and Security

The Comprehensive Guide to Penetration Testing and Cybersecurity

Kofler · Gebeshuber · Kloep · Neugebauer
Zingsheim · Hackner · Widl · Aigner
Kania · Scheible · Wübbeling

Rheinwerk
Computing

Michael Kofler, Klaus Gebeshuber, Peter Kloep, Frank Neugebauer, Andrè Zingsheim, Thomas Hackner, Markus Widl, Roland Aigner, Stefan Kania, Tobias Scheible, Matthias Wübbeling

Hacking & Security

The Comprehensive Guide to Penetration Testing and Cybersecurity

Rheinwerk
Computing

Dear Reader,

These days most websites require you to have a profile to use them fully, and the guidelines for creating a usable password have become more and more complex. It's a joke within internet spheres that soon you'll need a DNA sample just to be able to access your social media account successfully. Although comments like these are obviously a joke, the real fear of getting sensitive information stolen is evident in the rise of websites or browser add-ons with the main purpose of either saving or creating unique passwords.

As the age of technology moves forward at breakneck pace and storing sensitive information online becomes even more normalized, it's more important than ever to stay informed if you're a cybersecurity professional. Enter *Hacking and Security: The Comprehensive Guide to Penetration Testing and Cybersecurity*, a guide to help beginners and seasoned professionals alike navigate the cyber landscape with confidence. Our expert author team will teach you to use ethical hacking and other cybersecurity techniques to uncover security vulnerabilities and harden your sensitive systems against attacks.

What did you think about our book? Your comments and suggestions are the most useful tools to help us make our books the best they can be. Please feel free to contact me and share any praise or criticism you may have.

Thank you for purchasing a book from Rheinwerk Publishing!

Kyrsten Coleman
Editor, Rheinwerk Publishing

kyrstenc@rheinwerk-publishing.com
www.rheinwerk-computing.com
Rheinwerk Publishing • Boston, MA

Imprint

This e-book is a publication many contributed to, specifically:

Editor Kyrsten Coleman
Acquisitions Editor Hareem Shafi
German Edition Editor Christoph Meister, Anne Scheibe
Translation Wenema Language Services, Inc.
Copyeditor Melinda Rankin
Cover Design Graham Geary
Photo Credit Shutterstock.com: 186627704/© Nneirda, 1472495273/ © Alexander Supertramp
Layout Design Vera Brauner
Production E-Book Hannah Lane
Typesetting E-Book SatzPro, Germany

We hope that you liked this e-book. Please share your feedback with us and read the Service Pages to find out how to contact us.

The Library of Congress Cataloging-in-Publication Number for the printed edition is as follows: 2023019445

ISBN 978-1-4932-2425-8 (print)
ISBN 978-1-4932-2426-5 (e-book)
ISBN 978-1-4932-2427-2 (print and e-book)

Contents

Preface

News coverage of hacking attacks and security breaches affecting millions, sometimes billions, of devices is ubiquitous. It has brought the topics of hacking and IT security increasingly to the fore in recent years and has also created an awareness among "normal users" that the security of IT infrastructure affects everyone.

Many computer, smartphone, or, more generally, internet users are in danger of resigning themselves to the many risks. It's clear to most that "proper" passwords should be used and that updates should be applied regularly—but beyond that, users feel largely unprotected against the dangers of increasing digitization.

In fact, it's primarily the task of administrators, IT managers, and software developers to ensure greater security. Increasingly stringent legal requirements and the loss of image associated with security breaches are forcing companies to take a more intensive look at security. It's no longer enough for a device to simply work, for software to look "fancy," or for smartphones to be packaged in stylish, ever-thinner cases. The hardware and software, along with the associated server and cloud infrastructure, must also be secure—at least as secure as is currently technically possible.

What Hacking Has to Do with Security

Hacking is the colloquial term for finding ways to bypass the security measures of a program or system or to exploit known security gaps. The goal is usually to read or manipulate private or secret data.

Hacking often has a negative context, but it's not always a bad thing: when a company commissions a so-called penetration test to verify the security of its own IT infrastructure by external persons, the penetration testers use the same tools as criminal hackers. The same is true for security researchers trying to find new vulnerabilities. This is often done on behalf of or in collaboration with large IT companies, universities, or government security agencies. Whether a hacker is "good" or "bad" depends on how he or she behaves once a vulnerability has been discovered.

If you're an administrator or IT manager responsible for the security of a system, you need to know the tools that hackers use. To defend yourself or your company, you need to know how attackers operate. In that respect, this book is very concerned with giving you an overview of the most important hacking tools and techniques. However, we don't stop at that point. Rather, we'll focus on how you can defend yourself against attackers, what defensive actions you can take, and where you can improve the configuration of your systems. To put it another way: for this book, hacking is the *means*, rather than the *end*. The goal is to *achieve a higher level of safety*.

About this Book

In this work, we want to provide a broad introduction to the topics of hacking and IT security. With almost 1,200 pages on offer, it may sound like an understatement to speak of an "introduction." But the reality is that both hacking and security are immeasurably large areas of knowledge.

One could write a separate book on almost every topic we address in this book. In addition, there are all the special topics that we don't even touch upon in our book. In a nutshell: don't expect this book to be all-encompassing or that by reading it you will already be a hacking and security expert.

That being said, there has to be a starting point if you want to get into hacking and security. We tried our best to give you a good starting point with this book. Specifically, after an introduction to our range of topics, we'll address the following subjects:

- Kali Linux (distribution with a huge collection of hacking tools)
- Metasploitable and Juice Shop (virtual test systems for trying out hacking)
- Hacking tools (`nmap`, `hydra`, Metasploit, Empire, OpenVAS, SET, Burp, Wireshark, and so on)
- Offline hacking; access to other people's notebooks/hard drives
- IT forensics
- Password hacking; secure handling of passwords
- Wi-Fi, Bluetooth, and radio communication
- USB hacking and security
- Implementation of external security checks
- Penetration testing
- Basic coverage of Windows and Linux, Active Directory, and Samba
- Intrusion detection systems and Snort
- Exploit basics of buffer overflows, fuzzing, heap spraying, microarchitecture vulnerabilities (Meltdown and Spectre).
- Cloud security, focusing on Amazon S3, Nextcloud/ownCloud, Microsoft 365
- Hacking and security of smartphones and other mobile devices
- Attacking and securing web applications
- Securing and secure development of IoT devices
- Bug bounty programs

The wide range of topics explains why this book has not one author, but 11. A brief introduction to our team can be found at the end of the book.

What's New in the Third Edition

For this edition, we've comprehensively updated the book and added much new content. This includes, in particular, the following:

- IT forensics
- Intrusion detection systems and Snort
- Bug bounty programs
- Sliver, Starkiller, and MalDuino
- Purple teaming
- Linux kernel hardening

Target Group

This book is intended for system administrators, security managers, developers, and IT professionals in general who already have some basic knowledge. To put it bluntly: you should at least know what PowerShell or a terminal is. And you must be willing to think across operating systems: neither hacking nor IT security is limited to Windows or Linux computers today.

Pure IT users, on the other hand, are not in the focus. Of course, training computer users is an indispensable aspect of improving IT security both at home and in businesses. However, a compilation of more or less trivial rules and tips on how to use computers, smartphones, and the internet in general safely and responsibly does not seem to us to serve a purpose in this technically oriented book.

Let's Go!

Don't be put off by the size of the subject area! We've tried to divide our book into manageable chapters. You can read most of them largely independently to learn the ropes step by step, gain hacking expertise, and develop a better understanding of how to better secure your own systems. You'll quickly discover that a more in-depth look at hacking and security techniques is incredibly fascinating.

With our book, we hope to contribute to better management of IT security in the future than has been the case so far!

—Michael Kofler, on behalf of the entire team of authors

Foreword by Klaus Gebeshuber

Experience from numerous penetration tests shows that many administrators of computer systems and networks hardly know about the capabilities and audacity of hackers. An attacker needs exactly one vulnerability to penetrate a system; a defender needs to prevent many of the possible attacks. There are no rules; no path is off-limits to a hacker.

I've always been fascinated by the extreme creativity of and technical capabilities and variants that have been implemented by hackers. I've always wanted to know what the bad guys can do so I can use the knowledge to strengthen the good side. The book *The Art of Intrusion* by Kevin Mitnick (Wiley 2005) sparked my curiosity about the subject even more.

It is also a great concern of mine to show young people the fascinating technical possibilities on the one hand while also motivating their future work on the good side. The European Cyber Security Challenge, with local qualifications for pupils and students in 24 European countries and a European final, provides a great opportunity to discover and promote young security talents.

Foreword by Stefan Kania

I have often noticed that some aspects of security are ignored when it comes to Samba servers. Frequently, Samba shares are given permissions to prevent unauthorized access, but the security of the operating system is then sometimes neglected. A Linux host with Samba as a file server must always be viewed from two angles. I always address this in my seminars as well. For a long time, I wanted to describe this view of Samba systems in more detail.

That's when I got the request from Rheinwerk Publishing for this book, which was exactly what I had imagined. It's not just about configuring a Samba server, but about setting up a Samba server as securely as possible. The framework of the book covering various tools, services, and devices is also just right for the topic. So here's a book that I myself have always wanted. I'm very pleased that I can now contribute to it with my chapter. I hope you, reader, will enjoy this book as much as I did.

Greeting

IT security is a topic that no one can ignore. The German public is regularly startled by hacking incidents: In 2020, a cyberattack on Duesseldorf University Hospital led to the hospital having to sign off on emergency care and cancel surgeries. In 2021, Bitkom reported that annual damages from hacker attacks had exceeded 220 billion euros. At the same time, the highest number of new malware variants ever measured was recorded. And recently, since the start of the war against Ukraine in February 2022, the full implications of cyberwar are also being felt.

Thus the motto is: IT security must be at the top of the priority list—for companies, organizations, and the public sector. But IT security should also play a more prominent role for private users.

Attacks on IT systems are very attractive for perpetrators. From online payments and business processes to cloud-based services and the Internet of Things (IoT), digital infrastructures offer a large field of attack. The anonymity of the web lowers the inhibition threshold for attempting such attacks.

Anyone who cuts corners when it comes to IT and data security is ill-advised. If, on the other hand, you succeed in teaching your own employees how hackers think and act, you're already a big step closer to a robustly secured IT infrastructure. Those who understand their attackers are better defenders.

This compendium therefore goes in exactly the right direction with its concern: "For this book, hacking is the means, rather than the end. The goal is to achieve a higher level of safety," the preface states. I can only support this: as managing director of SySS GmbH, I am responsible for 90 IT security consultants who do nothing else every day but "hack" our customers' systems on demand.

Such penetration tests quickly and efficiently detect security gaps. IT managers can then fix them—before illegal hackers exploit them. At the same time, such a test and the associated final report also show our customers in detail how we act to detect and exploit weaknesses.

It is precisely such knowledge that is of inestimable importance when it comes to making one's own systems ever more secure. The book *Hacking and Security* provides this know-how for practical use. I can only warmly recommend to anyone who is professionally involved in IT security to read it. Stay one step ahead of the "bad" hackers.

—Sebastian Schreiber, Managing Director SySS GmbH

Chapter 1
Introduction

This chapter provides a first introduction to the huge topic of hacking and security and answers the following basic questions:

- What is hacking? Are there good and bad hackers?
- What is security?
- Why is software so insecure?
- What are attack vectors? Which attack vectors exist?
- What are (zero-day) exploits?
- What is the purpose of penetration testing?
- What laws and standards apply to hacking and security?

Because you bought this book, you're obviously interested in these topics and probably have prior knowledge. Nevertheless, we advise you to take a closer look at this relatively nontechnical chapter. It introduces terms and concepts used throughout the book. Even IT professionals, mostly specialists in a rather narrow field, are rarely familiar with the diverse terminology of the security world. Thus, not only is this chapter an introduction, but it also aims to provide a linguistic basis for a better understanding of all subsequent chapters.

1.1 Hacking

Wikipedia defines a *hack* as an action to break or bypass the security mechanisms of a system. A hack in this context is therefore an unintended way of breaking into a system, changing, manipulating, or destroying data. (A hack can also be a messy, quickly created solution to a problem or the misuse of a device to perform other tasks. But that's not the subject of this book).

Accordingly, *hacking* is the search for hacks and a *hacker* is the person who deals with them. In the media, the term *hacking* is usually used in a negative or criminal context, but that's not correct. Hacking in itself is value-neutral. Just as a knife can be used equally to cut vegetables or kill someone, finding a hack can be used to improve the security of a system or to attack the system and cause damage.

Rules also apply to hackers. On the one hand, laws prohibit any unauthorized data manipulation, sometimes including even the attempt to penetrate a computer system.

On the other hand, the hacking community has repeatedly defined its own ethical rules. Admittedly, there's no international standard for this. Rather, what a hacker may or should do depends heavily on cultural and political contexts. From this point of view, hackers are sometimes divided into three groups, although the boundaries cannot always be drawn exactly:

- *Responsible hacker*s abide by both laws and hacker ethics. They use their knowledge to improve the security of computer systems, share discovered security vulnerabilities with affected manufacturers, and so on. The term *ethical hacking* is used for this type of hacking.
- *Criminal* or *malicious hackers* use their knowledge for criminal activities and accept that their activities cause damage.
- In between there are hackers who don't play by the rules but pursue higher goals, such as improving society or using technology more responsibly. There's a large gray area here that makes a clear distinction between good and evil difficult or dependent on one's social or political position.

Politically Correct?

The hacker types just outlined are often referred to as *white hats*, *black hats*, and *grey hats*. In 2020, security expert David Kleidermacher initiated a discussion about these terms because they could be interpreted in a racist way. Many members of the community, on the other hand, argue that *white* and *black* in this context have nothing to do with skin color, but with the dualism between day and night, or with the colors of hats in old Westerns. (In some such films, the villains wear black hats.) For more, see *http://s-prs.co/v569600*.

The derogatory term *script kiddies* refers to people who, without in-depth knowledge, carry out hacking attacks with programs or scripts that are easy to find on the internet and sometimes cause great damage. But it's debatable whether script kiddies also count as hackers. In any case, the term *cracker*, which was suggested for better differentiation, has not caught on.

1.1.1 Hacking Contests, Capture the Flag

Hacking needs to be learned. Of course, you can read books like this one and try the techniques presented here yourself. Much more entertaining, and especially popular in IT student circles, are hacking competitions. In these competitions, participants are given access to specially prepared computer systems, usually in the form of virtual machines. The objective is often to penetrate the system and find hidden "treasures" ("flags") in it as quickly as possible. The collective name for such competitions is *capture the flag* (CTF). Frequently, participants are not only individuals, but entire teams.

There are also variants of the classic CTF competitions in which, for example, each team receives a server. The goal then is to protect your own server against the attacks of the other teams and at the same time to attack and “conquer” the servers of the other teams. Individual subtasks are rewarded with points. The team that scores the most points is the winner.

There are various sites on the internet where virtual machines from former hacking competitions are available for download (search for “hacking ctf images”, for example). With these downloads, you can try the former competition content for yourself and see how far you would get. Often, there are also more or less concrete solution instructions (search for “hacking ctf writeups”).

Metasploitable and Other Virtual Machines for Practice

Beginners often are overwhelmed by the mostly very specific tasks in hacking contests. A better place to start is with purpose-built virtual machines or Docker images that use outdated versions of popular software. Moreover, these machines are prepared with various security vulnerabilities, which almost guarantees a certain sense of achievement.

We’ll introduce the most popular of these test systems in Chapter 3.

1.1.2 Penetration Test versus Hacking

A *penetration test* (*pen test* for short) is a comprehensive security test for a computer system (see also Chapter 10 and Chapter 11). Often a person or organization from outside the company is commissioned to do this. The pen testers try to act like hackers—that is, attack the system and find security gaps. This means that the same working techniques are applied. The main difference between hackers and pen testers is therefore not so much in the way they work as in the fact that pen testers have an explicit mandate for their work, and they do not manipulate or destroy data as part of their tests but report the defects they find so that they can then be fixed.

But pen testers have a big advantage over hackers: they don’t need to operate in secret. A hacker usually won’t start his attack with a large scan because its intensive tests will set all alarm bells ringing on a well-secured server. A pen tester acting in agreement with the company, on the other hand, can use such tools without any problem.

1.1.3 Hacking Procedure

When it comes to accessing foreign data, manipulating it, or otherwise causing damage to IT systems, there are many paths that lead to the goal:

- **Network hacking**

  In a sense, this is the "classic" type of hacking; it's done via network connections. For example, it exploits insecure passwords, sloppy configuration, or known bugs to perform the attack. The goal is mostly to gain unrestricted access to the computer either directly or by guessing/listening to a password or password hash (root access).

  Variants of this are fictitious websites for password entry (*phishing*) or the exploitation of programming errors in order to execute one's own code or SQL statements on websites (HTML injections, SQL injections, and so on; see Chapter 17).

- **Password hacking**

  Knowing the correct password provides the easiest way into the attacked computer. Accordingly, many techniques are aimed at finding a password. These include systematic cracking, logging of all keystrokes by software or hardware (*key logging*), reading and reusing password hashes, and so on. However, most of these methods already require access to the computer, either via the network or physically (e.g., to apply a USB key logger or to tap the wireless keyboard).

- **Backdoors**

  An attacker can save himself all hacking effort if he knows about a so-called backdoor into a program or even installs it himself. In the simplest case, this is a combination of a login name and password known only to the manufacturer, as is common for many routers, mainboards, and the like. It's rarely possible to prevent these passwords from being discovered and published on the internet sooner or later. However, the backdoor also can use a much more sophisticated mechanism.

  With open-source software, permanent backdoors can almost be ruled out; they would be conspicuous in the publicly accessible code. However, there have been cases in which a hacker has offered a modified version of an open-source program for download. Such manipulations are easy to accomplish and are often noticed only after some time has passed. That's why it's recommended to download software only from official websites in general and take the trouble to check the checksums. (In real life, of course, one must assume that at best only a few enthusiasts with an affinity for security will make this effort.)

  The situation is quite different for commercial software for which source code isn't available. There are countless conspiracy theories floating around the web that manufacturers or intelligence agencies routinely build backdoors into operating systems and communications software. After the Snowden revelations, this can't be ruled out entirely. And as neither the existence nor the nonexistence of a backdoor can be proven due to the lack of source code, this uncertainty will prevail

- **Bugdoors**

  It's even more difficult to prove the existence of so-called bugdoors. These are errors—*bugs*—that pose a security problem and make it appear as if they were intentionally built in.

Software contains errors; that's incontrovertible wisdom. It's impossible to say whether bugs were inserted on purpose without knowing the developers' intentions. For this reason, it's very difficult to work with this category. However, a bad taste remains when you look at the quality of the code that led to security vulnerabilities

- **Viruses, worms, and other malicious software**

  A piece of malicious software (*malware*) is a program that performs unwanted functions on a computer or device. Depending on how such software spreads or is disguised, it can take the form of viruses, worms, Trojan horses, or backdoors.

  The technical design adapts over time to the IT infrastructure currently in use. Whereas the first viruses were spread via floppy disks, email has become the most popular means of transfer in the last decade.

  Malware is also extremely commonly encountered on smartphones (see Chapter 22). A classic example is a flashlight app in which, behind its intrinsically useful function, other functions for spying on the user are hidden. Today, the disguise is mostly better, but the idea has remained the same.

  The objectives of malware also change and are subject to fashion trends. Encryption programs (*ransomware*) that first encrypt as many files on the hard drive as possible have been particularly popular recently. This has been driven to perfection by the Emotet malware, which has caused hundreds of millions of dollars in damage worldwide in recent years.

  The key required to restore one's own data after a ransomware attack can be purchased (ransomed) from the blackmailers. This business model works so well that criminals can combine predefined components and buy their own encryption Trojan on corresponding websites with just a few clicks (*cybercrime as a service*).

- **Denial of Service (DoS): Denial-of-service**

  attacks have a completely different approach. Their sole purpose is to disrupt the operation of a company or access to a disliked website by sending so many requests that regular operation is no longer possible. DoS attacks work particularly well if a software bug can be exploited at the same time to specifically crash the server's software.

  Botnets are often used for DoS attacks. A *botnet* is a network of computers or devices that have been previously brought under the hacker's control using other methods. A botnet can be used to coordinate and send hundreds of thousands of requests per second to a particular server until it becomes overwhelmed by the onslaught and stops responding properly. This type of attack is referred to as a distributed denial of service (DDoS).

  Individual companies are usually not in a position to defend themselves against a targeted DDoS attack. This requires the help of the companies responsible for the internet infrastructure. These companies can, for example, intervene in large network nodes with filters or firewalls.

Particularly Dangerous in Combination

In practice, many attacks utilize multiple exploits and apply different methods simultaneously. Sophisticated hackers always manage to carry out a successful attack by combining vulnerabilities that are relatively harmless in themselves.

1.1.4 Hacking Targets

The number of hacking targets has increased dramatically in recent years. While "classic" hacking was directed against computers or servers, it's now also necessary to keep an eye on smartphones and all networked devices. These include network routers, switches, firewalls, printers, TVs, Wi-Fi- or Bluetooth-enabled loudspeakers, automatic vacuum cleaners, web cameras, other electronic devices and gadgets (Internet of Things [IoT] devices), heating, ventilation, and shading systems (home automation), electronic doors and locks, cars, airplanes, medical equipment, industrial facilities, and much more.

The cloud is a topic in itself. By its very nature, the cloud consists of computers or virtual machines that can be attacked as such. At the same time, however, the cloud system as a whole is also a target for attack: countless secret documents have already been downloaded from the Amazon cloud because an administrator overlooked the fact that the directories in question were publicly accessible without any protection. (However, it's debatable whether taking advantage of such negligence has anything to do with hacking).

Attacks on subcomponents of a device, such as a Wi-Fi chip or a CPU, go in a completely different direction. For example, in the fall of 2017, it emerged that many Intel CPUs produced over a two-year period have a mini operating system with management functions at the lowest level—the *management engine*. (Strictly speaking, this is an adapted Minix—that is, a tiny Unix variant developed for training purposes).

One might argue about who needs such functions at all, but the matter becomes disastrous when it turns out that the CPU and any software running on it can be attacked via these management functions due to basic and partly trivial errors. It's no wonder that some critics even suspect a backdoor here.

In early 2018, the next CPU-level security disaster was revealed: A flaw in several CPU architectures, which is particularly severe in Intel models, allows processes to access isolated memory areas of other processes. The error is so elementary that there is a whole range of attack variants. The two most important ones were given the names Meltdown and Spectre (see Chapter 18, Section 18.10).

These bugs affect billions of devices. Updates on the CPU level (via microcode updates) are only partially possible. For this reason, all operating systems (Windows, macOS, Linux, iOS, Android) and web browsers have to be adapted so that their code virtually bypasses the CPU bug—at the price of reduced system performance. Because many devices will never receive the required updates, this bug will probably have an impact for years to come.

Meltdown and Spectre were unfortunately just the beginning. Once on the right track, security researchers found a whole series of related vulnerabilities. Although there are bug fixes for these as well, they are associated with further speed losses.

Similarly problematic to CPU errors are errors in GPUs or in network chips. For example, the Kr00k security gap, which affects Wi-Fi chips made by Broadcom and Cypress, was discovered at the beginning of 2020. These chips are estimated to be installed in more than 1 billion devices (mainly smartphones)! While software updates are available, it's unclear how many devices will ever receive these updates.

You can see that errors at the hardware or firmware level are becoming more and more frequent, and their scope is enormous—on the one hand, such errors can be exploited regardless of the operating system, and on the other hand, they are particularly difficult to fix through updates. Although firmware updates are possible for most chips, their implementation is complicated in many operating systems, and not provided for at all with others. For employees responsible for the security of a company or organization, this is a nightmare: Do all PCs, smartphones, routers, and so on that do not have a firmware update available now have to be taken out of service? Who will pay or justify the associated costs?

Instead of attacking hardware components, hackers can also exploit flaws in software components, such as programming errors in libraries or design flaws in application programming interfaces (APIs). In this context, the best known example from the recent past is named Log4Shell. The hack is based on the very popular open-source library `Log4j`, which is used in many Java programs to log messages. Unfortunately, in 2021 it was found that many programs that use `Log4j` use it to provide an almost trivially easy way to execute foreign code—a paradise for any hacker.

In this case, it's even debatable whether the library is or was defective at all: in fact, using a particularly elegant logging syntax, the library has worked exactly as described since 2013. The fact that the mechanism can also be misused did not become known until eight years later.

The error behavior (or the too universal application possibility) of the library was quickly fixed after it became known. Nevertheless, countless vulnerable programs are still in use today. Every program that uses `Log4j` must be recompiled and then updated on the customer's side or on each respective computer or device. And this is precisely where the problem lies: there is a lot of software that is no longer maintained or for which the distribution of updates (e.g., in IoT devices) is very costly.

1.1.5 Hacking Tools

To facilitate hacking, countless programs have been developed. The range extends from simple scripts for a network scan to comprehensive analysis tools that systematically scan a server or device for all known security gaps and problems.

In addition, there are programs that were originally designed to analyze network, Wi-Fi, or Bluetooth issues or for similar tasks, but which can of course be perfectly misused for other purposes. Much of this software is available free of charge on the internet, often even in source code (open-source concept).

In addition, there are companies that focus on this segment and sell software for very specific hacking tasks, sometimes in an upscale price segment for elite target groups (police, intelligence agencies, military, international security companies).

In this book, we'll focus on common tools that are available for free and are correspondingly common in practice (see Chapter 4). Instead of searching for and downloading each hacking tool separately, many hackers and pen testers turn to complete toolboxes that provide a huge collection of tools in the form of a toolkit. The best known toolkit in this context is Kali Linux (see Chapter 2), a Linux distribution that bundles several thousand hacking programs that run on Linux.

Hacking Hardware

Hacking tools are by no means limited to software. An entire market has now established itself for hacking hardware. The offer starts with simple "gadgets" that look like a USB stick but behave like a keyboard and quickly open PowerShell on Windows, download malware with a command, and execute it. If the target doesn't manage to stop this process within two or three seconds, then it's already too late.

However, there are also much more intelligent devices, which in fact are inconspicuously packaged mini computers. If the hacker manages to place these devices correctly (this usually requires physical access to the target's computer), he can use them to hack into network, USB, or Bluetooth communications or perform other tasks.

In Chapter 9, we'll introduce some such hacking gadgets and show you how you can protect yourself against them. A whole range of other hacking devices has been developed over the last few years.

Finally, the Raspberry Pi is recommended as a quasi entry into the world of hacking hardware: This minicomputer is not designed for hacking tasks, but it can be configured as a Wi-Fi access point in no time. Hackers can use it, for example, to try to lure their targets into a free, but unfortunately unencrypted, Wi-Fi connection. Subsequently, all sorts of nastiness can be realized, such as manipulating DNS records to redirect the target to phishing websites.

1.2 Security

So what does hacking have to do with security? At first glance, they seem to be opposing concepts. The goal of this book is to help you secure computer systems. To do this, you need hacking knowledge for two reasons:

- First, you need to know what means and tools are commonly used to carry out attacks. A complete description of all hacking tools is beyond the scope of this book, but we try to provide you at least with a first overview.
- Second, it's important to understand why computer systems and software are vulnerable. That's why we go into the basics and internal details of security gaps (exploits) in several chapters.

The key approach in this book is first to show you how easy it is in many cases to attack a system; that's the hacking aspect. Then the second step is to take defensive measures. So our motto would be: More Security through Hacking.

To exaggerate a bit, one could even say: "Offense is the best defense!" So by attacking your own systems yourself or through pen testers commissioned for this purpose, you can learn about your systems' weaknesses and take appropriate protective measures.

We certainly do not want to raise false hopes: it isn't possible to achieve 100% security with current IT technologies. However, in no way does this mean that it isn't worth improving safety! Many cybercriminals simply look for the targets that require the least effort to attack. Even a few simple security measures thus can make all the difference.

Therefore, the measures presented in this book will not be sufficient to ward off professionally executed corporate espionage or even a hacking attack by an intelligence agency. Protection against state-sanctioned cyberattacks is clearly outside the scope of this book.

Security in the Context of This Book

When we talk about *security* in this book, we mean security from hacking attacks only. By its very nature, security goes much further. If you care about your business or your organization, or if you want to protect yourself from the legal consequences of negligent handling of someone else's data, then you need to consider entirely different factors.

What happens when a hard drive unexpectedly fails? When an excavator accidentally cuts the network connection to your office? If the company building or server location is destroyed by fire? Are there decentralized backups? Are there any security guidelines? Are there specific lists of responsible persons, tasks to be performed for a disaster? More generally, are there any contingency plans in place?

Naturally, there are a lot of security measures that not only protect against a hacking attack but also help in other emergencies. But in this book, we limit ourselves to the IT security aspect.

1.2.1 Why Are IT Systems So Insecure?

The more one is preoccupied with security, the more one may tend toward frustration or resignation: every program, every modern technical device—from network attached storage (NAS) hard drives to cars—seems to be full of bugs and security gaps. And that impression is unfortunately not misleading: there are various statistics on how many errors per 1,000 lines of code are common. If the number of errors drops to 0.5—that is, to *only* one error per 2,000 lines—this is already considered *stable code*. But operating systems like Windows, Linux, iOS, or Android consist of many millions of lines of code!

Why is the frequency of errors so high? Because programmers are human beings, and they make mistakes. Of course, the number of errors can be reduced through diligence, through reviews and test runs, but errors will always remain. (Fortunately, not every error is security-relevant, but hackers often manage to exploit even supposedly harmless mistakes.)

There is also the fact that not all software is developed under ideal conditions. The primary goal is usually to achieve a certain function in the first place ("first milestone"). This often takes longer than planned. Security checks and code protection are put on the back burner—and then not carried out at all due to time constraints.

Unfortunately, you can't tell by looking at a program or product how secure it is. Sales figures depend more on the packaging, functionality, and elegance of the user interface and marketing. When a security problem becomes known a year later, the company (if it still exists) has long been working on new products. From a purely economic point of view, fixing problems isn't worthwhile.

Another point is that even excellent programmers are not necessarily security experts. The IT world has long been too vast for one person to be the lead in every area. (It's no accident that this book was written by an entire team of authors).

Last but not least, fundamental errors happen even under ideal conditions: The development and implementation of new cryptographic methods is so complex that even teams of internationally recognized experts make mistakes. Such errors often lie dormant in the code for years until they are discovered—sometimes by accident. Then there's a fire in the house: frequently, millions of devices are affected that rely on standard libraries for TLS, HTTPS, WPA, or other common techniques of encryption or authentication.

IoT Is the Black Sheep

These problems are currently particularly serious in the case of IoT devices (see also Chapter 23). The webcam or LED lamp that can be controlled via Wi-Fi or Bluetooth yields such low profit margins that many companies dispense with serious security and long-term maintenance altogether.

Sometimes no update option is provided at all. From the outside, it isn't possible to see how an IoT device's security is technically implemented, so shifting the responsibility onto the buyer does not work. Here, only legislature has a chance to ensure more security through imposing clear regulations, including product liability laws for consequential damage.

The situation is only slightly better for Android smartphones (see Chapter 22). Most manufacturers provide updates for their devices for a maximum of one to one-and-a-half years, and often only for selected (expensive) devices. Currently, well over 90% of all such devices run outdated software versions that have known security problems. The fact that there has not yet been an outright security disaster in this segment (as of the end of 2022) is nothing short of a miracle.

1.2.2 Attack Vectors

The unwieldy term *attack vector* refers to a way a hacker can penetrate your company's or organization's computer system (see Figure 1.1).

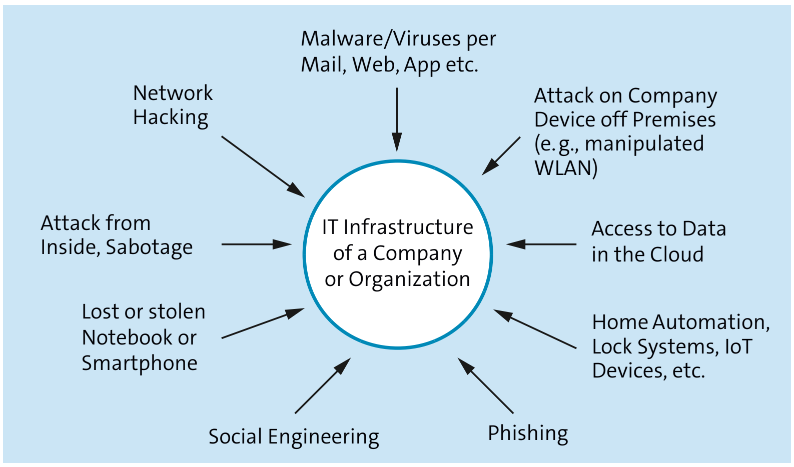

Malware/Viruses per Mail, Web, App etc.
Network Hacking
Attack on Company Device off Premises (e.g., manipulated WLAN)
Attack from Inside, Sabotage
IT Infrastructure of a Company or Organization
Access to Data in the Cloud
Lost or stolen Notebook or Smartphone
Home Automation, Lock Systems, IoT Devices, etc.
Social Engineering
Phishing

Figure 1.1 Popular Attack Vectors

The following sections summarize some common procedures and thus show the breadth of possibilities, but also the difficulty or impossibility of creating all-round security:

- **Network hacking**
  The term *network hacking* is often used to summarize "classic" ways of hacking. Using hacking tools that enable an attack via the network/internet, an attacker attempts to penetrate a company's or organization's computers and steal or manipulate data or otherwise cause damage. Configuration errors and unrepaired vulnerabilities are exploited. The target of the attack is usually not employees' computers, but servers.
- **Stolen/lost notebooks or smartphones**
  In whatever way a smartphone, tablet, or notebook falls into the hands of an attacker, it definitely contains a wealth of data that can be used for further attacks. This includes not only confidential files, but also, for example, access passwords for cloud and email accounts stored by the web browser.

  The question is whether an attacker can actually access this data. With modern smartphones, doing so is usually impossible without a password. For notebooks, it depends on whether the file system was encrypted (see Chapter 5). If not, accessing all its data will be a cinch for the thief.
- **Access to company devices on the go**
  Within a company, smartphones and notebooks are usually well protected, both physically and (thanks to firewalls and the like) within the company network. The situation is quite different when employees are not on their companies' premises with their devices, instead logging on to insecure Wi-Fi systems, using Bluetooth functions, and so on.

  Even without direct physical access to a device, attackers can attempt to exploit vulnerabilities in wireless protocols or their encryption techniques (see Chapter 8). Another variant is to trap a target person with a free Wi-Fi connection. By offering the Wi-Fi connection itself—often under a fictitious name, such as "free hotel Wi-Fi"—the attacker has various manipulation options at his or her disposal.

  Your company's employees can at least partially protect themselves against these types of attacks by avoiding unknown and unencrypted Wi-Fi connections, never entering passwords or other information on unencrypted websites, and always using virtual private networks (VPNs) to access the company network.
- **Infection of devices by malware, viruses, and the like**
  Another way into the target's computer is via emails, infected websites, or malware apps (especially for smartphones). For example, an email title may refer to an important company document that needs to be opened quickly, a website may promise iPhones or other products at half price, or an app may offer useful features that it can disguise malware behind. (A classic example is a flashlight app for turning on a phone's LED light. Depending on how you look at it, you can also regard some social media messengers as spyware.) In all cases, gullible users pave the way for the attacker. If the latest updates are missing on a user's device, an attacker is able to execute his own code via a vulnerability, and the disaster takes its course.

- **Tampered hardware**
  Of course, malware can also find its way into the computer via a USB stick or SD card. A USB stick, for example, could be attached to seemingly serious application documents or simply be lying on the street in front of the company building to arouse the curiosity of an employee.

  Moreover, the following applies: not everything that looks like a USB stick is one! An entire minicomputer can be inserted into a USB stick case, which, for example, can identify itself to the computer as a keyboard, simulate keystrokes immediately after insertion, open a PowerShell window within two or three seconds, download a malware program there, and start it. There are also purely destructive variants. These devices charge via the USB port for several seconds and then apply such a high voltage that the pulse destroys the USB port and, with a bit of bad luck, the entire computer.

  A company can only defend itself against such threats through targeted employee training: USB sticks and other electronic devices of questionable origin must *never* be plugged into a company computer!

- **Attacks on the cloud**
  Why should an attacker bother to penetrate an organization's network or devices if the data is in the cloud for free download anyway? In practice, it isn't always quite that bad, but in recent years there have been repeated cases of companies or organizations (in the fall of 2017, even the American secret service) leaving files unencrypted and without a login on cloud servers due to missing or incorrect configuration.

  Even if this worst-case scenario is not present, the cloud is a tempting target for attack. Hackers can try to attack the cloud as such. A more targeted approach is usually to try to steal the access data to the cloud or to use phishing to get the user to reveal the access data himself.

  Finally, it can be assumed that at least the US intelligence agencies have extensive access to all files stored in the clouds of the major US internet companies. If you haven't encrypted your files yourself, it can be assumed that the US intelligence service can evaluate them effortlessly.

- **Attacks on the network infrastructure**
  Since 2019, a battle has raged over which technology will be used to implement the next generation of mobile networks. Huawei seems to be the technological and price leader. This is a thorn in the side of the US, although it's difficult to determine whether there are not (also) economic interests behind this. The official line of argument is that China could build a backdoor into Huawei's hardware or software so that the 5G network could be the basis for espionage attacks by Chinese intelligence. (And this possibility does indeed exist; there is no denying that.)

  Largely lost in the discussion is the fact that even the established 3G and 4G networks are full of security gaps. It would be fascinating to know which states have built in or found (due to planning or implementation errors) monitoring capabilities here. One gets the impression that the US is so afraid of Huawei because its devices and software are completely beyond its control. This leads to the almost philosophical question of whether a state or its communications companies would prefer to be spied on by the US or by China: a rock or a hard place, then?

Of course, it's also clear that attacks on infrastructure rarely come from "small hackers" but rather from states or state-related organizations. In this respect, the infrastructure issue is far beyond the scope of this book.

- **Phishing**
  In *phishing*, an attacker tries to persuade the user to enter a password—for example, in an email that asks for verification of an online account and leads to a fake website. More details on phishing and password handling will follow in Section 1.4.
- **Social engineering**
  Hacking is a very technical discipline. It's easy to forget the human element. Why make a huge technical effort when with a little research and two phone calls you can find a password or get the down payment for a fictitious large order? Neither firewalls nor updates help against such attacks; clear work guidelines and regular training are needed instead.
- **Physical access**
  The simplest form of hacking is often underestimated: physical access to hardware. There are many forms of this, from the notebook forgotten (or stolen) on the train ride to the disgruntled employee taking the NAS device with all the backups from the open computer room.

  In this context, hardware hacking tools also play a major role. If a visitor plugs a minicomputer disguised as a USB stick into a desktop computer or plugs it into a free ethernet socket in an unnoticed moment, with a bit of bad luck this can go unnoticed for weeks or months. However, such devices often provide an attacker with far-reaching monitoring and manipulation functions (see also the Hacking Hardware box at the end of Section 1.1).
- **Attack from the inside**
  The larger the company, the more likely it is that there will be unhappy employees. What good are the best firewalls, VPNs, and so forth if an employee sabotages your computer system from the inside due to anger or frustration or for money? The greatest danger is when this employee is one of your administrators: then he or she often has almost unlimited access to all computers and data, knows all the security and backup procedures, and so on. As the Snowden case has shown, not even the National Security Agency (NSA) was prepared for this situation.

  There is no complete protection against an attack from the inside. But with some general measures, the risk can be mitigated at least a little. In general, employees should only have access to the data/computers/systems that they actually need for their work. When employees leave the company, all passwords, network access, and the like should be reset immediately, and company notebooks, cell phones, and other equipment should be confiscated as quickly as possible. Of course, a good working atmosphere doesn't hurt either!

1.2.3 Who Is Your Enemy?

"I have no enemies," you may think. Of course, it's nice when that's true in a private setting. However, as soon as you are responsible for the security of a company or organization, you need to rethink that. Consider the following possibilities:

- **Untargeted attacks by criminal hackers**
  Many hacking attacks do not have a specific target. Rather, some attackers are concerned with finding the easiest possible way to make money. This is true, for example, of most *crypto trojans*, which first encrypt a hard drive's files and then demand a "ransom" for the key to recover the data. At the same time, the attackers often extract personal data (records from a hospital, to name just one example) and threaten to publish it unless a ransom flows quickly.

  If you are affected by such an attack, it is not because someone has gone to the trouble of attacking you or your company personally. Rather, the attacker tries to find as many targets as possible. If one in a hundred of those affected pays, the income is already considerable.

  Annoyingly, the more companies or organizations (or their insurance companies!) choose to pay, the better this business model works. Even if this appears to be the most favorable way out for the company concerned under time pressure, the risk for the general public increases with each payment. The US Federal Trade Commission (FTC) estimates the damage caused by crypto scams at $680 million for the first quarter of 2022 alone. (The key word *crypto* here refers to both the encryption of the files and the common means of payment—namely, Bitcoin or other cryptocurrencies.)

- **Script kiddies**
  It's hardly likely that a script kiddie will explicitly target you or your company. Such attacks on a school's IT infrastructure are most likely to be carried out by young people to whom the dimension of such a "prank" is unclear.

  That being said, the role of script kiddies tends to be played up in the media. Most young people are also aware of the implications of hacking attacks. It may happen that major damage is caused by intentionally using or even just trying out a script, but such cases should be the exception.

- **Targeted espionage/sabotage**
  Much more real is the danger that your company becomes a target because an attacker steals company secrets or tries to damage your business through sabotage. By its very nature, this is especially true for companies that produce high-tech products—whether they are measuring devices, medications or modern consumer

products. But pure data is also valuable and therefore an attractive target; for example, the results of an elaborate scientific study or the script or film of a TV or movie series that has not yet been broadcast.

The possibility that the attacker is directly commissioned by a competing company cannot be ruled out in the international environment, but it's quite unlikely. Even a successful attack by a hacker who is not interested in the matter itself makes you or your company vulnerable to blackmail or can cause huge damage!

Some countries give the impression that they do not officially support organized hacking groups, but they do tolerate them—at least as long as they don't get caught.

- **Intelligence agencies**
  The task of intelligence services is to protect their respective states from attacks. The argument "I have nothing to hide anyway" may be true, but it still isn't desirable to have your internal company communication routinely read and your files stored in the cloud automatically evaluated.

  It's also unclear what role intelligence agencies play in corporate espionage. What is certain is that US intelligence agencies have extensive access to data stored in the cloud. However, it cannot be proven clearly whether or to what extent intelligence services—regardless of their nationality—also pass on (accidentally?) discovered knowledge to companies in their home country. With a little room for interpretation, one can easily argue that the economic success of companies from the vehicle and aircraft, mechanical engineering, or chemistry/pharmaceuticals industries ultimately serve the interests of the state.

  There are definitely suspicions in this direction. In this respect, companies can only be advised to also regard intelligence services as the "enemy" and, if possible, only store data that they themselves have encrypted in the cloud. Admittedly, this is more difficult than it sounds here (see also Chapter 20).

- **State-directed hacking, terrorist attacks**
  When you hear the term *cyberwarfare*, don't think of lurid motion pictures: this type of warfare has long been a reality, even if the parties involved won't admit it, of course. For example, the computer worm Stuxnet was developed specifically to sabotage uranium enrichment in Iran. The immense effort that went into this extremely focused attack, the deep inside know-how that was required, rules out "ordinary" hacker groups as the originators. In the Ukraine conflict too, there are many indications that hacking attacks carried out before or during the war cannot simply be attributed to local hackers, but were carried out or at least supported by state actors. Wikipedia lists a host of other incidents in which the hackers are suspected to have acted with state support (see *https://en.wikipedia.org/wiki/Cyberwarfare*).

  As of the publication of this book, successful hacking attacks by terrorist groups have not occurred (or have not become public knowledge). But it is to be feared that this too is only a matter of time. In addition to military facilities, power plants, waterworks, and other infrastructure facilities are considered to be particularly at risk.

The North Atlantic Treaty Organization (NATO) now considers cyberwarfare to be a central aspect of defense and has coordinated corresponding activities at the Cooperative Cyber Defence Centre of Excellence (CCDCOE) in Tallinn, Estonia, since 2008.

As we've mentioned before: we absolutely do not want to give the impression that you can use the know-how from this book to oppose the concentrated power of an intelligence agency. But many hackers, regardless of their background, act like burglars: they choose the targets that make it easiest for them to attack. That is why even basic security measures are sufficient to ward off at least untargeted attacks.

1.2.4 Intrusion Detection

Some types of hacking attacks do not remain hidden from the target for long: for example, when a computer reboots and demands an immediate Bitcoin payment to prevent encrypted files from being deleted, it's clear even to users without prior security knowledge that they have become the target of a hack.

However, many hackers, regardless of their background, are not interested in a few quickly earned Bitcoins. Rather, the longer-term analysis of the target may also be the real goal—for example, for corporate or government espionage or to explore even more worthwhile attack opportunities.

This is where the term *intrusion detection* comes into play (see also Chapter 16). It refers to the detection that a computer is (at least partially) under foreign control. At first glance, malware detection may sound trivial, but in fact it's extremely difficult. Malware is often only located in RAM—so a hard disk scan by an antivirus program therefore reveals nothing. The software is often tiny; a process hides behind innocuous names and hardly consumes any resources. Malware is most likely to be detected by particular behavior patterns or by conspicuous network packets. However, filtering out the mostly encrypted packets from the rest of the network traffic has similarities to the proverbial search for a needle in a haystack.

Different terms are commonly used for the time span from the intrusion to the detection of a hack, depending on the source, such as *detection time span* or *breach detection gap*. According to various statistics, the time span is in any case alarmingly high, often many months in length.

1.2.5 Forensics

IT forensics refers to the analysis of computers, smartphones, or other IT devices. Forensics is used for two main reasons:

- On the one hand, after a successful hacking attack, you usually want to find out who or what group gained access to the device and how. This root cause analysis helps to avoid similar mistakes in the future.
- On the other hand, after a fraud, robbery, or terrorist attack, the police, intelligence services, and so on naturally want to know who the perpetrator worked with and what other explosive data is hidden on the device. By their very nature, forensic analyses also play a major role in court proceedings.

Because the file systems of modern smartphones and notebooks are mostly encrypted (see also Chapter 5), forensic analyses also take into account the perpetrators' traces on the internet or in the cloud. An insight into forensic methods is provided in Chapter 7.

1.2.6 Ten Steps to Greater Safety

The following list doesn't replace the reading of the following chapters, but it can serve as an initial checklist:

- Keep the software on your devices up to date. Sort out all devices (e.g., smartphones, NAS devices, webcams, network printers, switches, or computers with Windows versions that are no longer maintained) for which there are no longer any updates or, if using such devices is unavoidable, operate them exclusively in separate networks.
- Regularly train your employees in how to use computers, smartphones, and other devices in a security-responsible manner. Point out current trends, such as social engineering, phishing, or malware attacks.
- All notebooks, smartphones, and tablets used by employees outside the company should have encrypted file systems (e.g., BitLocker on Windows).
- Just as Windows clients are normally administered centrally in the company, this should also apply to smartphones that are used for business purposes. We'll present the corresponding enterprise mobility management (EMM) tools in Chapter 22.
- Access to business-critical data from the outside, such as from your employees' notebooks, should only be possible via encrypted connections (HTTPS) or via a VPN. Check its functionality, or hire someone to set up a VPN.
- Perform initial basic security tests for your company's IT infrastructure, such as port scans for all computers, exploit scans, checking for trivial passwords, and so on. Suitable tools are provided free of charge by the Kali Linux distribution, for example.
- If you don't have sufficient hacking expertise, you should hire someone to perform an external penetration test. (This is a good idea even if you do have a competent security department. It's all too easy to become operationally blind!)
- In your considerations, you should also include external root servers, the cloud infrastructure used by your company, and the backup system.

- If you develop apps, web apps, or devices with integrated software yourself, include these products in your security considerations and controls as well.
- Don't forget things that are beyond the scope of this book, but still elementary. These include in-house organizational measures (emergency plans, clarifying responsibilities), the physical security of your IT infrastructure (e.g., Who has the key to the server room? Is the room really always locked?), and a legal assessment of your IT security or the consequences if something should go wrong.

The Time Factor

Hackers in motion pictures always work incredibly fast. Of course, this is part of the script, but the impression remains that hackers are generally omniscient and immediately know the right solution for every hurdle.

This is not true at all! Hackers, whether responsible or criminal, are IT specialists with an often pretty narrow focus. Anyone who regularly looks for security vulnerabilities in Microsoft networks or Active Directory isn't necessarily an expert on web servers running Linux.

Hacking takes time. If you're an administrator or security manager, the time factor works in your favor. The better the basic security measures, the more elaborate the attack—and the greater the chance that attackers will turn to another target.

1.2.7 Security Is Not Visible

To put it more succinctly: when Apple introduces a new iOS version, you can read on IT websites and in magazines about what new emoticons you can now enrich text messages with. These are features visible to end users. If, at the same time, an entire team of developers has implemented new security mechanisms with a huge expenditure of time and resources, this is usually not worth a single line of text to anyone. And even if the editor acknowledges the security effort, it's probably not possible to explain the new security mechanisms in three sentences in a way that readers will understand.

More generally, working for greater security is unrewarding. If all goes well, your efforts as an administrator or IT security officer will be taken for granted. At most, the top management will occasionally ask why you're needed: Everything's fine anyway, isn't it? So where's the benefit? You'll only be the center of attention when something has actually gone wrong. Then they all have always known it; it's only you who has recognized the obvious problem too late.

1.2.8 Security Is Inconvenient

As the person responsible for security, you shouldn't hope for praise from the user side. At most, criticism will come from there if the login process to the new VPN system is more cumbersome than before, if the two-factor login takes three seconds longer than the conventional login, if encrypting the hard drive makes the notebook a little slower, if security policies block the installation of the company's own apps on the company cell phone, and so on.

1.2.9 The Limits of This Book

Compared to other hacking or security books, which often focus on *one* aspect of hacking and corresponding countermeasures, we take a much broader approach in this book. We consider Windows *and* Linux, smartphones and IoT devices in addition to conventional computers and servers, present the risks of outsourcing data to the cloud in two chapters, address safe coding at least briefly in some chapters, and so on.

Nevertheless, a comprehensive and complete description of hacking and security in one book is impossible from the outset. Even 10 books of this caliber would not be enough. The following list briefly mentions some topics that are left out of or only briefly touched upon in our book:

- Security measures for client computers (antivirus programs, security settings, VPN, etc., including the consideration of macOS).
- Organizational measures (employee training, backup strategies and systems, logging and monitoring, contingency plans, certifications) and legal safeguards.
- Application-specific security, such as hacking procedures and security measures for specific programs or groups of programs. These include SAP and other standard business software; Oracle and other database servers; Joomla, TYPO3, and other web applications; Java EE and other software frameworks for implementing custom software solutions; and so on. This list could be continued almost endlessly.
- Security of mobile networks (GSM, UMTS, LTE, 5G, etc.).
- IT security in industrial plants, devices, and buildings (home automation, Industry 4.0, security of cars, airplanes, public transport, hospitals, etc.).
- Physical security (locking and fire protection systems, electronic key systems, etc.).
- Safe coding (selection of programming languages and test tools, safe programming techniques, etc.).
- Mathematical and IT-theoretical basics (e.g., cryptographic algorithms, hash methods, random numbers).

1.3 Exploits

An *exploit* is used to exploit a flaw (*vulnerability*) in a computer system in order to gain access to the system or its data, or even to a single (software) component, or to disrupt the operation of the system. Computer systems on which an attacker discovers a known exploit are wide open for them to attack. From the point of view of the person responsible for security in a company, it is important to eliminate known vulnerabilities as quickly as possible—usually by updating the affected program.

Vulnerability versus Exploit

The terms *vulnerability* and *exploit* are related, but not synonymous. A *vulnerability* refers to an error in a program. For example, if a program can be crashed by an incorrect input, then that's an error.

It only becomes an *exploit* if the vulnerability can be exploited in such a way that the affected program does not simply crash but does what the attacker wants—for example, discloses data or makes targeted changes. Hackers are incredibly creative when it comes to turning a seemingly harmless vulnerability into an exploit. This is also the reason that many people involved in IT security no longer differentiate between "common" errors (*bugs*) and security-relevant vulnerabilities: it's quite difficult to say in advance whether an error is only annoying or also security-relevant.

If a vulnerability or even an exploit based on it becomes known, the affected company or organization naturally tries to fix the problem as quickly as possible. This is often achieved within a few days, but sometimes it takes months. The commitment and agility of IT firms in the security space is quite variable but has consistently improved in recent years. For large companies in particular, it isn't desirable to be pilloried regularly in the IT press for a negligent update policy.

But fixing vulnerabilities unfortunately is only half the battle. The question now is how the corrected program gets onto the user's computer, onto the company's server, or into the affected device. The easiest way to do this is on conventional computers and on smartphones: here, there are established update mechanisms—and it's only up to the users or administrators to run these updates. This usually works well in tightly organized companies, but often less so in the private sector.

Safety Rule Number One: Import Updates

It actually seems unnecessary in a book for this target group, but for safety's sake we want to say it again: the regular, possibly even automated application of updates is indispensable! Don't be annoyed by a flood of updates; be happy when there are updates for your computers/devices.

Of course, every administrator knows the problem of defective updates: after applying an update, the computer no longer boots, the server no longer functions as before, the user interface behaves differently, and so on. Disgruntled employees and a lot of extra work are the result. Such problems are the exception, but they happen again and again. (Microsoft seems to have a particularly high incidence of such problems. At the same time, Windows in particular is known for preferring to start updates at the worst possible time, such as two minutes before the start of a presentation. But maybe these are subjective impressions.)

So the trouble of updates is an integral part of every administrator's work and a downside of any IT device application. But that's no excuse for not implementing updates!

1.3.1 Zero-Day Exploits

Zero-day exploits play a special role within the exploit world. These are exploits against which there are no patches, updates, or other defenses yet. Developers have no time (*zero days*) to fix the code, and administrators often have no meaningful ways to protect their systems against that.

Some exploits are zero-day exploits from the start—namely, when the underlying vulnerability was not previously known. In this respect, a crucial question is what happens when a vulnerability is discovered. Ideally, a responsible hacker or security researcher reports the error to the respective manufacturer. Only after the bug has been fixed and some time has passed for the delivery of the updates will the vulnerability and a possibly derived exploit be published (*responsible disclosure*), be it at a hacking conference, in a scientific paper, or simply as a security advisory on the internet.

In real life, it rarely works out that well for many reasons:

- Exploits are valuable (see the following section). The temptation is great to turn an exploit into money, either by criminally exploiting the flaw or by selling the know-how to someone else (possibly even an intelligence agency).
- Many reports of vulnerabilities never reach those responsible. There are many reasons for this, from sloppiness and disinterest in companies to the classification of the email in question as spam.
- Even if a vulnerability is known to a manufacturing company, the bug isn't always fixed immediately. Perhaps the software is no longer maintained at all, or the scope of the error is underestimated. Even responsible hackers lose patience at some point and, if they don't succumb to the temptations of the exploit black market, at least want to take credit for it. That's why it's common to publish vulnerabilities and exploits after a certain time (often after 90 days), regardless of whether the problem has been fixed in the meantime or not.

In a nutshell, the situation in which an exploit is public or at least known in hacker circles but not yet fixed by the manufacturer gives administrators a hard time. The often-heard advice to simply take the affected systems offline is rarely feasible.

The Dubious Role of Intelligence Services

We don't want to start a sociopolitical discussion about secret services here. The fact is that intelligence agencies also use exploits—for example, to intercept communications with suspected enemies of the state. This creates a dilemma: From the perspective of the intelligence agency, which may have paid a lot of money for the exploit, it is desirable to keep the exploit secret for as long as possible. From the company's point of view, on the other hand, a quick fix to the problem is important. It happens again and again that an exploit is "discovered" or passed on several times and then gets into the hands of not only the intelligence service, but also criminals. Thus, the exploit ultimately also endangers the state that the intelligence service is supposed to protect.

1.3.2 The Value of Exploits

It is probably clear to you now that exploits are valuable. There are several ways to make money with exploits:

- Many manufacturers have set up bug bounty programs and pay hackers and security researchers for qualified descriptions of vulnerabilities or even exploits (see also Chapter 19). This gives manufacturers the opportunity to fix the bugs directly without negative commentary in the media.
- On an international basis, there are regular hacking events (e.g., Pwn2Own) at which prize money is offered for particularly coveted exploits. Participants who manage to hack popular web browsers or operating systems acquire both reputation and money there.
- Finally, there is a (black) market for exploits, which is used by criminal organizers, but also by state organizations and intelligence services, among others. The best-known public site is *https://zerodium.com*. It boasts of paying six-figure dollar amounts for selected exploits. It's unclear which organizations are behind this site or to whom these exploits are later resold.

1.3.3 Exploit Types

Exploits can be classified according to various criteria. One aspect is the nature of the attack. A *local exploit* requires that the vulnerability be exploited directly on the target's machine. This can happen, for example, when opening a prepared file that was sent in an email. Even more attractive from a hacker's point of view is a *remote exploit*, in which it is sufficient, for example, to send special network packets to the computer that's being attacked. This is especially the case with most DoS exploits, in which "only" the operation is to be disrupted.

Another aspect is what kind of vulnerability an exploit is based on. The following list identifies some common types of vulnerabilities:

- **Memory access (buffer overflow, faulty pointers)**
  In these low-level vulnerabilities, a program writes to memory areas that are actually reserved for other purposes or reads memory areas that no longer contain valid

data. In both variants, a crash can occur in the best case; in the worst case, the attacker manages to execute targeted code that was previously injected.

- **Input validation**
  If a program doesn't sufficiently check the data or input to be processed, code can be injected into the program in this way, which the program then executes. This group of vulnerabilities includes, for example, SQL and HTML injections.
- **Race conditions**
  Some errors only happen when parts of the code are executed in parallel by multiple processors or cores in an unfavorable order for the program. An exploit is possible if the processing sequence can be influenced from the outside.
- **Privilege escalation/confusion**
  Here, due to an error, statements can be executed with higher rights than intended. This group includes *cross-site request forgeries*, in which a website that a user is logged into is exploited by an attacker who isn't logged in to execute their own commands (HTTP requests).

1.3.4 Finding Vulnerabilities and Exploits

Of course, it does happen that vulnerabilities are discovered by pure chance. However, it's much more common for the discovery to be preceded by a targeted search—by hackers who want to exploit the vulnerability or make money from it as part of a bug bounty program, but also by security researchers who, for example, want to point out hidden dangers in common algorithms or their sometimes sloppy implementation.

There are a number of techniques to systematically look for exploits. We'll present some tools in Chapter 17, Chapter 18, and Chapter 22.

1.3.5 Common Vulnerabilities and Exposures

Common Vulnerabilities and Exposures (CVE) is a standard for uniform naming of security vulnerabilities. CVE designations are intended to avoid the same vulnerability being named differently on security sites or by security products.

New vulnerabilities can be reported at *https://cve.mitre.org*. If the report is followed up on by the CVE team, the vulnerability will be given a unique name that starts with "CVE-" and the year. After that follows a number of at least four digits. For example, CVE-2015-1328 refers to a flaw discovered in 2015 in the `overlayfs` module of the Linux kernel.

1.3.6 Common Vulnerability Scoring System

The CVE entry doesn't care about the possible impact (exploits) of a vulnerability and doesn't contain a corresponding rating. In practice, however, a benchmark is needed to separate comparatively harmless vulnerabilities from harmful ones. The Common Vulnerability Scoring System (CVSS) has been established for this purpose. The benchmark was originally developed by the National Infrastructure Advisory Council (NIAC), a working group of the US Department of the Interior. Since 2005, the Forum of Incident Response and Security Teams (FIRST) has been responsible for the continued development of this standard.

The current CVSS version 3.0 has been in effect since June 2015, but values according to version 2.0 are still common. CVSS values are between 0 and 10 (Table 1.1). Various factors go into calculating this value, including what authentication is required, what the impact is, and how complex the application of an exploit is. Note that the same vulnerability may have different CVSS values in versions 2.0 and 3.0 due to different calculation methods.

Value Range	Severity
0	none
0.1% to 3.9%	low
4.0% to 6.9%	medium
7.0% to 8.9%	high
9.0% to 10.0%	critical

Table 1.1 Severity Levels for CVSS Values, Version 3.0

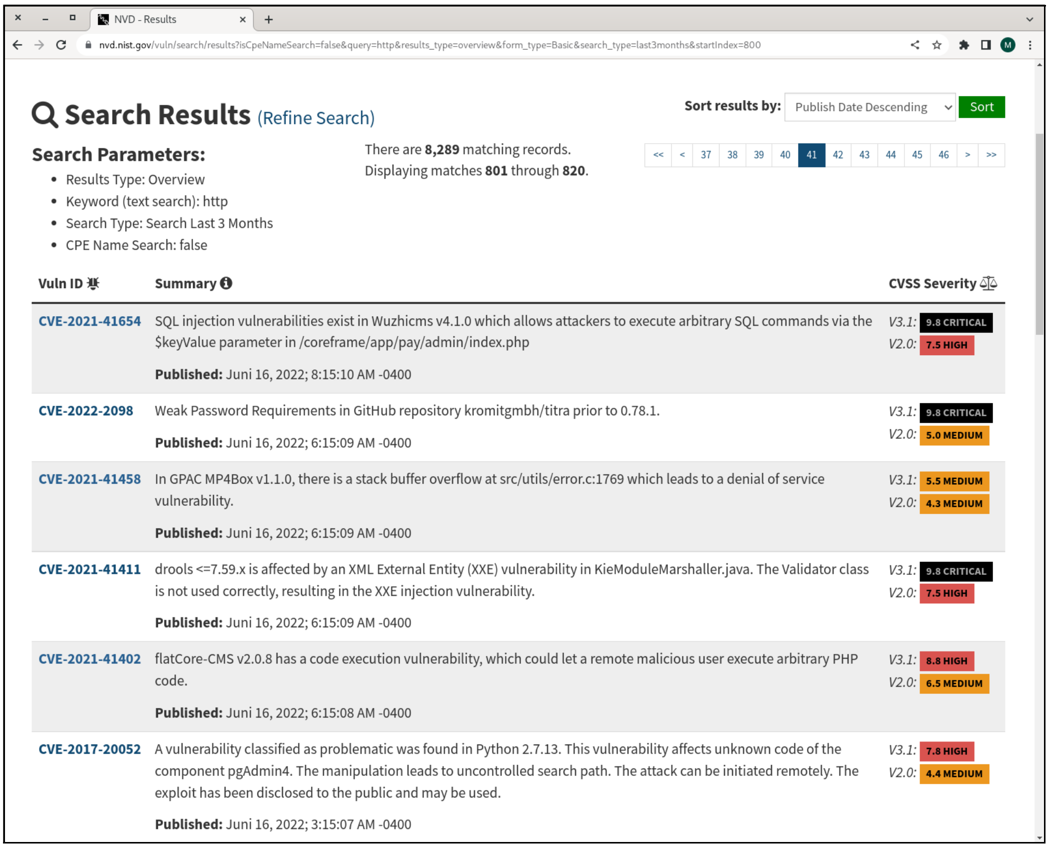
1.3.7 Vulnerability and Exploit Databases

There are various databases on the internet that collect important vulnerabilities (see Figure 1.2). Almost without exception, the CVE number and CVSS rating are given, as well as various additional information such as links to temporary bug fixes and permanent updates, but also to exploits.

Examples of such databases include the following:

- **National Vulnerability Database**
  The *https://nvd.nist.gov/vuln* website of the National Vulnerability Database (NVD) is maintained by the National Institute of Standards and Technology (NIST). NIST, in turn, is under the control of the US Department of Commerce.
- **Exploit Database by Offensive Security**
  The Exploit Database is maintained by Offensive Security (*https://www.exploit-db.com*).
- **Vulnerability & Exploit Database (Rapid7)**
  The Vulnerability & Exploit Database is operated by Rapid 7 (*https://www.rapid7.com/db*).

- **CVE Details**
  The CVE Details site at *https://www.cvedetails.com* contains automatically generated information from other sources. At first glance, this doesn't sound particularly imaginative, but the advantages of the site lie in its clear presentation and good search and filter options.


NVD - Results

nvd.nist.gov/vuln/search/results?isCpeNameSearch=false&query=http&results_type=overview&form_type=Basic&search_type=last3months&startIndex=800

Search Results (Refine Search)

Sort results by: Publish Date Descending | Sort

Search Parameters:
- Results Type: Overview
- Keyword (text search): http
- Search Type: Search Last 3 Months
- CPE Name Search: false

There are **8,289** matching records.
Displaying matches **801** through **820**.

<< < 37 38 39 40 41 42 43 44 45 46 > >>

Vuln ID	Summary	CVSS Severity
CVE-2021-41654	SQL injection vulnerabilities exist in Wuzhicms v4.1.0 which allows attackers to execute arbitrary SQL commands via the $keyValue parameter in /coreframe/app/pay/admin/index.php **Published:** Juni 16, 2022; 8:15:10 AM -0400	V3.1: 9.8 CRITICAL V2.0: 7.5 HIGH
CVE-2022-2098	Weak Password Requirements in GitHub repository kromitgmbh/titra prior to 0.78.1. **Published:** Juni 16, 2022; 6:15:09 AM -0400	V3.1: 9.8 CRITICAL V2.0: 5.0 MEDIUM
CVE-2021-41458	In GPAC MP4Box v1.1.0, there is a stack buffer overflow at src/utils/error.c:1769 which leads to a denial of service vulnerability. **Published:** Juni 16, 2022; 6:15:09 AM -0400	V3.1: 5.5 MEDIUM V2.0: 4.3 MEDIUM
CVE-2021-41411	drools <=7.59.x is affected by an XML External Entity (XXE) vulnerability in KieModuleMarshaller.java. The Validator class is not used correctly, resulting in the XXE injection vulnerability. **Published:** Juni 16, 2022; 6:15:09 AM -0400	V3.1: 9.8 CRITICAL V2.0: 7.5 HIGH
CVE-2021-41402	flatCore-CMS v2.0.8 has a code execution vulnerability, which could let a remote malicious user execute arbitrary PHP code. **Published:** Juni 16, 2022; 6:15:08 AM -0400	V3.1: 8.8 HIGH V2.0: 6.5 MEDIUM
CVE-2017-20052	A vulnerability classified as problematic was found in Python 2.7.13. This vulnerability affects unknown code of the component pgAdmin4. The manipulation leads to uncontrolled search path. The attack can be initiated remotely. The exploit has been disclosed to the public and may be used. **Published:** Juni 16, 2022; 3:15:07 AM -0400	V3.1: 7.8 HIGH V2.0: 4.4 MEDIUM

Figure 1.2 Search Results for HTTPD Vulnerabilities on NVD Website

There used to be yet another important database, the Open Source Vulnerability Database (OSVDB). However, this project of the Open Security Foundation (OSF) was discontinued in April 2016 because the cooperation required with other IT security companies and organizations did not work out.

1.3.8 Vulnerability Scanner

A *vulnerability scanner* is a program that examines computers accessible on the network and tries to identify which known security issues the programs are vulnerable to. The largest of these programs is OpenVAS (see Chapter 4, Section 4.8).

By their very nature, vulnerability scanners are useful to both sides: the attacker looking for easy prey, and the defender or pen tester looking for gaps in the computing landscape being managed.

1.3.9 Exploit Collections

From the vulnerability and exploit database, it's only a small step to exploit collections such as Metasploit (see Chapter 4, Section 4.9). Such programs make the application of exploits particularly easy: basically, you just need to select the exploit you want to use, specify the IP address of the computer you want to attack and some other parameters, and launch the exploit.

You can also specify what should happen in case of a successful attack. The *payload* is code that is to be executed on the attacked machine. Often, it's a rootkit, a virtual network computing (VNC) server, or other hacking tools, such as the Metasploit-specific program Meterpreter.

Rootkits

A *rootkit* is a program that is executed inconspicuously or possibly even permanently installed after a successful attack and subsequently allows controlling the computer. Rootkits communicate with the attacker via a network connection. Various measures are taken to ensure that the rootkit leaves as few telltale traces as possible (no files of its own, no manipulated logging entries, etc.).

1.4 Authentication and Passwords

The beginning of all security evils is all too often passwords. Countless IT security problems have to do with the authentication of users as well as with the handling of passwords. In the past, errors happened at all levels (see also Chapter 6): in the authentication algorithms, in the storage of passwords or hash codes derived from them, and in the handling of passwords by the user.

Authentication versus Authorization

The terms authentication and authorization are easily confused:

- *Authentication* refers to the login process—that is, the process of determining whether someone is who they say they are. In the simplest case, this control is via a password, but there are of course many other methods. (Depending on the point of view: A user *authenticates* with a web server. The server *authenticates* the user who is currently logged in. We look at the process in this book mostly from a computer point of view, so we talk or write about authentication.)

- *Authorization* determines who can do what and with what rights. In operating and database systems, there are often different roles (administrators, ordinary users) with different rights.

1.4.1 Password Rules

Many organizations or companies have strict password rules. These not only prescribe the minimum password length (e.g., 8 or 10 characters), but also enforce the use of special characters and numbers, as well as regular schedules for changing passwords (often every six months, and the new password must not be too similar to the old one). Users are also instructed to use different passwords for each account so that a compromised password cannot compromise all other accounts as well.

On the one hand, these rules can be well justified. Time and again, research shows that the most popular passwords are “hello”, “secret”, “123456”, or birth dates when users are given a free choice. It’s precisely such passwords that automated password cracking tools check first, so they’re completely insecure.

On the other hand, these rules lead directly to the next dilemma: no human can remember many complex, constantly changing passwords. Therefore, passwords are written down on slips of paper, in text files, or in Excel spreadsheets. Alternatively, password managers integrated into a web browser or standalone password managers are used.

All these “solutions” are, of course, absolutely undesirable in terms of security. In the worst-case scenario, not just one password but hundreds or thousands of passwords could be compromised at once—for example, if an attacker manages to access the password database of the web browser. In fact, in the past, procedures to read login data stored in the web browser, including passwords, were regularly disclosed.

In addition, countless websites offer a forgotten password function. This will send a new password or a link to reset the password to a previously set email address. If an attacker manages to take control of the email account, they can get countless passwords delivered to their doorstep.

1.4.2 Phishing

According to research by Google, phishing is the most common cause of accounts being taken over with passwords that are actually secure—that is, not *123456* or similar. The goal is to persuade the user to enter a password. There are many ways to do this:

- An email indicates a problem with an online service (eBay, Amazon, etc.). The recipient will be asked to verify their account. Of course, the corresponding link doesn’t reference the real page, but a deceptively real-looking duplicate, the address of which may differ from the original by only one letter. If the recipient enters his or her account data there, it ends up directly in the hands of the attacker.

- A harmless-looking malware app (e.g., something that's ostensibly a game) pops up a dialog that looks exactly like the dialog for entering an iCloud password or any other password of a smartphone provider or cloud provider. The target thinks the input is necessary to install an update or plug-in—and reveals the password.
- In a public place (train station, airport, hotel), the target uses a free but unencrypted Wi-Fi service. The name of the network (e.g., "Free Hotel Wi-Fi") doesn't suggest that the access point is actually operated by a hacker. All data that is then transmitted unencrypted via this Wi-Fi connection can be read directly by the attacker. Moreover, the attacker can redirect popular pages, such as from webmail providers, to their own servers. If the target is negligent enough and ignores any certificate warnings, he or she logs into the attacker's fake site and reveals the password.
- *Social engineering* is also a popular means of phishing. For example, the target is ostensibly called by Microsoft or another large IT company. The caller explains any licensing issues or other discrepancies that need to be resolved. But this resolution requires the user's password!

Users who use the same password or very similar passwords for multiple accounts are a particularly attractive phishing target: they can allow the attacker to take over several accounts at once.

Anyone who has been involved in IT security for a long time will not fall for phishing tricks. But not everyone is a security expert. The scammers act very cleverly and always come up with new ideas. Especially on smartphones, where only the name and not the actual email address is displayed in emails and where the web browser also doesn't display the full address, it is almost impossible to distinguish legitimate emails or websites from phishing emails or websites.

An alternative to phishing is software-based or hardware-based *key loggers*, devices or programs that log everything that's typed on a keyboard. However, installing a corresponding program requires that the computer has been infected with a rootkit or malware. It's even more complex to set up a hardware key logger: to do this, the attacker needs physical access to a computer or at least to the room where the computer is located.

1.4.3 Storage of Passwords (Hash Codes)

Passwords must never be stored in plain text in databases. If the database falls into the hands of an attacker, they would have direct access to all stored passwords.

That is why it is common to store hash codes of the passwords instead (see Chapter 6). Thanks to sophisticated mathematical algorithms, hash codes allow the verification of a given password, but not the reconstruction of a password that is unknown. Modern hash algorithms also include a random component in the hash code so that each hash code (even for matching passwords) is unique. This makes it impossible for hackers to generate in advance a huge database containing the corresponding hash code for popular passwords.

Nevertheless, hash codes also pose security risks:

- In recent years, it has happened time and again that hash codes have been generated using outdated procedures or faulty algorithms. In such cases, the mass reconstruction of passwords from the hash codes was virtually a walk in the park for the attackers.
- Even when used correctly, with suitable hardware one can try and verify (many) thousands of passwords using available hash codes. Thus, an attacker who gets hold of a database of hash codes can, with enough patience and computing power, figure out passwords by guessing.

Pass-the-Hash/Pass-the-Ticket

Some systems internally assign users a hash code (a *ticket*) as an authentication token at login. If an attacker manages to get hold of this hash code (this usually requires that they already have access to the affected computer), they can use it to extend their rights (in a *pass-the-hash* or *pass-the-ticket attack*). A popular hacker tool for reading hash codes from memory is `mimikatz` (see also Chapter 13, Section 13.6).

1.4.4 Alternatives to Passwords

Many IT researchers seek to replace passwords with better authentication methods. Unfortunately, there is no ideal alternative:

- **Biometric procedures**
  Biometric procedures such as fingerprint, face, and retina scanners are convenient and comparatively reliable. However, security experts view these procedures with skepticism: a compromised password can be replaced with a new one, but biometric data is unique and cannot be replaced. In addition, the broad collection of biometric data causes surveillance nightmares.
- **Hardware authentication**
  Authentication by a chip, magnetic or RFID card, or similar device is ubiquitous for access control in companies, hotels, and other buildings. However, the application of these devices on computers fails due to the lack of distribution of reading devices. One variant is USB plugs, but they haven't been able to establish themselves so far either. In addition, many hardware-based methods have themselves shown massive security issues in the past, so in some cases they create more problems than they solve.

- **Two-factor authentication**

  Two-factor authentication (2FA) is performed by two components, such as a conventional password plus a one-time code that is sent to a smartphone or generated there as needed (see, for example, Chapter 14, Section 14.4). Biometric or hardware-based methods can also be used as a second factor.

  For the user, 2FA currently has the disadvantage of being much more cumbersome. That's why some vendors use 2FA only on a case-by-case basis, such as when logging in for the first time after a long period of time. Of course, this reduces the security gain.

  Another problem with 2FA is that there must be a way out in case the second factor (e.g., a USB token like YubiKey) is lost. On the one hand, the fear of losing the second factor is a major reason that 2FA is so reluctantly used in real life; on the other hand, storing replacement 2FA codes or procedures for reauthorization again poses new security risks.

  Fast Identity Online (FIDO) is being used to try to fix many issues related to 2FA (see the following section).

1.4.5 Fast Identity Online

FIDO could be the fourth item in the preceding list of password alternatives. But FIDO is too important for that! This procedure deserves its own section.

Behind the FIDO authentication process is an alliance of the same name made up of several companies, originally including Facebook, Google, and Amazon. Its goal is to establish a secure and license-free standard for authentication on the internet. The basic idea is simple: authentication uses not a password, but a cryptographic key. This is located directly in the device, on an attached USB stick, or on a Bluetooth device. This makes every login as secure as the authentication of the device. In other words: anyone who can unlock their smartphone can subsequently use all FIDO-compatible websites without having to enter additional passwords.

A so-called crypto chip guarantees secure key matching. For this purpose, the FIDO2 device generates an individual key pair consisting of a private and a public key for each server to be logged into. The public key is stored on the server and used for further log-ins. This gives each login option an individual key. The private key remains unreadable in the crypto chip; the public key stored on the website or other device is used for verification.

Many notebooks and smartphones are FIDO-compatible out of the box. Often a CPU or an additional processor contains the required crypto functions. For older notebooks or PCs, a FIDO USB stick (costing approx. $30) can help.

Of course, the client hardware is only half the battle. The website where the registration is to take place must also play along and support FIDO. As of the spring of 2022, this was only exceptionally rarely the case.

If both sides are FIDO-compatible, a login on the user side in the minimal variant requires only a confirmation of physical presence. On notebooks and smartphones, FIDO is usually linked to biometric procedures. In this case, FIDO authentication starts only after facial recognition or touching the finger sensor. In some cases, however, FIDO is “only” used as a second factor for a 2FA.

FIDO has undisputed advantages. Authentication itself is secure according to the current state of the art and extremely convenient for the user as he or she no longer has to deal with passwords. Because the key is on a physical device (smartphone, notebook, USB stick), the phishing attacks that are currently so popular invariably come to nothing. There is no risk of the key being stolen by malicious software.

While IT heavyweights Microsoft and Google have supported FIDO for some time, Apple only got its act together in 2020. With iOS version 16 or macOS version 13, most current Apple devices are FIDO-compatible. However, Apple does not call the process FIDO, but Passkey. In contrast to other implementations, Apple provides for cloud synchronization of the keys across multiple devices, whereby the keys cannot be read by Apple (end-to-end encryption). Google has announced that it will also implement a comparable solution in the future. Microsoft’s position was unclear at last check. Although the company has been one of the most active proponents of FIDO from the beginning, it hasn’t yet presented a coherent strategy for the secure exchange of keys across multiple devices or programs.

With the support of Apple, Google, and Microsoft, nothing stands in the way of establishing the standard worldwide. It is to be hoped that this will finally also massively increase the number of websites that accept FIDO as an authentication method. So far, the number of FIDO-compatible websites is still very limited. What good is the best technology if it isn’t used across the board?

1.5 Security Risk IPv6

When the internet was conceived, a central idea was that direct connections could be established among all devices on the network. This concept failed because the address space of IPv4 proved to be too small. It was impossible to give each device a unique, globally valid IP address.

Since then, countless local networks of companies, organizations, and private households are not located in the public part of the IPv4 network, but in private networks with address spaces in 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8, respectively. For devices on private networks to communicate with the internet, all IP packets must be manipulated as they leave the private network. The network address translation (NAT) required for this is usually performed by a router or on a gateway computer.

For network purists, this procedure is a nightmare. Special adjustments have to be made for a lot of protocols. Some network services still suffer from the complications and delays caused by NAT today.

The obvious solution to all these problems is IPv6. This “new” internet protocol—which is in fact more than 20 years old—has solved the problem of the address space being too small. However, it hasn’t really caught on so far. Anyone who always uses IPv6 also needs IPv4 access (e.g., in the form of a so-called tunnel); otherwise, large parts of the internet are currently not usable at all.

1.5.1 Security Complications

From the perspective of IT security, the IPv6 switchover is by no means as desirable as it is from the perspective of network theorists. Thanks to NAT, every computer on a private network is protected against direct attacks from the internet (see Figure 1.3). For example, a Windows directory shared across the network is visible on the local network, but not beyond that boundary.

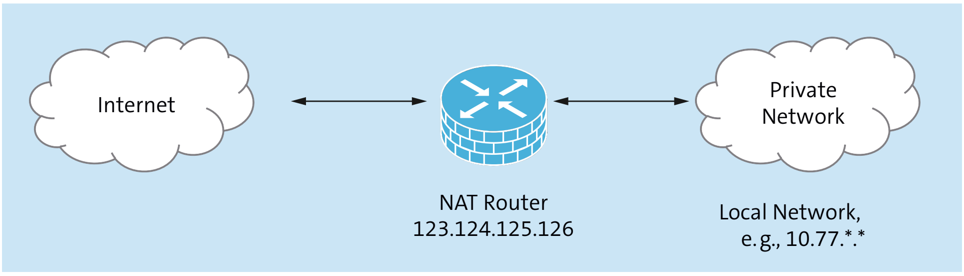


Figure 1.3 Via NAT, Company Computers in Private IPv4 Networks Can Use the Internet but Are Difficult to Attack Directly from It

As soon as all computers in an organization receive IPv6 access in addition to IPv4, a direct attack on every single company computer is possible via IPv6. An attacker who has IPv6 can connect directly to any company computer, provided he or she knows or guesses its IP address. Naturally, this problem can be fixed by firewalls, but the protection automatism then drops out.

Another disadvantage of IPv6 is an (additional) loss of privacy. While internet traffic can currently be assigned to a specific company, but not easily to a specific device from that company’s private network, unique IPv6 addresses allow unambiguous identification and recognition. This problem can also be solved, especially by using regularly changing IPv6 addresses (*IPv6 privacy extensions*).

Finally, many firewalls and other network defense tools are based on counting the number of DoS attacks and login attempts originating from an IP address (e.g., the program Fail2ban; see Chapter 14, Section 14.7): after a certain number of failed logins, the IP address is simply blocked for a while. IPv6 makes it easy for attackers to hide behind an almost infinite number of addresses.

From this point of view, many companies are currently of the opinion that an internal introduction of IPv6 has few advantages, but is at least problematic from a security point of view. However, as a security manager, you cannot avoid the topic quite so easily:

- As soon as employees of your company use a smartphone or are on the road with a notebook, it may very well happen that they use IPv6 networks—often without knowing it. You must take this into account in client configuration, either by generally blocking IPv6 for certain services or by making appropriate firewall settings.
- Even root servers that are operated outside the company, such as in a data center, usually have an IPv6 connection. Here too it must be questioned for each server service whether it must or should support IPv6. However, a general deactivation of IPv6 is not expedient in this case! Especially for web and mail servers, it's absolutely desirable that they are reachable via IPv6.

1.6 Legal Framework

Let's note one thing right away: none of the authors involved in this book have legal expertise. We are all computer technicians with various specialties, but law is not one of them. However, a few general statements can be made.

1.6.1 Unauthorized Hacking Is Punishable by Law

Although the exact wording varies, hacking without permission is a criminal offense in most countries. For example, in Germany, the so-called Hacker Article, Section 202c of the German Criminal Code, applies.

Section 202c: Preparing to Spy on and Intercept Data

"Any person who prepares an offense under Section 202a or Section 202b by producing, obtaining for himself or another, selling, giving to another, distributing or otherwise making accessible passwords or other security codes that enable access to data (Section 202a(2)) or computer programs whose purpose is the commission of such an offense shall be punished by imprisonment for not more than two years or a fine. (2) Section 149 (2) and (3) shall apply mutatis mutandis."

In this context, Section 202a or Section 202b deal with further aspects of IT security, namely the spying on and interception of data. Section 149 deals with the counterfeiting of currency and stamps.

In the Austrian Criminal Code, there are comparable formulations in Section 118a and Section 126a to c. Similarly, you can read Articles 143 and 144 in the Swiss Penal Code.

A simple port scan (see Chapter 4, Section 4.1) can therefore already be considered preparation for a criminal offense under Section 202c. At first glance, this seems absurd: such scans are ubiquitous, and there is no reasonable means against them. If your company's security system or firewall detects such a scan and you can trace the underlying IP address back to Ukraine, for example, then what do you want to do as the company's security manager?

Of course, you can try to find out who owns the IP address or from which internet provider the scan originated. Even if you succeed, you may well end up only encountering computers that are themselves compromised and remotely controlled by the attacker from a completely different location. So in a nutshell: even if you know that other hackers from abroad perform port scans incessantly, you still must not start a port scan on someone else's computer yourself.

Although the law does not state it explicitly and does not differentiate between responsible and criminal hackers, "goodwill" use of hacking tools, such as in the context of a pen test, is usually accepted. Nevertheless, it should be clear to you that the use of hacking programs outside of test systems absolutely requires written permission!

Also keep in mind that hacking often crosses national borders: even if a company is headquartered in Germany, for example, it may have servers that are located in Ireland or the US. This makes the legal assessment even more complicated.

1.6.2 Negligent Handling of IT Security Is Also a Criminal Offense

It's not just unauthorized hacking that can get you into hot water. Neglecting your company's security is also increasingly becoming a problem. In doing so, it's better not to look to the past, when even monumental data leaks went unpunished or resulted in only comparatively small fines.

In the meantime, both public perception and the range of punishment have changed drastically: in 2019, Facebook was fined $5 billion in the US for sharing its members' personal data too carelessly with third-party companies. In the UK, a further fine of £500,000 was added for the same offense—with a note that the penalty would have been significantly higher had the General Data Protection Regulation (GDPR) already applied at the time of the data transfer (we'll get to that in a moment).

In Germany, the Federal Data Protection Act (BDSG) formulates the rules to which companies that manage and store personal data must adhere. From the perspective of this book, the safety and protection requirements formulated in Section 9 of the act are particularly relevant. In addition to more general security measures (physical protection including fire protection measures, password checks, backups, etc.), it's stipulated there, among other things, that the transmission of data must be encrypted, and that this must be done in accordance with the current technological state of the art. In the case of serious violations, fines and imprisonment are provided for. This also applies, for example, in the event that a hacker was able to steal and publish data from your company because your protective measures were not state of the art.

In Austria, the handling of personal data is regulated by the Data Protection Act of 2000. In particular, Section 14 requires a company or organization to take appropriate data security measures.

1.6.3 European General Data Protection Regulation

Detached from national laws, the GDPR has applied to all EU countries since May 2018. Rules concerning the processing of personal data are laid down there. The provisions will be integrated into the corresponding national laws and replace or supplement the previous provisions.

The aim of GDPR is to achieve uniform standards within the EU. For many countries, this is accompanied by a tightening of the provision and a much higher range of penalties. Fines of up to 4% of the company's worldwide turnover are possible! After an initial grace period, there have now been a number of proceedings, some of which have resulted in severe penalties for the companies responsible.

In the United Kingdom, the GDPR also applied until Brexit. Since then, there have been transitional rules. The EU has issued an adequacy decision in which the UK is considered a safe third country (in terms of data protection). This is particularly important for cross-border cloud solutions.

1.6.4 Critical Infrastructure, Banks

A special case is the area of critical infrastructures which includes energy and water supply, healthcare and finance, and telecommunications. In accordance with the guidelines of the European Programme for Critical Infrastructure Protection (EPCIP), for example, the Act to Increase the Security of Information Technology Systems (IT Security Act) was passed in Germany in mid-2015.

In addition to an obligation to implement comprehensive security measures, these laws also contain significant threats of punishment and the obligation to report security incidents to a government reporting office.

Stricter security rules also apply to banks and financial service providers. This includes the obligation to have a security officer who is independent of day-to-day operations and, in particular, the regular IT security department to monitor IT security and provide a report on it on a quarterly basis.

1.6.5 Security Guidelines and Standards

European standards stop at nothing, not even IT security. Worth mentioning in the context of this book in particular are the international standards ISO/IEC 27001 and 27002:

- **ISO 27001**
  The Information Security, Cybersecurity, and Privacy Protection: Information Security Management Systems: Requirements standard defines guidelines for the establishment and operation of a documented information security management system.
- **ISO 27002**
  This international standard contains recommendations for control mechanisms for information security.

Unfortunately, the full text of the standards is only available for a fee (see *https://www.iso.org*). But you can find brief summaries of the standard on Wikipedia. See *http://s-prs.co/v569601* and *http://s-prs.co/v569602*.

1.7 Security Organizations and Government Institutions

Both internationally and in the English-speaking world, there are countless governmental and public organizations and institutions that deal with IT security. The following list is provided by way of example and makes no claim to be exhaustive. We are also concerned here with being able to use abbreviations such as BSI or CERT in the further course of the book without having to explain their meaning each time:

- **BSI**
  The Federal Office for Security in Information Technology is a federal agency of the German Ministry of the Interior. Its tasks include protecting the federal government's IT systems, defending against cyberattacks, testing and certifying IT products and services, and developing IT security standards. With over 1,500 employees, it's probably the largest such institution in the German-speaking world.
- **NIST**
  The National Institute of Standards and Technology is part of the US Department of Commerce and is responsible for standardization processes.

NIST is relevant to IT security in that it's also responsible for the standardization of encryption protocols, for example. In addition, NIST maintains the National Vulnerability Database described earlier in this chapter.

- **CERTs and FIRST**
  Computer Emergency Response Teams (CERTs) are groups of IT security professionals who help each other resolve IT security incidents and deal with IT security. The US CERT group is best known for its regular publications of alerts (formerly CERT Advisories, now CERT Alerts): *https://www.first.org/*, *https://www.govcert.admin.ch*, *https://www.switch.ch/security*, and *https://www.cisa.gov/*.

  The global umbrella organization for all CERTs is called the Forum of Incident Response and Security Teams (FIRST). FIRST maintains the Common Vulnerability Scoring System for evaluating security vulnerabilities.
- **CCC**
  The Chaos Computer Club is by no means as chaotic as its name implies. This is a group of hackers that has been established for over 30 years and who exchange views on hacking successes and inadequate IT security in regular meetings and lectures (often definitely with a sociopolitical flavor).

In addition, of course, there are countless other companies, organizations, associations, and universities and other educational institutions that deal with IT security.

Chapter 2
Kali Linux

Kali Linux is, in a way, the Swiss Army knife of every security and hacking professional. It's a Linux distribution that combines a seemingly endless collection of hacking tools. Of course, you can install most of the tools in other Linux distributions as well. There are even Windows versions of some of the hacking tools. But Kali Linux has the advantage that the distribution makes many important commands for penetration testing and related tasks conveniently accessible via a central menu. There is no need to spend time searching for the commands, installing them, or, if necessary, compiling them yourself.

From the point of view of Linux professionals, Kali Linux is simply one of the countless Linux distributions based on Debian. The installation and operation (apart from the use of the hacking tools) are therefore the same as for other Debian variants.

If you're primarily at home in the Windows or macOS worlds, then you may not yet be familiar with Linux-specific details. Although this book is not the appropriate place for a full Linux introduction, we have tried to give some consideration to readers who are not that Linux-savvy in this chapter.

The chapter describes how you can first try Kali Linux and then install it on a virtual machine. We cover a relatively large number of installation variants and include VirtualBox, Hyper-V, WSL2, and the Raspberry Pi, among others. As a kind of preview for the chapters that follow and in which we will describe many hacking tools in detail, two short examples will demonstrate the practical use of Kali Linux.

2.1 Kali Alternatives

Kali Linux has established itself as the most popular Linux distribution for pen testers, hackers, and security experts, but it is by no means the only option:

- Parrot OS (*https://www.parrotsec.org*) is one of the most interesting alternatives. Like Kali Linux, this increasingly popular distribution is based on Debian and provides a similarly wide range of hacking tools to choose from. Parrot OS stands out from the competition with its colorful desktop layout.
- BlackArch (*http://blackarch.org*) is aimed specifically at Linux experts. For beginners, this distribution based on Arch Linux is not suitable.

- For those who prefer to work on Windows and also want to do without virtual machines, PentestBox (*https://pentestbox.org*) offered a good set of many Windows-compatible tools. Unfortunately, the latest version is from 2018, which is why we no longer recommend using PentestBox.

2.2 Trying Out Kali Linux without Installation

On Windows and Linux, you can try Kali Linux without installing it. This is of course a particularly attractive option for getting to know the distribution. You can choose from a variety of download options at *https://kali.org/get-kali* that might be confusing at first glance.

To try it out, you must choose the **Live Boot • 64-bit • Point release live image** variant. The image file was about 3.5 GB when we last checked it. After verifying the authenticity of the download (details will follow shortly), there are two procedures to choose from:

- You can transfer the ISO file to a USB stick. To do this, you must use a special program that transfers the file block by block. The easiest way to do this is to use the free Etcher program (*https://balena.io/etcher*), which runs on all common operating systems. Linux professionals can also use the `dd` command to copy the image. Once the USB stick is prepared, you must reboot your notebook and use the USB stick as boot media. (You may need to change your BIOS/EFI settings beforehand to make the notebook accept external boot media.)

  This variant is especially perfect if you want to use Kali Linux to read data on a foreign device whose password you do not know (see Chapter 5). Note, however, that Kali Linux is not UEFI Secure Boot compatible. You must disable this protection mechanism before you can start Kali Linux (see also Chapter 5, Section 5.1).
- Alternatively, you can try Kali Linux in a virtual machine, which is the option we'll focus on ahead.

2.2.1 Verifying the Download

Kali Linux consists mainly of open-source software. Therefore, with the appropriate knowledge, it's possible to download the source code of Kali Linux and compile and assemble your own Kali Linux from it. However, this option can also be abused: someone could sneak malicious code into Kali Linux and make the resulting product available for download as Kali Linux. In a book about hacking and security, we hope you realize what the consequences could be if you fall victim to such a manipulation.

To avoid this danger from the outset, you should keep two things in mind:

- Download Kali Linux exclusively from *https://www.kali.org*.
- After downloading, check the SHA checksum and the signature.

You can easily determine the checksum of the downloaded ISO file on Linux and macOS using the `sha256sum` command:

```
sha256sum kali-linux-2022.2-live-amd64.iso
 eee4eab603b10a0618e1900159cb91b8969bf13107e5d834381ecb21a560e149
```

On Windows, you must start a console with PowerShell and run `Get-FileHash`:

```
> Get-FileHash kali-linux-2022.2-live-amd64.iso
```

The resulting code must exactly match the checksum given on the Kali Linux download page.

2.2.2 Verifying the Signature of the Checksum File

You can achieve an even higher level of security if you also ensure that the checksum file is signed by the Kali developers. If you don't check this, it's conceivable that someone has not only slipped you a manipulated ISO file of Kali Linux but also a correspondingly adjusted checksum. What good is the best checksum if you can't trust the authenticity of the checksum itself?

The procedure is a bit more cumbersome because you first need to download the public part of the Kali developers' GPG key and import it into your GPG system. The infrastructure required for this is only available on Linux by default. In the Linux distribution of your choice, you must run the following commands:

```
wget -q -O - https://archive.kali.org/archive-key.asc |
  gpg --import

  gpg: Key ED444FF07D8D0BF6: public key
    "Kali Linux Repository <devel@kali.org>" imported
  ...

gpg --list-keys --with-fingerprint ED444FF07D8D0BF6

  pub   rsa4096 2012-03-05 [SC] [expires: 2025-01-24]
    44C6 513A 8E4F B3D3 0875  F758 ED44 4FF0 7D8D 0BF6
  uid   [ unknown] Kali Linux Repository <devel@kali.org>
  sub   rsa4096 2012-03-05 [E] [expires: 2025-01-24]
```

At *http://cdimage.kali.org*, you'll find a directory for each Kali Linux version, which contains the `SHA256SUMS` and `SHA256SUMS.gpg` files in addition to the ISO files. The first file is simply a text file with the checksums. The second file is a signature of the first file. After downloading both files, you can use the following command to make sure that the checksum file was actually signed by the Kali developers (only the Kali developers have the private part of the key required for this):

```
gpg --verify SHA256SUMS.gpg SHA256SUMS

  Signature made Sun May 15 05:27:29 2022 CEST
    using RSA key 44C6513A8E4FB3D30875F758ED444FF07D8D0BF6
  Good signature from "Kali Linux Repository
    <devel@kali.org>" [unknown]
  WARNING: This key is not certified with a trusted signature!
  There is no indication that the signature belongs to the owner.
  Primary key fingerprint: 44C6 513A 8E4F B3D3 0875
                           F758 ED44 4FF0 7D8D 0BF6
```

At first glance, the output here doesn't inspire confidence. But in fact, everything is in good order: the `gpg` command confirms that the signature and checksum file match. The third line of the `gpg` output starts with `Good signature`. It would be fatal if the message `Wrong signature` were displayed at this point!

The warning following that refers to the fact that the Kali developers' key is self-signed, but this signature could not be verified. You can ignore this warning: you've downloaded the key from *https://www.kali.org* and verified that its fingerprint (i.e., the hexadecimal digit sequence `44C6 513A ... 7D8D 0BF6` in the preceding example) matches the expected values. Furthermore, to verify the signature of the key itself, you and the Kali developers would have to exchange your personal GPG keys either in a face-to-face meeting or via a web of trust (see *https://en.wikipedia.org/wiki/Web_of_trust*).

You can find more details about the download verification at *http://s-prs.co/v569603*.

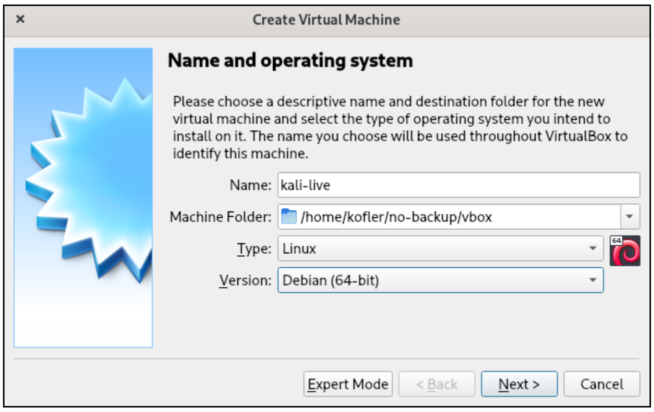
2.2.3 Trying Kali Linux in VirtualBox

Before you can try Kali Linux in VirtualBox, you must first install VirtualBox itself. The installation process is equally straightforward on Windows, macOS, and Linux. Many Linux distributions even provide ready-made VirtualBox packages. On Ubuntu, for example, you install VirtualBox via `apt install virtualbox`.

VirtualBox Alternatives

Of course, you can also use another virtualization system instead of VirtualBox, such as VMware, Hyper-V, or Parallels. We'll focus on VirtualBox in this book. If VirtualBox causes problems on Windows, which unfortunately does happen frequently, you should take a look at Section 2.4.

With VirtualBox installed, start it and set up a new virtual machine via **New**. In the first dialog of the installation wizard, select **Linux** for the operating system **Type** and **Debian (64-bit)** as the **Version** (see Figure 2.1).


Create Virtual Machine

Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: kali-live
Machine Folder: /home/kofler/no-backup/vbox
Type: Linux
Version: Debian (64-bit)

Figure 2.1 Setting Up Virtual Machine for Kali Linux in VirtualBox

In the second dialog, you need to specify how much memory you want to allocate to the virtual machine. For simple tests, 2,048 MiB is recommended. However, some programs available in Kali Linux, such as OpenVAS, require more RAM.

In the third dialog, you can assign an equally virtual hard drive to the virtual machine. However, this is only required for installation. To try it out without installing, you can select the **No hard drive** option.

This also completes the preparatory work. When you start the virtual machine now, VirtualBox asks from which optical drive it should read the boot media. In this dialog, select the **kali-linux-nnn.iso** file, which is probably located in your **Downloads** directory. If this dialog does not appear, you must set up the image file in the virtual machine settings (**Mass Storage** dialog).

VirtualBox then reads the boot files from the virtual DVD drive and displays a boot menu. From this menu, you want to select the **Live (amd64)** entry (see Figure 2.2). (*Live mode* in Linux refers to booting a distribution from a disk without installation.)

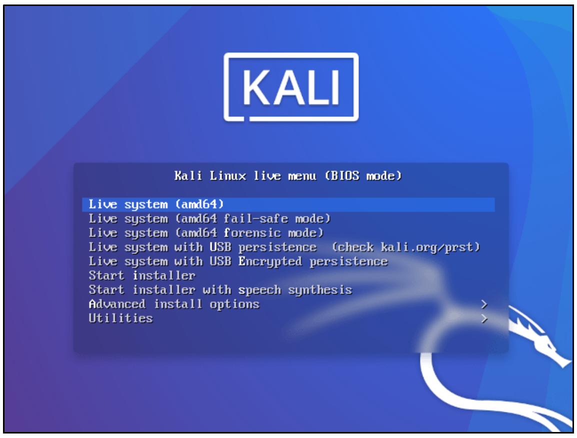

```
Kali Linux live menu (BIOS mode)

Live system (amd64)
Live system (amd64 fail-safe mode)
Live system (amd64 forensic mode)
Live system with USB persistence  (check kali.org/prst)
Live system with USB Encrypted persistence
Start installer
Start installer with speech synthesis
Advanced install options                                  >
Utilities                                                 >
```

Figure 2.2 Kali Boot Menu

A few seconds later, the Kali desktop appears in the VirtualBox window. In the main menu, you'll find a systematically arranged selection of the most important hacking tools. The numbered headings branch to corresponding submenus (see Figure 2.3)!

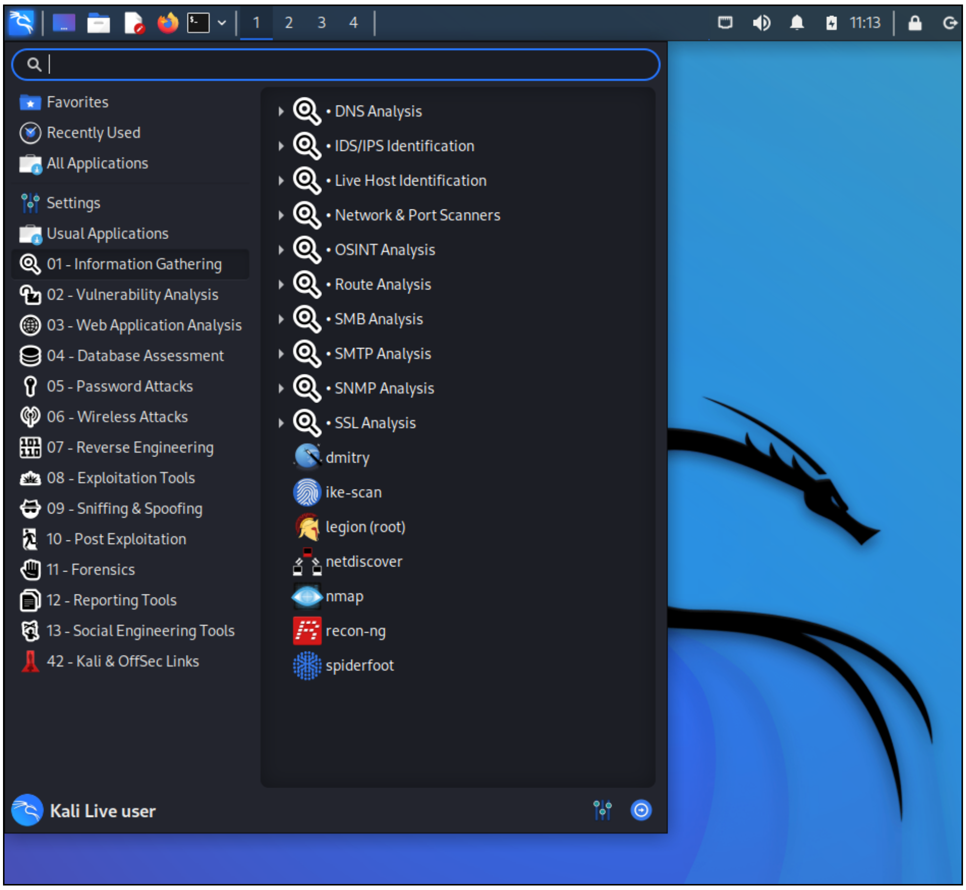


Figure 2.3 Start Menu of Kali Desktop

However, don't be under any illusions: though the start menu is nicely presented graphically, the operation of almost all hacking tools is done in a terminal window. Thus, after the menu selection, a terminal window usually opens. Often a summary of the most important options of the respective command is displayed in it.

After that, you're on your own: you can execute the command in question in the terminal—or any other command. Thus, the start menu only provides assistance in finding hacking commands; it can't save you the trouble of entering the particular command.

As soon as you've memorized the names of the most important commands, you can dispense with the start menu altogether and simply open one or more empty terminal windows to work.

Password for the Lock Screen

After the start you'll be logged in automatically as *kali* and won't need to enter a password. But if you don't use Kali Linux for a long period, the screensaver will be activated automatically. To exit it, you must first press `Enter` and then specify the default password, "kali".

2.2.4 Saving Data Permanently

You can essentially use all functions of Kali Linux in live mode, but you can't permanently save settings or files. Each time the virtual machine is restarted, everything starts from the beginning. The most useful way out of this dilemma is to install Kali Linux permanently, either on a virtual machine (see the following section) or on a real machine.

Apart from the live mode and as an installation, Kali provides a third option—a compromise so to speak: in USB persistence mode, data can be stored on the USB stick where the Kali image file is located. If required, the persistent file system can be encrypted.

Before you can use the persistence functions (with or without encryption), the USB stick must first be prepared accordingly. In particular, a second partition with a Linux file system must be set up on the USB stick. The required steps can only be performed on Linux, for example, on a Linux notebook into which the USB stick intended for Kali has been inserted. You can find detailed instructions at *https://www.kali.org/docs/usb/usb-persistence*.

We believe that the relatively high configuration effort is worth the trouble only in exceptional cases. One such case would be the desire to travel *without* a computer and use your Kali USB stick to work on other computers over and over again. If you want to use Kali Linux on your own computer anyway, the installation into a virtual machine described in the next section is the easier and more convenient option.

2.2.5 Forensic Mode

In the boot menu of Kali Linux, the **Linux (forensic mode)** option is also available for selection. Forensic mode excludes any unintentional modification of files on the computer running Kali Linux. As long as you try Kali Linux on a virtual machine, this mode doesn't bring any advantages. If, on the other hand, you start Kali Linux from a USB stick or from another boot medium on a real computer whose file systems you want to analyze, the forensic mode is absolutely recommended.

2.3 Installing Kali Linux in VirtualBox

Now that you've gotten to know Kali Linux a bit, it's time to install this Linux distribution permanently on a virtual machine. A proper installation has several advantages over the live mode:

- You can keep Kali Linux up to date by installing updates regularly.
- You can install additional packages permanently and increase the functionality of Kali Linux.
- You can permanently store files—for example, with the results of penetration tests.
- You can install the VirtualBox drivers. This allows you, for example, to exchange text between the virtual machine and your host computer via the clipboard.

Installing Kali Linux Directly or Using It in the Cloud

Operating Kali Linux on a virtual machine is ideal for an introduction to the topic of hacking and security. But there are exceptions: for example, if you want to store huge network logging files, you'll find that such I/O-heavy tasks are noticeably slower on a virtual machine.

The solution to this is to install Kali Linux directly on the SSD of your notebook—a matter of course for professional penetration testers. If you want to use the entire disk of your computer completely for Kali Linux, the installation is basically the same as in the following instructions. A parallel installation to Windows or to another Linux distribution is somewhat more complicated: in this case, you must first shrink the existing partitions of the SSD to make room for one or two partitions reserved for Kali Linux. Detailed instructions, which are not Kali-specific but apply to any Linux distribution, can be found in any Linux tutorial.

Another variant is to use an instance of Kali Linux in the cloud. The main advantages are that you have both (very) high computing power and excellent internet connectivity, depending on your needs and budget. Kali Linux is offered for free in Amazon's AWS Marketplace (see *http://s-prs.co/v569604*), among other places. (Kali Linux itself is free, but you have to pay the usual prices for using the cloud instance.)

Trouble with VirtualBox on Windows

If VirtualBox doesn't work on Windows, there can be several reasons. First of all, you should make sure that the virtualization functions of the CPU are activated in the BIOS settings of your computer (Intel VT or AMD V).

The second most common cause of errors is Hyper-V. Theoretically, Microsoft's virtualization system and VirtualBox should be compatible with each other; in real life, however, we have repeatedly found the opposite to be the case. The problem manifests itself in the fact that the Linux kernel of the virtual machine with Kali Linux crashes within a few seconds.

Various posts in the VirtualBox forums recommend simply turning off Hyper-V—but many Windows functions depend on Hyper-V. For this reason, you need to disable the following options in the Windows Features program: Container, Hyper-V, Microsoft Defender Application Guard, Windows Hypervisor Platform, Windows Sandbox, and Windows Subsystem for Linux. Especially on computers that are also used for software development, disabling Hyper-V is not a good idea!

Instead of wasting time on tedious and often useless troubleshooting, it's usually better to run Kali Linux directly on Hyper-V or possibly in the Windows Subsystem for Linux (WSL2). We cover both variants in this chapter: Section 2.4 and Section 2.5.

2.3.1 Option 1: Using a Prebuilt VirtualBox Image

Instead of going through the trouble of installing Kali Linux inside VirtualBox yourself, you can simply use a prebuilt image. This can be found at *https://www.kali.org/get-kali* under **Virtual Machines • 64-bit • VirtualBox**. After downloading the approximately 4 GB *.ova file, you need to run **File • Import Appliance** in VirtualBox. During the import, a new disk image gets created, which is why the process takes about a minute. After starting the virtual machine, log in with the user name *kali* and the password, also *kali*.

2.3.2 Option 2: Installing Kali Linux Yourself

Alternatively, you can install Kali Linux yourself. This entails some more work, but also gives you more control over the process (e.g., in setting the desired disk image size) and is also good practice if you plan to install Kali Linux directly onto your notebook later.

As a basis for the installation, you need to download an almost 3 GB *.iso file from *https://www.kali.org/get-kali* under **Installer Images • 64-bit • Installer**. Then you set up a new virtual machine. In the first dialog, select **Linux** as the type and **Debian (64-bit)** as the version. You should allocate 2 GB of RAM to the virtual machine. You also need to set up a virtual disk. A reasonable size is 25 to 30 GB. Leave all other presets for the disk unchanged. At the first start, select the previously downloaded *.iso file as the installation image. The installation starts with the **Graphical Installer** menu item.

If you've tried the Kali Live system before, you can also use this image for a permanent installation. In this case, you must select the **Start Installer** menu item.

2.3.3 Installation

The installation process is characterized by many queries. This means that the installation can also be adapted to exotic requirements. In the first dialog you set the language, in the second your default location (e.g., Germany). Based on this information, the installer selects a geographically close mirror server for later updates. In the next dialog, you specify the keyboard layout.

After that, Kali performs a network configuration, automatically detecting the network adapter provided by VirtualBox. The installation program suggests *kali* as the host name. For installations on the local network, you do not need to change anything.

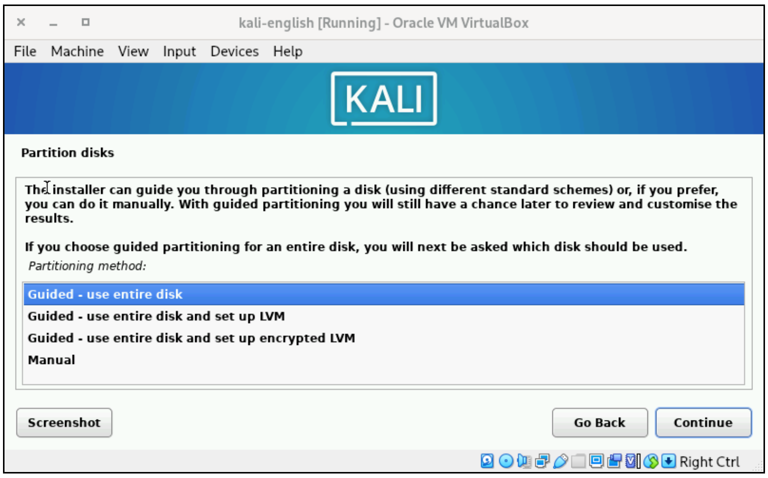
In the dialogs that follow, you will set up a user and assign a password. Once the installation is complete, you can use this user to perform administrative work or run hacking tools with admin privileges thanks to `sudo`. Even if you're only installing on a virtual machine: use a secure password for the Kali account!

The subsequent dialog is about partitioning the hard disk. For "real" installations on a computer on which Kali Linux is to be used in parallel with other operating systems, this is a tricky point. When installing onto a virtual machine, on the other hand, you don't need to consider other operating systems. Therefore, manual partitioning is unnecessary. Instead, you should select one of the following three variants (see Figure 2.4):

- **Guided—use entire disk**
  Kali Linux will use the entire virtual hard disk.
- **Guided—use entire disk and set up LVM**
  Like the previous option, but the Logical Volume Manager (LVM) is set up at the same time. The advantage here is that it is relatively easy later (at least for experienced Linux users) to increase the size of the virtual machine disk and the file system used by Kali Linux. So you gain flexibility without any disadvantages.
- **Guided—use entire disk and set up encrypted LVM**
  Like the previous option, but the LVM system is encrypted at the lowest level. You must specify the password used for encryption every time you start the virtual machine. This has the following advantage: if someone gets hold of your computer, including the Kali virtual machine, they won't be able to read a single file without this password. As things stand today, this encryption is secure. The concept is similar to BitLocker (Windows) or FileVault (macOS).

After selecting the partitioning method, the installation program asks whether you want all files to be installed into a single file system or whether you want separate partitions for your own files (i.e., for the */home* directory) and also for variable data (i.e. for */var* and */tmp*). You should choose the first option! Separation into several partitions doesn't provide any significant advantages in a virtual machine. After several queries about whether you really want to do the partitioning this way, the installation program finally gets down to business and installs a Linux base system into the newly set up file system. This process takes about one minute.

In the next dialog, you can specify the desktop system and the installation scope (see Figure 2.5). Basically, you won't do anything wrong if you simply confirm the defaults. Kali Linux then uses the lean (and fast!) Xfce desktop and includes a basic set of the most important hacking tools. You can install additional programs later if necessary.


kali-english [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
KALI
Screenshot
Go Back
Continue
Right Ctrl

Partition disks

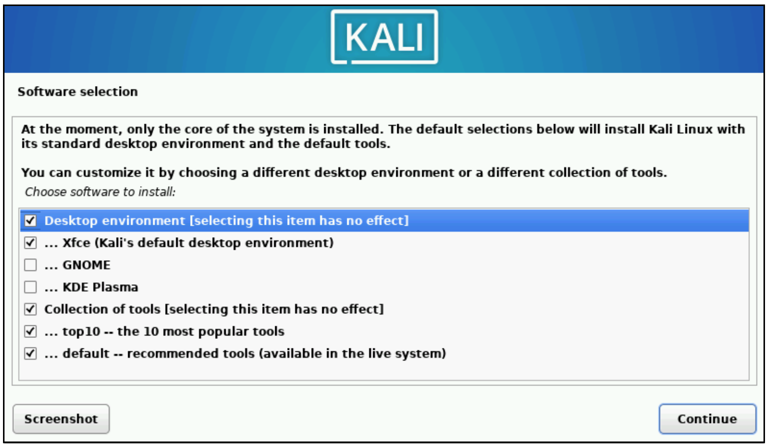
The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

- Guided - use entire disk
- Guided - use entire disk and set up LVM
- Guided - use entire disk and set up encrypted LVM
- Manual

Figure 2.4 Kali Linux: Graphical Installation Program Provides Various Partitioning Options



Software selection

At the moment, only the core of the system is installed. The default selections below will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different collection of tools.

Choose software to install:

- ☑ Desktop environment [selecting this item has no effect]
- ☑ ... Xfce (Kali's default desktop environment)
- ☐ ... GNOME
- ☐ ... KDE Plasma
- ☑ Collection of tools [selecting this item has no effect]
- ☑ ... top10 -- the 10 most popular tools
- ☑ ... default -- recommended tools (available in the live system)

Figure 2.5 Setting Scope of Installation

The installation of the components you just selected takes a few minutes. After that, the installation program wants to set up the GRUB boot loader. This is essential so that Kali Linux can be started once the installation is complete. The appropriate location is the boot sector of the first hard disk, which is designated by the device name `/dev/sda` in Linux terminology.

Further Information

The installation program of Kali Linux corresponds to that of Debian Linux except for a very few details. So if you encounter any problems installing on a real PC, you can find further information and tips in any Debian installation guide. A good place to start is *https://www.debian.org/releases/bullseye/installmanual*.

By the way, Kali Linux is not based on Debian Bullseye, as the provided link might suggest. Rather, Kali follows the rolling release model and uses packages from Debian Testing (Section 2.9).

2.3.4 Login and sudo

After restarting the virtual machine, you must log in. To do this, enter your account name and the password you set during the installation.

In contrast to older Kali versions, you have only "ordinary" rights after login, so you are not the root user. However, your account is assigned to the sudo group. This enables you to execute individual commands in the terminal with root privileges. You will have to authenticate yourself again with your password. (This authentication is then valid for five minutes. During this time, you can use sudo without entering your password again.):

```
user$ sudo command
[sudo] password for user: ********
```

If you want to run multiple commands with root privileges, you can use sudo -s to permanently switch to root mode; Ctrl+D terminates this mode:

```
user$ sudo -s
[sudo] password for user: ********
root# nmap -F  -T4 10.0.0.0/24
root# <Ctrl>+<D>
user$
```

There are also various hacking commands in the Kali menu that only work with root privileges. When executing such commands (e.g., **Information Gathering • Live Host Detection • Arping**) Kali Linux will ask for your password and then run the command with sudo.

2.3.5 Time Zone and Time Display

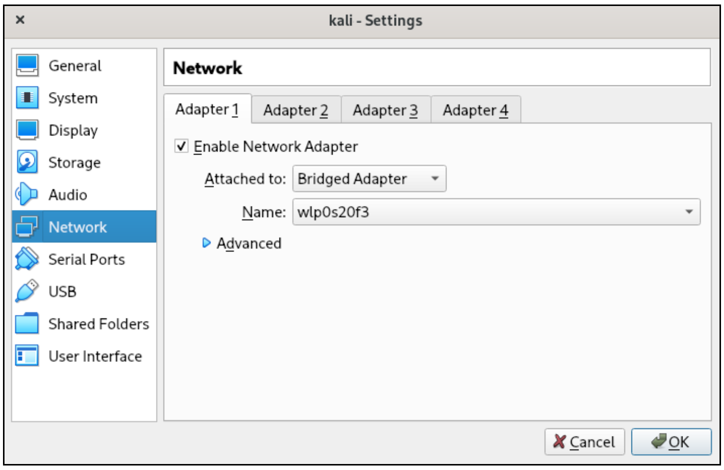
Kali Linux displays the current time in the panel. If the time is not correct or if the a.m./p.m. display confuses you, right-click the time display. The **Properties** dialog allows you to enter the necessary configuration.

2.3.6 Network Connection

By default, VirtualBox uses network address translation (NAT) to establish a network connection to virtual machines, which provides the virtual machines with internet access, but no direct access to the local network. This can be desirable for two reasons:

first, you can then use Kali Linux to check the security of computers on the local network, and second, you'll be able to easily establish an SSH connection to Kali Linux from your local computer.

Thus, to integrate the virtual machine with Kali Linux into the local network, you need to click the **Change** button in VirtualBox. In the **Network** dialog, you want to change the connection type of the first adapter to **Bridged Adapter**. In the list field below, you need to specify how your local computer or notebook is connected to the local network—that is, usually via an Ethernet cable or via WLAN (see Figure 2.6). The network bridge connects this network interface to the virtual machine.



☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: wlp0s20f3

Figure 2.6 Network Settings for Virtual Machine in VirtualBox

Ethernet Cable or WLAN?

If you use a notebook, it's of course obvious to establish the network and internet access via WLAN.

But unless you're concerned about the security of the wireless network itself, it's often better to establish network access via a cable, or even a USB adapter if your fancy notebook lacks an Ethernet jack. This is especially true if your company's WLAN router has its own firewall and assigns its own IP addresses ("guest mode"), and so on, thus making access to the regular company network difficult or impossible altogether.

2.3.7 Using Kali Linux via SSH

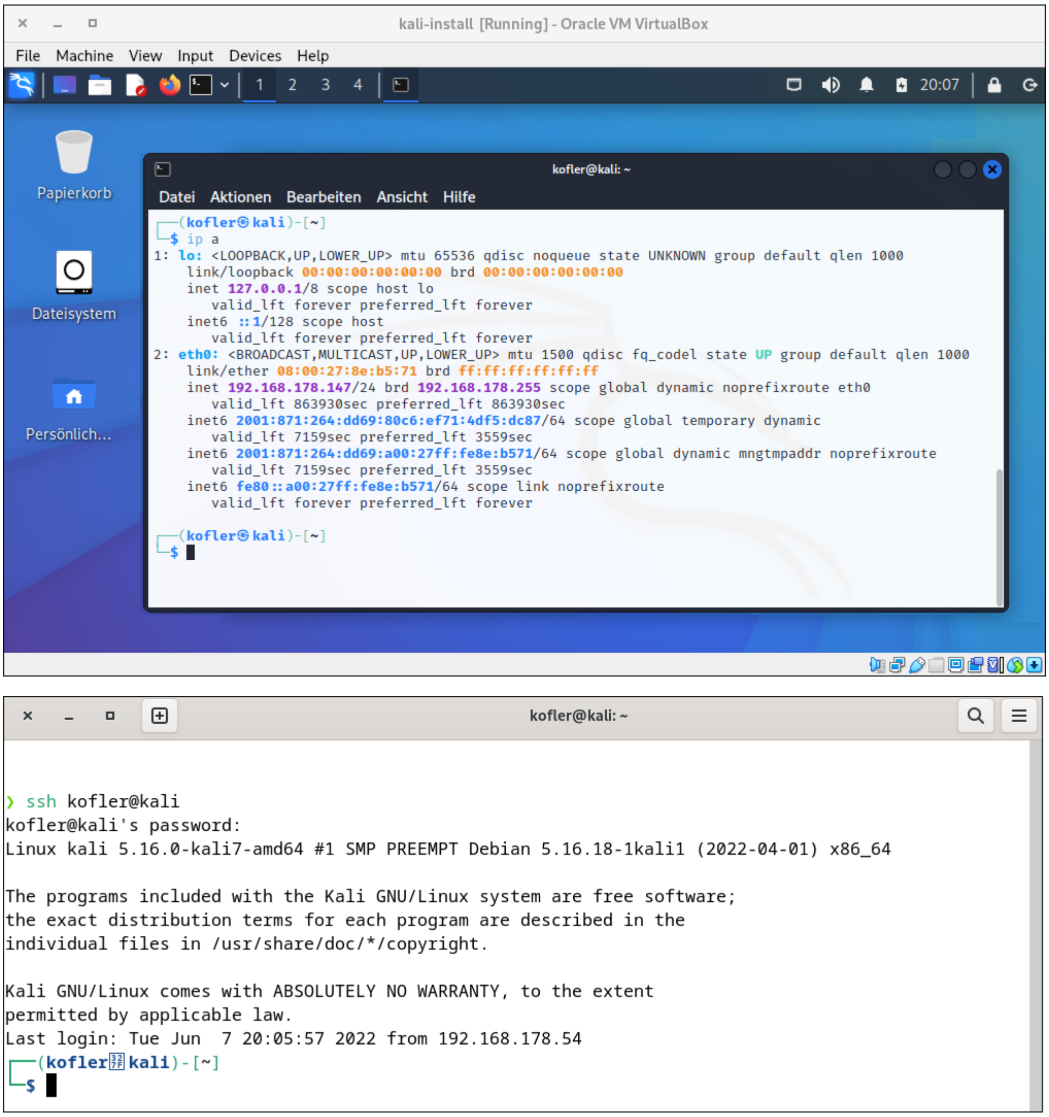
Experienced users whose PCs run Linux or macOS often prefer to operate Kali Linux via a terminal of the host operating system (i.e., not in the virtual machine window). For this to work, you need to change the virtual machine network connection as just described.

Furthermore, you must include the `PasswordAuthentication yes` parameter in `/etc/ssh/sshd_config`. (There is already a corresponding line, but it's preceded by the # comment character. To enable the option, you just need to remove the comment character.)

Finally, you must enable the SSH service, which is installed by default but not running:

```
systemctl enable --now ssh
```

After this preparatory work, you can establish an SSH connection to the Kali Linux virtual machine from a terminal window on your computer (see Figure 2.7).


kali-install [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
20:07
Papierkorb
Dateisystem
Persönlich...
kofler@kali: ~
Datei Aktionen Bearbeiten Ansicht Hilfe
kofler@kali: ~

```
┌──(kofler㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8e:b5:71 brd ff:ff:ff:ff:ff:ff
    inet 192.168.178.147/24 brd 192.168.178.255 scope global dynamic noprefixroute eth0
       valid_lft 863930sec preferred_lft 863930sec
    inet6 2001:871:264:dd69:80c6:ef71:4df5:dc87/64 scope global temporary dynamic
       valid_lft 7159sec preferred_lft 3559sec
    inet6 2001:871:264:dd69:a00:27ff:fe8e:b571/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 7159sec preferred_lft 3559sec
    inet6 fe80::a00:27ff:fe8e:b571/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kofler㉿kali)-[~]
└─$
```

```
❯ ssh kofler@kali
kofler@kali's password:
Linux kali 5.16.0-kali7-amd64 #1 SMP PREEMPT Debian 5.16.18-1kali1 (2022-04-01) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun  7 20:05:57 2022 from 192.168.178.54
┌──(kofler㉿kali)-[~]
└─$
```

Figure 2.7 Top: Kali Linux in VirtualBox Window. Bottom: Terminal Window with SSH Connection to Kali Linux.

I assume here that kali is the host name of Kali Linux. Instead of the host name, you can also specify the IP address of the virtual machine. You can find this by running `ip addr` in Kali Linux. If you want to execute commands with root privileges via SSH, you must switch to root mode after logging in with `sudo -s`:

```
ssh myname@kali

  myname@kali's password: ******
```

2.3.8 Clipboard for Kali Linux and the Host Computer

You can conveniently copy text between your computer and Kali Linux in the virtual machine using the clipboard, provided the following two conditions are met:

- Kali Linux must have the VirtualBox guest extensions installed. This is the case by default in current Kali versions.
- In the virtual machine settings, you must have enabled the **Shared clipboard = bidirectional** setting in the **General • Advanced** dialog. By default, the shared clipboard is disabled.

2.4 Kali Linux and Hyper-V

In the following sections, we assume that you are using Windows Professional and have enabled Hyper-V with all suboptions in the Windows Features program. If so, you can start Hyper-V Manager from the Start menu.

Before you start installing Kali Linux, you must first set up a network switch to connect your virtual machine to the local network. To do this, click the **Virtual Switch Manager** item in the **Actions** pane, create a virtual switch with the **External** type, and connect it to the network adapter of your computer. (Such a switch is equivalent to a network bridge in VirtualBox. Your virtual machine is thus embedded in the local network and obtains the network configuration from there.)

NAT Switch, Where Are You?

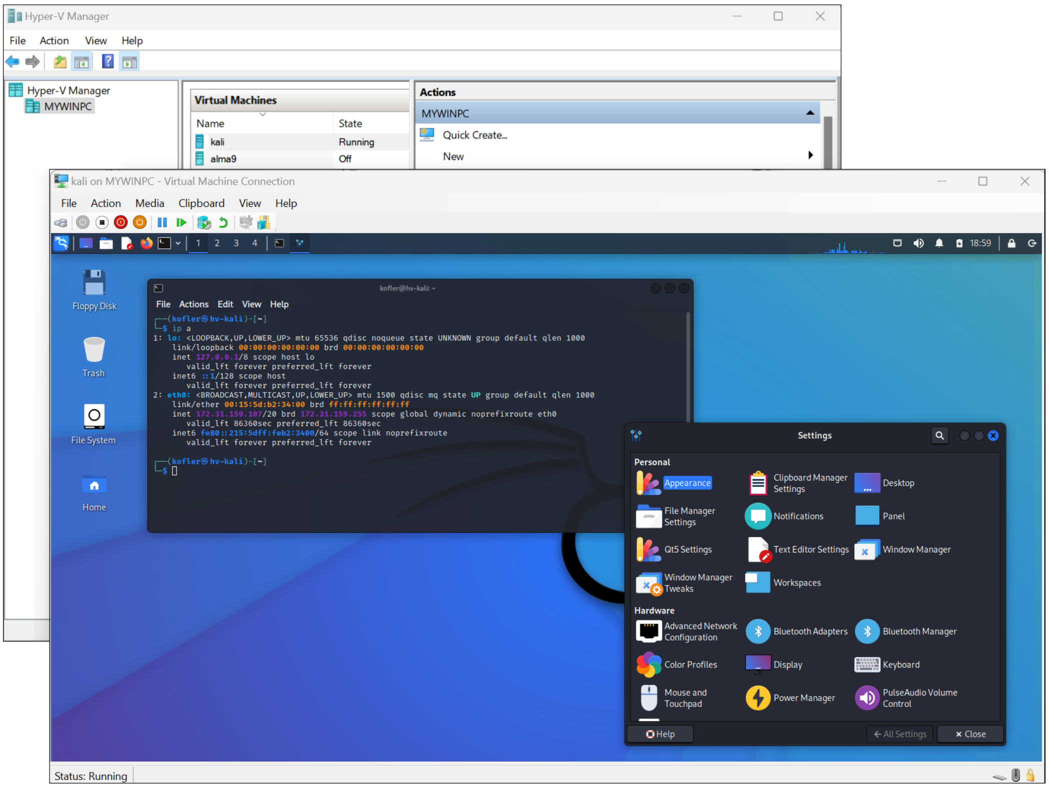
For the typical application of Kali Linux, the external network switch just outlined is ideal. If you are just learning to use Kali Linux and only want to "hack" another virtual machine, then running it on a private network would make more sense.

In terms of network configuration, Hyper-V is way behind the competition. You cannot set up a switch with network address translation in Hyper-V Manager, nor does Hyper-V contain any features to run a simple DHCP server. On the internet, you can find various tips on how to fix these shortcomings (search for "hyper-v nat dhcp"). One useful guideline is the following: *http://s-prs.co/v569605*.

You can use **New • Virtual Computer** to start configuring a new virtual machine. A wizard will guide you through this process. In this process, you first give the virtual machine a name. In the second step, you can choose between two VM types: **Generation 1** or **Generation 2**. The types differ primarily in their EFI support. Generation 1 is sufficient for Kali Linux.

Next, you assign the memory to the virtual machine. As long as you don't use memory-intensive tools, 2 GB is sufficient. During network configuration, you need to select the previously configured switch into the local network. A reasonable size for the virtual hard disk is 25 to 30 GB. Finally, under **installation options**, you want to select the Kali Linux ISO file as the installation source.

After starting the virtual machine, the actual installation proceeds exactly as in VirtualBox (see Figure 2.8).



```
┌──(kofler㉿hv-kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:b2:34:00 brd ff:ff:ff:ff:ff:ff
    inet 172.31.159.107/20 brd 172.31.159.255 scope global dynamic noprefixroute eth0
       valid_lft 86360sec preferred_lft 86360sec
    inet6 fe80::215:5dff:feb2:3400/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kofler㉿hv-kali)-[~]
└─$ 
```

Figure 2.8 In the Background, Hyper-V Manager; in the Foreground, Kali Linux

After the reboot, the graphics system of Kali Linux runs in a resolution of 1,152×864 pixels. The resolution can only be set by changing a Linux kernel option. To do this, you must use an editor to modify the `/etc/default/grub` file and add the following option to `GRUB_CMDLINE_LINUX_DEFAULT`:

```
# File /etc/default/grub
...
GRUB_CMDLINE_LINUX_DEFAULT="... video=hyperv_fb:1920x1080"
```

After that, update the GRUB installation and reboot the virtual machine:

```
update-grub
reboot
```

By default, only one CPU core is assigned to the virtual machine. If your computer has a CPU with multiple cores, it's advisable to assign two or more cores to Kali Linux. To do this, shut down the virtual machine and then open its settings. In the hardware settings, you can change the number of cores in the **Processor** dialog.

2.5 Kali Linux in the Windows Subsystem for Linux

A relatively new variant for running Kali Linux is the Windows Subsystem for Linux, which is also available for Windows Home.

Two steps are required for the installation:

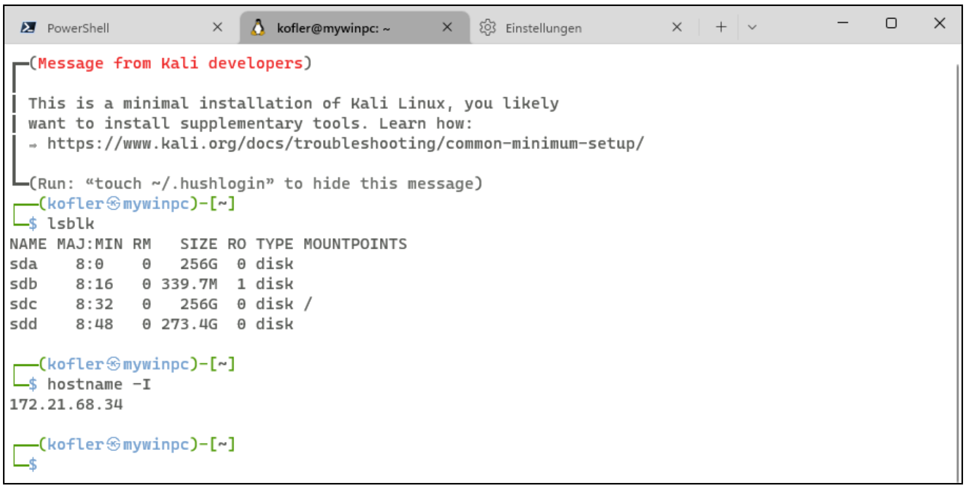
1. First, start the Enable or Disable Windows Features program and enable the **Windows Subsystem for Linux** option. The activation requires a Windows restart.
2. After that, launch the Microsoft Store, search for "Kali Linux", and click the **Download** and **Launch** buttons. A little later, you'll need to set a user name (not root!) and password.

After the login, Kali Linux runs in a window that looks like the cmd.exe program or PowerShell (see Figure 2.9). To work with root privileges in Kali Linux, you must use `sudo -s`.

Kali Linux for WSL is reduced to the absolute minimum; even very basic tools like `nmap` are missing. This means you must install the hacking tools you need by yourself:

```
sudo apt update
sudo apt install nmap
```

If you close the Kali WSL window, you can run Kali again later from the Windows Start menu (select the **Kali Linux App** entry). Alternatively, you can start the Terminal program and select **kali-linux** from the tab menu to run Kali Linux in a pane of the terminal. A third startup option is provided by the `wsl -d kali-linux` command, which you must execute in a console.



```
┌──(Message from Kali developers)
│
│ This is a minimal installation of Kali Linux, you likely
│ want to install supplementary tools. Learn how:
│ ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
└──(Run: “touch ~/.hushlogin” to hide this message)
┌──(kofler㉿mywinpc)-[~]
└─$ lsblk
NAME MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda    8:0    0   256G  0 disk
sdb    8:16   0 339.7M  1 disk
sdc    8:32   0   256G  0 disk /
sdd    8:48   0 273.4G  0 disk

┌──(kofler㉿mywinpc)-[~]
└─$ hostname -I
172.21.68.34

┌──(kofler㉿mywinpc)-[~]
└─$
```

Figure 2.9 WSL Variant of Kali Linux Typically Runs in Text Mode

2.5.1 Kali Linux in Graphic Mode

By default, Kali Linux runs in text mode in WSL. But that can be easily changed! To do this, you need to install the `kali-win-kex` package within Kali Linux:

```
sudo apt update
sudo apt install kali-win-kex
```

Then execute (without `sudo`) the `kex` command. It sets up a VNC server and first asks twice for a password for the VNC connection. Optionally, you can set up another password for a read-only operation, but this is rarely practical:

```
kex

  Password: *******
  Verify:   *******
  Would you like to enter a view-only password: n
```

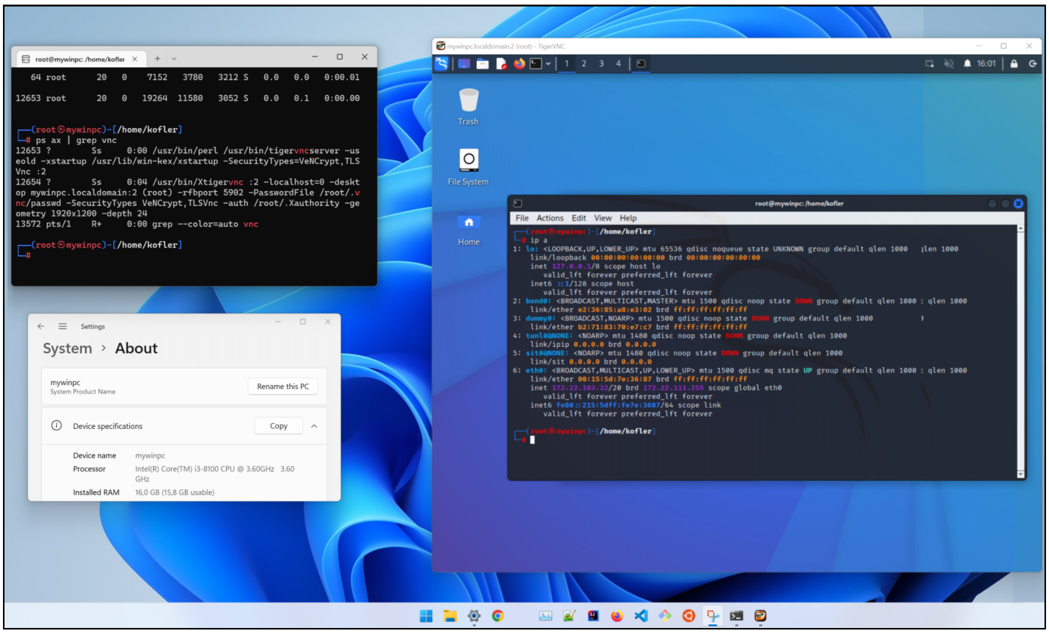
After that, Kali Linux appears in full screen mode (see Figure 2.10). Pressing F8 takes you to a context menu of the VNC viewer. There you can not only disable the **Full Screen** option, but also minimize the VNC window or end the session altogether (**Exit viewer**). The Kali desktop automatically adjusts to the window size of the VNC viewer.

Seamless Mode

Another way to run Kali Linux in full screen mode or in a window is to start the seamless mode:

```
kex --sl
```

At first startup, you must accept a firewall exception rule for the `vcxsrv` program. As a result, Kali displays a panel at the top of the Windows desktop. There you can launch Kali programs that appear in parallel with Windows programs in separate windows on the common Windows desktop. (Similar features are provided by commercial virtualization systems such as VMware and Parallels.) Seamless mode allows Kali Linux and Windows to coexist seamlessly. You can find more tips about using Kali Linux in graphics mode at *https://www.kali.org/docs/wsl*.


root@mywinpc: /home/kofler
mywinpc.localdomain:2 (root) - TigerVNC
Trash
File System
Home
root@mywinpc: /home/kofler
File Actions Edit View Help
Settings
System › About
Rename this PC
Device specifications
Copy

```
   64 root      20   0    7152   3780   3212 S   0.0   0.0   0:00.01

12653 root      20   0   19264  11580   3052 S   0.0   0.1   0:00.00

┌──(root㉿mywinpc)-[/home/kofler]
└─# ps ax | grep vnc
12653 ?        Ss     0:00 /usr/bin/perl /usr/bin/tigervncserver -us
eold -xstartup /usr/lib/win-kex/xstartup -SecurityTypes=VeNCrypt,TLS
Vnc :2
12654 ?        Ss     0:04 /usr/bin/Xtigervnc :2 -localhost=0 -deskt
op mywinpc.localdomain:2 (root) -rfbport 5902 -PasswordFile /root/.v
nc/passwd -SecurityTypes VeNCrypt,TLSVnc -auth /root/.Xauthority -ge
ometry 1920x1200 -depth 24
13572 pts/1    R+     0:00 grep --color=auto vnc

┌──(root㉿mywinpc)-[/home/kofler]
└─#
```

mywinpc
System Product Name

Device name	mywinpc
Processor	Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz 3.60 GHz
Installed RAM	16,0 GB (15,8 GB usable)

```
┌──(root㉿mywinpc)-[/home/kofler]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000   ;len 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000 : qlen 1000
    link/ether e2:36:85:a8:e3:02 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000          )
    link/ether b2:71:83:70:e7:c7 brd ff:ff:ff:ff:ff:ff
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000 : qlen 1000
    link/ether 00:15:5d:7e:36:87 brd ff:ff:ff:ff:ff:ff
    inet 172.22.103.32/20 brd 172.22.111.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe7e:3687/64 scope link
       valid_lft forever preferred_lft forever

┌──(root㉿mywinpc)-[/home/kofler]
└─#
```

Figure 2.10 Kali Linux in VNC Window on Windows Desktop

2.5.2 WSL1 versus WSL2

WSL has been available in version 2 for several years, and this version is used by default. As an alternative, you can still use WSL1. Why's that?

WSL1 and WSL2 differ fundamentally from a technological point of view. With WSL1, the most important Linux kernel functions are emulated by Microsoft software. WSL2, on the other hand, runs a real Linux kernel. WSL2 therefore offers a much greater degree of compatibility and, for most applications, significantly higher speed. Also, the graphics mode can only be used in combination with WSL2.

The decisive disadvantage of WSL2 (from Kali's point of view) is the network connection: while Kali Linux shares the host computer's network connection in WSL1, Kali Linux receives an IP address in a private network in WSL2. This makes penetration testing within the local network largely impossible.

The `wsl` command, which you can run either in cmd.exe or in PowerShell, determines a list of all WSL distributions and switches them between WSL1 and WSL2 as needed:

```
> wsl -l -v

    NAME           STATE            VERSION
  * Ubuntu         Stopped          2
    kali-linux     Running          2

> wsl --set-version kali-linux 1

  The conversion will be executed. This process can take some
  minutes...
> wsl -l -v

    NAME           STATE            VERSION
  * Ubuntu         Stopped          2
    kali-linux     Stopped          1
```

2.5.3 Practical Experience

Kali Linux runs amazingly fast in WSL and without noticeable delays even in graphics mode. The biggest problem in our tests was the network connection mentioned earlier: when Kali runs with WSL1, commands like `nmap` don't work because they can't access the network interface (you'll see *failed to open device eth0* and so on). Similar limitations unfortunately apply to other commands that require low-level access to network functions or hardware.

In Kali Linux with WSL2, there are no driver problems. But because Kali Linux is now on a private network, most hacking tasks can't be performed.

2.6 Kali Linux on Raspberry Pi

Kali Linux is also available in a version for the Raspberry Pi and various other minicomputers with ARM CPUs. You can download ready-made images from *https://www.kali.org/get-kali* under **ARM • Raspberry Pi 2, 3, 4 and 400**. Both a 32-bit and a 64-bit version are available there. Currently, the 32-bit version is still recommended.

You can use Etcher (*https://www.balena.io/etcher*) or a similar tool to write the image to an SD card (minimum size: 16 GB), plug it into the Raspberry Pi, and start the computer. When you log in for the first time, you log in as kali using the password kali.

This allows you to get started right away! The Raspberry Pi version of Kali Linux runs the Xfce desktop just like the PC variant. Kali Linux thus looks and behaves the same as the PC variant.

Kali Linux on the Raspberry Pi displays menus and dialogs basically in English. If you prefer other languages—like German, for example—you can use an editor to enter the following two statements in the `/etc/default/locale` file:

```
LANG=de_DE.UTF-8
LANGUAGE=de_DE:de
```

Kali Linux runs extremely smoothly on current Raspberry Pi models. However, CPU- or I/O-intensive tools are still noticeably slower than on a modern notebook. So long as you don't call memory-intensive tools, 1 GB of working memory is absolutely sufficient; otherwise, you should opt for an only slightly more expensive model with 2 or 4 GB of RAM.

2.7 Running Kali Linux on Apple PCs with ARM CPUs

If you use an "old" Apple machine with an x86 processor, you should first install VirtualBox and there, in a virtual machine, Kali Linux. The procedure is exactly the same as on Windows or Linux.

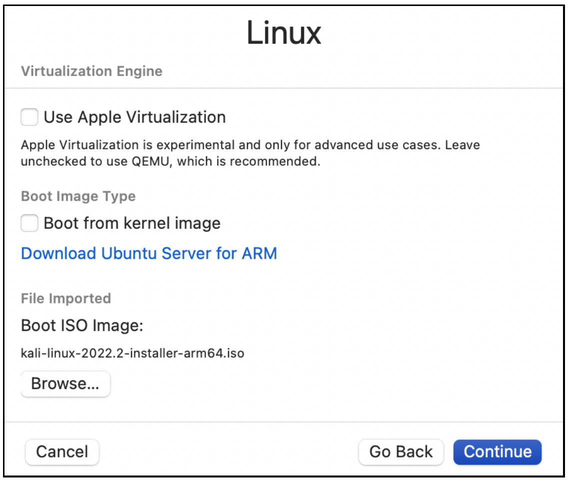
The situation changes if you own a newer device with an M1 or M2 CPU (Apple silicon based on ARM) as VirtualBox is not available for this CPU architecture. You can choose between the two commercial and relatively expensive virtualization systems Parallels and VMware Fusion. You can also use the UTM program, based on the virtualization software QEMU that's pretty popular in the Linux world.

We'll focus on UTM in this section. You can either buy the app for only $10 in the Apple Store and thus support the developers a little bit or download it for free from *https://mac.getutm.app*.

As a basis for installation, select the **Installer Images • Apple M1 • Installer** variant on the *https://www.kali.org/get-kali* page. The resulting ISO file is about 2.5 GB in size.

Then you want to set up a new virtual machine in UTM. In the first dialog of the wizard, select the **Virtualize** option, and in the second dialog, specify that you want to run a Linux distribution. In the third dialog, use **Browse** to select the previously downloaded Kali installation image (see Figure 2.11). The **Use Apple Virtualization** and **Boot from kernel image** options remain disabled.

The next two dialogs are about the hardware equipment of the virtual machine. Depending on how extensively equipped your Mac is, you should allocate 2 to 4 GB of RAM and two CPU cores to the virtual machine. Kali requires a virtual disk of at least 15 GB. With 20 to 25 GB, you have a little space reserve. In the following **Shared Directory** dialog, you can select a macOS directory for data exchange with Kali Linux. As the use of this shared directory is not provided for under Kali Linux, you can skip this step.



Virtualization Engine

Use Apple Virtualization

Apple Virtualization is experimental and only for advanced use cases. Leave unchecked to use QEMU, which is recommended.

Boot Image Type

Boot from kernel image

Download Ubuntu Server for ARM

File Imported

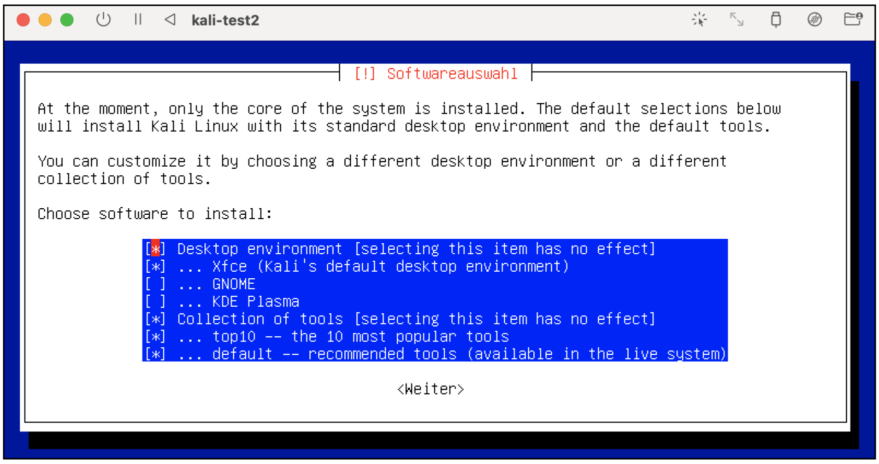
Boot ISO Image:

kali-linux-2022.2-installer-arm64.iso

Figure 2.11 Setting Up Virtual Machine in UTM

In the final **Summary** dialog, you should enable the **Open VM Settings** option. This subsequently gives you the option to choose among several network modes. For using Kali Linux, **Bridged** is mostly recommended: this provides Kali Linux with an IP address on the local network and allows it to communicate with the local network. (However, you can also make this setting later. To do this, stop the virtual machine and then open the virtual machine configuration dialog in the main UTM window.)

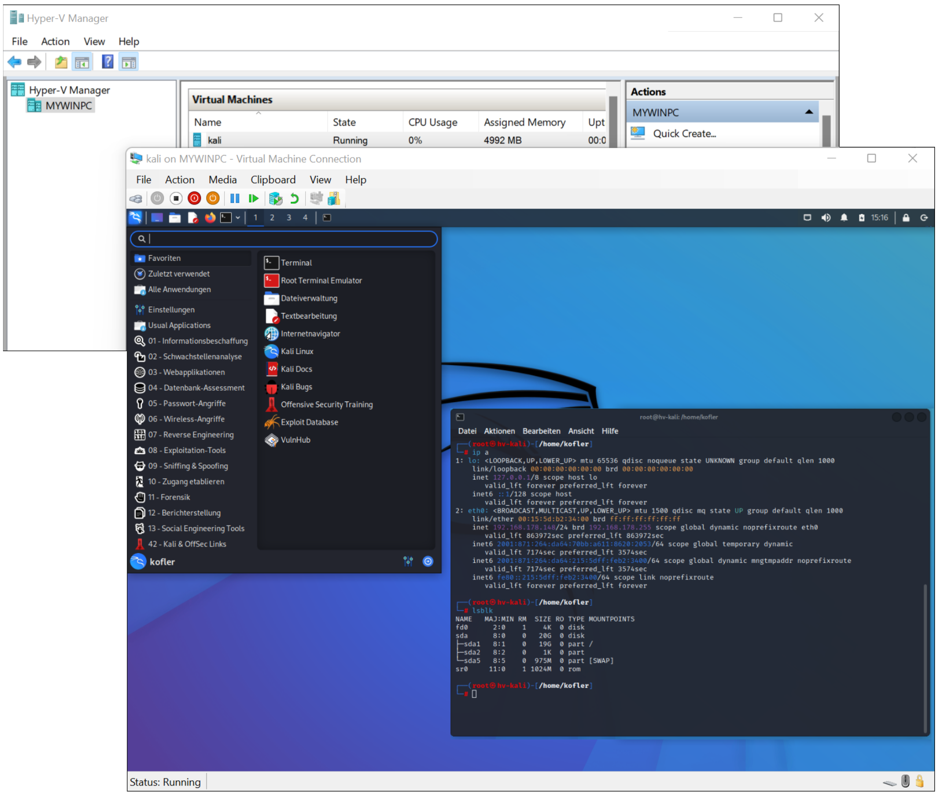
After starting the virtual machine, you will enter the Kali boot menu. In our tests, the **Graphical Install** command did not prove useful: the UTM window will turn completely black after a few seconds, preventing operation of the installation program. Therefore, you should choose **Install** and choose an installation in text mode. The procedure is exactly the same as for a graphic mode installation, but the dialogs look less nice, and the tab key must be used to navigate between the input fields (see Figure 2.12).



```
[!] Softwareauswahl

At the moment, only the core of the system is installed. The default selections below
will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different
collection of tools.

Choose software to install:

        [*] Desktop environment [selecting this item has no effect]
        [*] ... Xfce (Kali's default desktop environment)
        [ ] ... GNOME
        [ ] ... KDE Plasma
        [*] Collection of tools [selecting this item has no effect]
        [*] ... top10 -- the 10 most popular tools
        [*] ... default -- recommended tools (available in the live system)

                              <Weiter>
```

Figure 2.12 Kali Installer in Text Mode

Once the installation is completed, the virtual machine will be restarted. But instead of the freshly installed system, the installation program appears again. This is because the virtual machine still uses the ISO image as the boot medium. Stop the virtual machine using the **Shut down** button, then click the CD icon in the toolbar on the right of the window title and run **CD/DVD (ISO) Image • Eject**. On the next reboot, Kali Linux boots from the virtual disk and then fortunately also runs in graphics mode (see Figure 2.13).



```
┌──(root㉿hv-kali)-[/home/kofler]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:b2:34:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.178.148/24 brd 192.168.178.255 scope global dynamic noprefixroute eth0
       valid_lft 863972sec preferred_lft 863972sec
    inet6 2001:871:264:da64:70bb:a611:8620:2053/64 scope global temporary dynamic
       valid_lft 7174sec preferred_lft 3574sec
    inet6 2001:871:264:da64:215:5dff:feb2:3400/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 7174sec preferred_lft 3574sec
    inet6 fe80::215:5dff:feb2:3400/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(root㉿hv-kali)-[/home/kofler]
└─# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
fd0      2:0    1    4K  0 disk
sda      8:0    0   20G  0 disk
├─sda1   8:1    0   19G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sr0     11:0    1 1024M  0 rom

┌──(root㉿hv-kali)-[/home/kofler]
└─# 
```

Figure 2.13 Kali Linux in a Window of the UTM Window on macOS

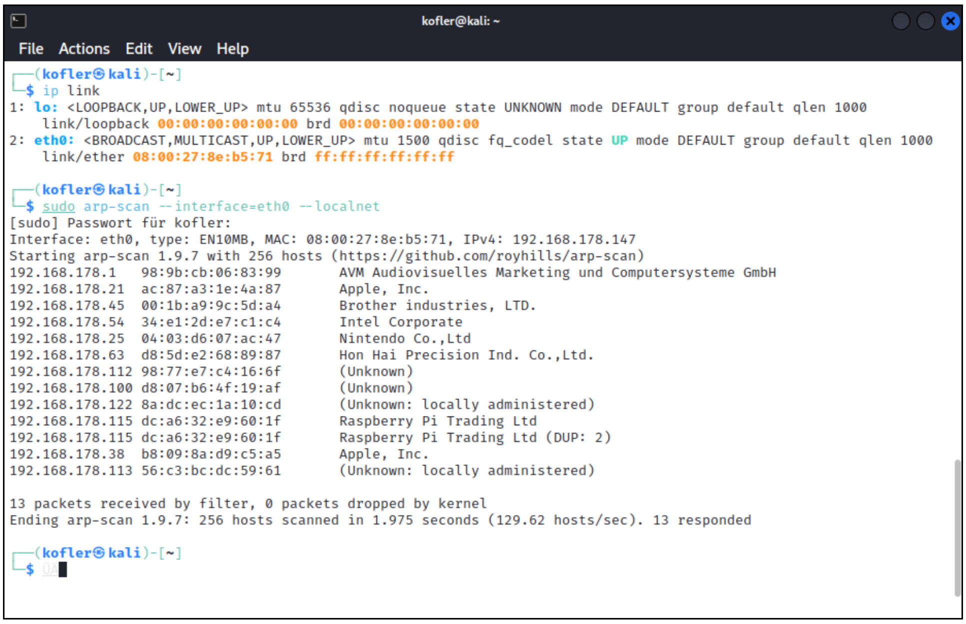
You can set the desired desktop resolution (and thus the window size) via **Settings • Display**. In our tests, Kali Linux worked perfectly within UTM. But due to the automatic scaling between the virtual machine's graphics system and the Mac's monitor, there are unsightly color shifts in the screen display, especially when displaying text.

2.8 Simple Application Examples

You can think of this section as a sort of "Hello World: Hacking!" Some very simple examples show what you can do with Kali Linux. Things really get down to business in future chapters, but to get to know Kali Linux and get a feel for how to work in the terminal, the following examples are perfect.

2.8.1 Address Scan on the Local Network

The first example is not a “real” hacking example, but an innocuous administrative task: you want to find out the IP addresses of all devices on the local network at home or in your company. This can be interesting, for example, because some computers can use the DLNA client or printer on the local network as if by magic, while other devices cannot. Mostly these are older devices that do not support modern zero configuration networking (zeroconf) protocols. These devices could use the printer or DLNA client just as well if only the IP addresses of the printer or DLNA client were known. And it is precisely these addresses that you now want to find out. Kali Linux provides various commands that can help you with this. Unless you’re working with an IPv6 network, `arp-scan` is a good choice. The command is easy to use and returns the desired results within a fraction of a second. You simply run it in a terminal window (see Figure 2.14).



```
┌──(kofler㉿kali)-[~]
└─$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:8e:b5:71 brd ff:ff:ff:ff:ff:ff

┌──(kofler㉿kali)-[~]
└─$ sudo arp-scan --interface=eth0 --localnet
[sudo] Passwort für kofler:
Interface: eth0, type: EN10MB, MAC: 08:00:27:8e:b5:71, IPv4: 192.168.178.147
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.178.1   98:9b:cb:06:83:99       AVM Audiovisuelles Marketing und Computersysteme GmbH
192.168.178.21  ac:87:a3:1e:4a:87       Apple, Inc.
192.168.178.45  00:1b:a9:9c:5d:a4       Brother industries, LTD.
192.168.178.54  34:e1:2d:e7:c1:c4       Intel Corporate
192.168.178.25  04:03:d6:07:ac:47       Nintendo Co.,Ltd
192.168.178.63  d8:5d:e2:68:89:87       Hon Hai Precision Ind. Co.,Ltd.
192.168.178.112 98:77:e7:c4:16:6f       (Unknown)
192.168.178.100 d8:07:b6:4f:19:af       (Unknown)
192.168.178.122 8a:dc:ec:1a:10:cd       (Unknown: locally administered)
192.168.178.115 dc:a6:32:e9:60:1f       Raspberry Pi Trading Ltd
192.168.178.115 dc:a6:32:e9:60:1f       Raspberry Pi Trading Ltd (DUP: 2)
192.168.178.38  b8:09:8a:d9:c5:a5       Apple, Inc.
192.168.178.113 56:c3:bc:dc:59:61       (Unknown: locally administered)

13 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.975 seconds (129.62 hosts/sec). 13 responded

┌──(kofler㉿kali)-[~]
└─$ ÜÄ
```

Figure 2.14 Simple Address Scan on Local Network

Because `arp-scan`, like many other hacking tools, requires administrator privileges, you must precede the command with `sudo`. This requires you to enter your password again. Then the command gets executed with root privileges.

The `arp-scan` command requires that Kali Linux has access to the local network. If you run Kali Linux in a virtual machine, the network interface must be configured as a bridge. (With most virtualization programs, you can also change this setting later. You may need to shut down the virtual machine to do this.) Use `hostname -I` to determine the active IP addresses of Kali Linux if required.

Instead of eth0, you must specify the active network interface of Kali Linux. On a Raspberry Pi with a WLAN connection, this is wlan0. In general, you can list all active interfaces using the ip link command:

```
sudo arp-scan --interface=eth0 --localnet
[sudo] Password for <username>: ********

  Interface: eth0, datalink type: EN10MB (Ethernet)
  Starting arp-scan 1.9.5 with 256 hosts
    (https://github.com/royhills/arp-scan)
  192.168.178.1      98:9b:cb:06:83:99   AVM Audiovisual Ma...
  192.168.178.21     ac:87:a3:1e:4a:87   Apple, Inc.
  192.168.178.45     00:1b:a9:9c:5d:a4   Brother industries, LTD.
  192.168.178.54     34:e1:2d:e7:c1:c4   Intel Corporate
  192.168.178.25     04:03:d6:07:ac:47   Nintendo Co.,Ltd
  192.168.178.100    d8:07:b6:4f:19:af   (Unknown)
  192.168.178.112    98:77:e7:c4:16:6f   (Unknown)
  192.168.178.63     d8:5d:e2:68:89:87   Hon Hai Precision Ind. Co
  192.168.178.122    8a:dc:ec:1a:10:cd   (Unknown: locally admi...
  192.168.178.124    2e:fb:ca:4f:90:41   (Unknown: locally admi...
  192.168.178.115    dc:a6:32:e9:60:1f   Raspberry Pi Trading Ltd
  192.168.178.38     b8:09:8a:d9:c5:a5   Apple, Inc.

  13 packets received by filter, 0 packets dropped by kernel
  Ending arp-scan: 256 hosts scanned in 1.92 seconds.
```

The arp-scan command provides the IP and MAC addresses for all devices found on the local network. Some devices can be assigned a manufacturer based on their MAC address, which of course facilitates identification. If that isn't the case, you can use the host command to try to find the hostname of the device in question:

```
host 192.168.178.100
  100.178.168.192.in-addr.arpa domain name
    pointer Archer-C6.fritz.box.
```

Thus, hidden behind the IP address 192.168.178.100, there's a device named Archer-C6. A quick internet search reveals that this is a WLAN router from TP-Link.

2.8.2 Port Scan of a Server

You can use a port scan to determine which network services a computer provides to the outside world. For a web server, these are ports 80 and 443; for a mail server, 25, 143, 587, and 993. Basically, the motto is that a computer—and especially a server accessible from the internet—should not have more open ports than is absolutely necessary.

The nmap command “knocks” on all ports and evaluates the responses. With the options used in following example, nmap performs a relatively thorough scan that takes some time (about a minute). For reasons of space, we have reproduced the result in a highly abbreviated form:

```
nmap -T4 -A <hostname>

  Starting Nmap 7.92 ( https://nmap.org )
  Nmap scan report for <hostname>
  Host is up (0.037s latency).
  Other addresses for <hostname> (not scanned): 2a01:...
  rDNS record for n.n.n.n: <hostname>
  Not shown: 990 closed ports

  PORT    STATE    SERVICE      VERSION
  22/tcp  open     ssh          OpenSSH 8.2p1 Ubuntu
    ssh-hostkey:
      2048 2c:4a:df:c8:1c:6b:5a:8b:91:d3:da:23:ec:ed:46:9b (RSA)
      256 6f:6f:e2:bb:07:07:83:24:e3:a0:20:c3:d4:5e:6e:d5 (ECDSA)
      256 b0:a9:58:4b:26:fa:0d:6a:fe:76:0e:fe:3c:39:12:36 (...)
  25/tcp  open     smtp         Postfix smtpd
    ssl-cert: Subject: commonName=<hostname>
    Subject Alternative Name: DNS:<hostname2>, DNS: <hostname3>, ...
    smtp-commands: PIPELINING, SIZE 20480000, ETRN, STARTTLS,
      AUTH PLAIN, ENHANCEDSTATUSCODES, 8BITMIME, DSN,  ...
  80/tcp  open     http         Apache httpd 2.4.41 ((Ubuntu))
      ...
  135/tcp filtered msrpc
  139/tcp filtered netbios-ssn
  143/tcp open     imap         Dovecot imapd
  ...
  Nmap done: 1 IP address (1 host up) scanned in 25.51 seconds
```

Perform Scans Only with the Consent of the Host Owner!

You should generally start a port scan only for your own servers or after getting approval from the server administrator. Although a port scan is not a break-in, it can be considered an unfriendly act. It would be like sneaking around someone’s house to see if there are any open windows or doors. Such behavior is unlikely to meet with enthusiasm from the owner.

The nmap command provides more accurate results the more direct the connection between Kali Linux and the target computers is. Routers, firewalls, and so on located between Kali Linux and the target computer can falsify the results, in both directions:

on the one hand, nmap may miss some details that are blocked by firewalls; on the other hand, nmap sometimes includes services in the results that are offered not by the target machine, but by another machine between the target machine and Kali Linux—such as a router. You can avoid such errors by running nmap on another computer as close as possible to the target computer. The nmap command can be installed effortlessly on almost any Linux distribution. You don't need Kali Linux to run nmap!

db_map versus nmap

Within the Metasploit exploit toolkit, the db_nmap command is available as an alternative to nmap. It stores the scan results in a database so that you can access them later without calling nmap again, thus saving you time.

2.8.3 Hacking Metasploitable

To get to know Kali Linux better, you can install the Metasploitable test system in a virtual machine and then "attack" or "hack" it. Installation instructions for Metasploitable 2 and 3, as well as a few simple hacking examples, can be found in Chapter 3.

2.9 Internal Details of Kali

In this section, we'll give you some tips on configuring and securing Kali Linux. We'll also briefly summarize how Kali Linux differs from other Linux distributions.

2.9.1 Basic Coverage

If you use a prebuilt image of Kali Linux, both the username and the associated password are kali. This is unsafe! To change the password, you want to execute the passwd command in a terminal window.

The sudo command mentioned in the previous examples allows other commands to be executed with root privileges. However, this is only permitted for selected users—usually the account created during installation or kali. In addition, sudo usually prompts you for your own password beforehand.

A peculiarity of the Raspberry Pi variant of Kali Linux is that sudo works without a password prompt. This is convenient, of course, but confusing from a security point of view. A solution to this could look as follows: open the /etc/sudoers file in an editor (e.g., by running sudo nano /etc/sudoers in a terminal window), then remove the line kali ALL=(ALL) NOPASSWD at the end of the file: ALL:

```
# remove from /etc/sudoers
...
kali ALL=(ALL) NOPASSWD: ALL
```

2.9.2 Package Sources

Kali Linux is based on the testing branch of Debian Linux, so it generally uses more up-to-date software than the official stable branch of Debian. However, Kali Linux doesn't access the Debian package sources directly but has its own package source. In many cases, this package source contains the same packages as the Debian testing branch, but in some cases newer packages from the unstable or experimental branches also. Sometimes the packages are modified to be Kali-specific. Finally, the Kali package source also contains some packages that are not maintained in Debian at all. You can read details about the specifics of working with Debian at *https://docs.kali.org/policy/kali-linux-relationship-with-debian*.

The package sources of Kali Linux are defined in the */etc/apt/sources.list* file. There is only one active entry in it:

```
# File /etc/apt/sources.list
deb http://http.kali.org/kali kali-rolling main non-free contrib
```

2.9.3 Rolling Release

Unlike most other Linux distributions, Kali Linux is maintained as a *rolling release*. This means that the update system is used not only for security updates but also to regularly update the entire system. Thus, the latest versions of the Linux kernel and various hacking tools are also delivered as part of the updates. New installations of Kali Linux are therefore rarely necessary; updates are usually sufficient.

The rolling release model is not undisputed in the Linux scene. As desirable as the highly up-to-date nature of the software versions is, regular updates also bear the risk of stability problems.

2.9.4 Performing Updates

By default, a desktop installation of Kali Linux has about 2,300 packages installed. There are regular feature and security updates for these packages. To install them, you need to run the following two-part command now and then in a terminal window. Here, `apt update` updates the package sources, and `apt full-upgrade` downloads all changed packages and installs them:

```
sudo apt update && sudo apt full-upgrade
```

If `apt update` complains that the package source signature is invalid or expired, you may need to import a new key:

```
sudo -s

wget -q -O - https://archive.kali.org/archive-key.asc | \
  apt-key add
```

While the update is being performed, texts are displayed that you can scroll through using the cursor keys. The process will not continue until you press `Q` to end the text display and thus confirm that you have read the notes.

After major updates, you can use `apt autoremove` and `apt autoclean` to delete unnecessary files (e.g., downloaded package files that are no longer needed after installation):

```
apt autoremove
apt autoclean
```

2.9.5 Installing Software

Kali Linux does contain countless security and hacking tools, but you may be missing a specific command or may want to summarize the results of your work in an editor that is not available on Kali Linux. In such cases, you can use `apt install name` to install the relevant package with the missing component. In total, there are about 50,000 packages to choose from. However, this doesn't mean that there are also as many programs. Many packages contain libraries, localization files for various languages, and so on.

Searching Packages

If you know the package name, installing new packages is very easy. It's more difficult if you do not know the package name. In that case, an internet search often helps—for example, with the additional search term "Debian package". Another useful tool is the Synaptic program, which you can install using `apt install synaptic`. Synaptic is a graphical user interface for package management with good search functions.

Command Reference

A clear reference sheet showing the most important Kali hacking tools is provided by the Kali Linux Cheat Sheet. You can download the cheat sheet in PDF form from *http://s-prs.co/v569606*.

2.9.6 Python 2

Python 2 is officially no longer supported as of early 2020. Although Python 2 packages are still available in Debian testing and in Kali Linux, many hacking tools that require Python 2 have already disappeared from the package sources. (A particularly prominent victim of the end of support for Python 2 is Zenmap, an immensely handy interface to the `nmap` command.)

In the longer term, Python 2 will disappear completely from Kali Linux. Of course, this will also affect all programs and scripts based on Python 2. As a rule, these are tools that have not been maintained for a long time; otherwise, their developers would have made the switch to Python 3 long ago.

2.9.7 Network Services and Firewall

Although Kali Linux comes with preinstalled programs for several externally accessible network services (SSH server, web server, FTP server, etc.), such services are not active without exception. This has to do with the fact that the developers wanted to maximize the security of Kali Linux and open as few potential entry gates as possible. You can start such services temporarily using `systemctl start` (valid until the next reboot) or enable them permanently via `systemctl enable` (valid from the next reboot).

So, for example, to enable the SSH server permanently, the following command is required:

```
systemctl enable --now ssh
```

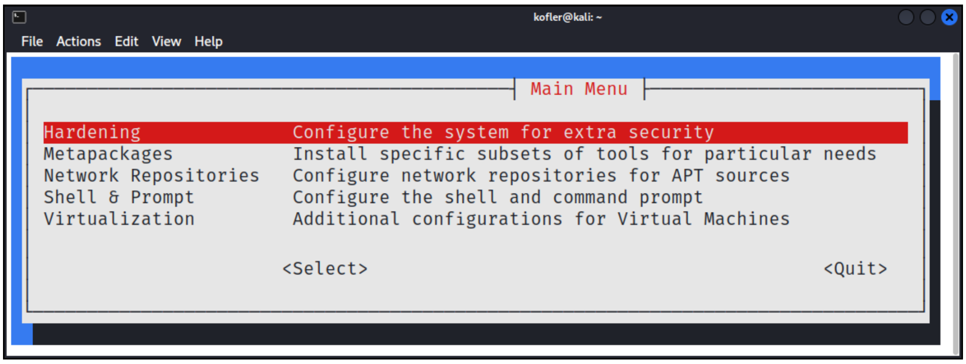
To disable the service again if necessary, you must run `systemctl disable --now`.

No Firewall

Given that no externally accessible network services are active by default, it is understandable that no firewall is active in Kali Linux. This would also interfere with working with various hacking tools.

2.9.8 kali-tweaks

The `kali-tweaks` script enables you to change some basic settings of Kali. A simple menu provides control functions (see Figure 2.15).



```
                              Main Menu

Hardening               Configure the system for extra security
Metapackages            Install specific subsets of tools for particular needs
Network Repositories    Configure network repositories for APT sources
Shell & Prompt          Configure the shell and command prompt
Virtualization          Additional configurations for Virtual Machines

                       <Select>                                   <Quit>
```

Figure 2.15 Changing Basic Settings Using kali-tweaks

The `kali-tweaks` script allows you to disable insecure protocols of SSH, SSL, and Samba; install entire groups of packages for specific hacking tasks; set up complementary package sources for add-on packages; modify the shell and shell input prompt; and install software to better support the active virtualization system. Although the operation of `kali-tweaks` is simple, the tool is intended for more advanced users.

2.9.9 Undercover Mode

Kali undercover mode is a nice gimmick. If you activate this mode, Kali Linux looks—at least at first glance—like Windows 10 (see Figure 2.16). This may come in handy if you use Kali Linux in a public place: not everyone looking over your shoulder will recognize you as a hacker.

To activate this mode, search for "undercover" in the Start menu or run `kali-undercover` in the terminal. Running the script again restores the previous appearance of your Linux desktop.

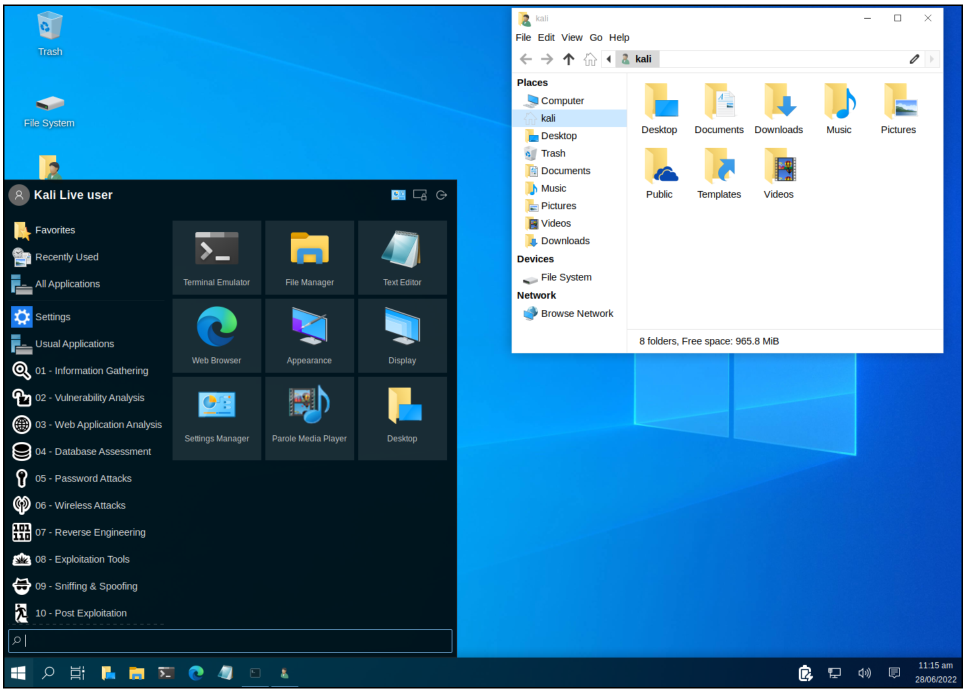


Figure 2.16 Undercover Mode: Kali Linux Bears Distant Resemblance to Windows

2.9.10 PowerShell

What the terminal is for Linux freaks, the PowerShell is for Windows fans. Since 2016, there is also a PowerShell variant for Linux, which can be quickly installed in Kali Linux. Then you can start the PowerShell vis the `pwsh` command. In PowerShell, you can combine Windows and Linux commands:

```
sudo apt install -y powershell

pwsh
  PowerShell 7.1.3
  Copyright (c) Microsoft Corporation.

PS /root> Update-Help

PS /root> Get-Process | wc -l
  152
```

For x86 Installations Only

The `powershell` package is currently only available for x86 platforms. If you run Kali Linux on the Raspberry Pi or on a virtual machine on a Mac with an M1/M2 processor, you can’t use PowerShell.

Chapter 3

Setting Up the Learning Environment: Metasploitable, Juice Shop

Hacking has to be learned. It isn't useful to try the tools from Kali Linux or other tools for penetration testing on real targets. Before scanning your company server for security gaps, you should first gain some experience in using the various tools.

Virtual machines in which security gaps are deliberately built in are very well suited as a test and learning environment. You can download the image files of such virtual machines from the internet and run them in a virtualization program. Then you can try to find the configuration errors or vulnerabilities—that is, try to "crack" the virtual machine. This is not only educational, but even fun!

The best known example of such learning environments is Metasploitable. Currently there are three Metasploitable versions available:

- Metasploitable 2 (based on Ubuntu 8.06)
- Metasploitable 3, Linux variant (based on Ubuntu 14.04)
- Metasploitable 3, Windows variant (based on Windows Server 2008)

Metasploitable versus Metasploit

The two names are similar, but they refer to completely different things: Metasploit is a framework for finding and exploiting security gaps (exploits; see Chapter 4, Section 4.9). Metasploitable, on the other hand, is a learning environment and was created by the Metasploit developers as such. However, this doesn't mean that you can't attack Metasploitable with other tools.

Besides Metasploitable, there are of course other popular test environments, such as the following:

- Badstore
- bWAPP (buggy web app)
- Damn Vulnerable Web App (DVWA)
- Juice Shop
- Web Security Dojo
- WebGoat 8
- GOAD (test environment for Active Directory hacking; see *http://s-prs.co/v569607*)

Without any installation at all, you can try your first hacking experiments on intentionally insecure websites, such as those at *http://zero.webappsecurity.com* or *https://hackxor.net*.

On the internet, you can find a lot of other virtual machines that invite hacking. Some of them, like Metasploitable, were created as training or education bases, while others were used for hacking competitions. You can find a large selection of such virtual machines at *https://www.vulnhub.com*.

In this chapter, we'll focus on the three Metasploitable variants as well as Juice Shop. We'll show you how to properly set up the test environment and give you some tips on how to launch the attack. (Don't worry—we won't give away too much. If you fail in your own attempts to attack, you'll find instructions galore on the internet.)

3.1 Honeypots

As useful as a system with known vulnerabilities is in a local network for learning purposes, it's dangerous to make such virtual machines freely available on the web. It wouldn't take long for someone to discover the vulnerabilities, take over the virtual machine, and exploit it for their own purposes—in the most innocuous case, to further increase the ubiquitous avalanche of spam.

And yet sometimes that is exactly what happens: that is, a deliberately insecure system is put on the web. Such a system is then referred to as a *honeypot*. Its task is to study the behavior of hackers: How long does it take to take over the machine? Who are the hackers? How do you proceed?

Honeypots can be the basis of scientific studies, or they can be deliberately laid traps for hackers to fall into. Of course, it's essential to be smarter than the attacker. On the one hand, the system must appear so real that the attacker doesn't immediately recognize it as a trap; on the other hand, it's vital to avoid the attacker completely taking over the system too quickly and covering his traces before the honeypot operator has a chance to draw any conclusions.

3.2 Metasploitable 2

Metasploitable 2 is probably the most widely used hacking learning environment. It's available free of charge, is very easy to install, and is used in countless trainings and courses.

The biggest drawback of Metasploitable 2 is that the test system is very old. Metasploitable 2 was released in 2012. It is based on Ubuntu 8.06, which is a Linux distribution that is about 15 years old. Metasploitable 2 thus fulfills its purpose, of course, in that it is indeed full of security gaps: those that were configured intentionally, and those that became known only later and for which there are now no updates with bug fixes. This makes Metasploitable suitable for trying out security tools. However, the extent and nature of the security vulnerabilities are unrealistic. In addition, many security problems cannot be fixed because there are neither package sources nor updates for the ancient Ubuntu system. In other words: you can learn hacking using Metasploitable 2, but you can't learn how to secure Linux computers with it.

3.2.1 Installation in VirtualBox

Metasploitable 2 is available for download as a ZIP file from *https://sourceforge.net/projects/metasploitable*.

The archive file contains five files that together form a VMware virtual machine. However, this does not mean that you need VMware to run Metasploitable 2. To set up the virtual machine in VirtualBox, you need to create a new machine in VirtualBox with the following key data:

- **Type**
  Select **Linux**.
- **Version**
  Select **Linux 2.6/3.x/4.x (32 Bit)**.
- **Memory size**
  256 MB
- **Disk**
  Use existing hard disk; then select **Metasploitable.vmdk** as the virtual hard disk. This file, which is about 2 GB in size, can be found in the ZIP file mentioned earlier. The *.vmdk file is the only file you need to set up the machine in VirtualBox. (If you like to keep your computer well ordered, you should move the file beforehand to the new directory that has been set up for the virtual machine by VirtualBox.)
- **CPU**
  Before you start the virtual machine for the first time, you must select the **Enable PAE/NX** option in the settings via **System • Processor**. Otherwise, the virtual machine won't boot, issuing the following error message: **This kernel requires the following features not present on the CPU: 0:6**.

3.2.2 Network Settings

Metasploitable must not be accessible from the internet, of course. This is usually not the case with an installation on a desktop virtualization system anyway.

In the previous chapter, we recommended using the **Network Bridge** setting when configuring the virtual machine for Kali Linux. This allows you to use Kali Linux to analyze other computers on the local network. To be able to attack Metasploitable from Kali

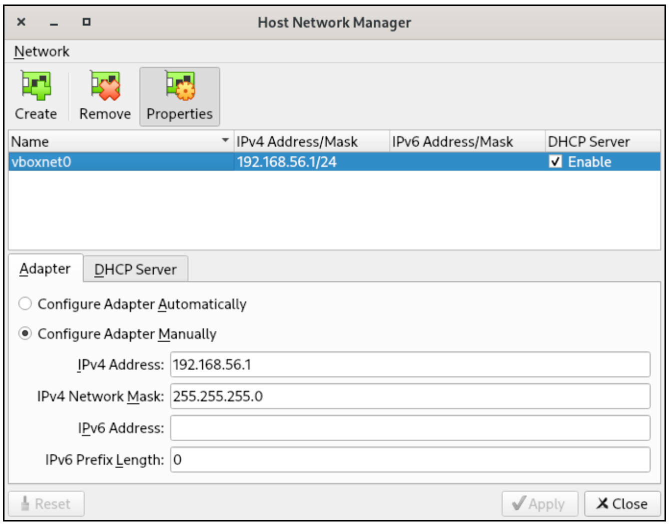
Linux, you must also change its network configuration to **Network Bridge**. The **Adapter 1** button in the status bar of the VirtualBox window allows this change even during operation.

3.2.3 Host-Only Network

Note that in larger local networks (e.g., in a company or at a university/school), Metasploitable can also be attacked from computers *within* the network, and in this respect it represents a security risk!

Ideally, you should therefore use a host-only adapter for both Kali Linux and Metasploitable. This way, you can completely disconnect Kali Linux and Metasploitable from both the internet and the local network. Such a configuration is useful, for example, for teaching or training purposes. However, the configuration has the disadvantage that you cannot even download updates or new packages in the virtual machines.

Before you can use a host-only adapter, you must set up an appropriate network. To do this, in the main window of VirtualBox (i.e., the window where all virtual machines are listed), you need to select the **File • Host-only Network Manager**. In a new window, you can then set up a new private network, freely choosing the IP address range and the configuration of the assigned DHCP server (see Figure 3.1). By default, the first adapter is named vboxnet0 and gets the address range 192.168.56.1/24.



Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server
vboxnet0	192.168.56.1/24		✓ Enable

( ) Configure Adapter Automatically
(●) Configure Adapter Manually

IPv4 Address:	192.168.56.1
IPv4 Network Mask:	255.255.255.0
IPv6 Address:	
IPv6 Prefix Length:	0

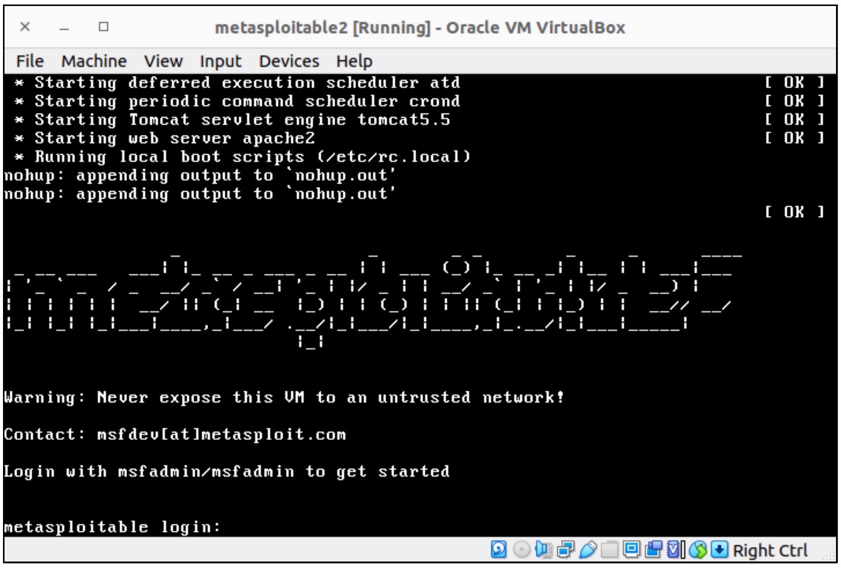
Figure 3.1 Configuring Host-Only Network for VirtualBox

Then, for each of the Kali Linux and Metasploitable virtual machines, you want to change the network adapter configuration to **Connected to = Host-only Adapter** and **Name = vboxnet0**.

3.2.4 Using Metasploitable 2

After starting the virtual machine, Metasploitable 2 presents itself in a plain text interface (see Figure 3.2). Metasploitable 2 is configured as a server in the best Linux manner. There is no graphical user interface. You can use the following data to log in:

- **Login**
  msfadmin
- **Password**
  msfadmin


metasploitable2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
 * Starting deferred execution scheduler atd                          [ OK ]
 * Starting periodic command scheduler crond                          [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                           [ OK ]
 * Starting web server apache2                                        [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                      [ OK ]

                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login:
```

Figure 3.2 Metasploitable 2 in a virtual machine

The msfadmin user doesn't directly have any root privileges. However, you can use `sudo -s` to switch to root mode by specifying msfadmin again, and then you'll have unrestricted administration rights.

Metasploitable 2 uses the US keyboard layout by default. There is no easy way to change this. The `loadkeys de-latin1` command doesn't work because the file for the keyboard layouts are not installed. And a subsequent installation is impossible because Metasploitable 2 uses Ubuntu 8.06 as a base. There have been no active package sources for this ancient Linux version for years.

To bypass any keyboard problems, it's best to do administration work in Metasploitable 2 via SSH. For this purpose, you need to run `ssh msfadmin@metasploitable` from another machine or virtual machine on the same network. If the host name metasploitable is unknown, you can find the IP address of the eth0 network interface in the Metasploitable virtual machine via `ip addr show` and then establish the SSH connection using `ssh msfadmin@n.n.n.n`.

However, in Kali Linux and most other current Linux distributions, the connection attempt fails with the following error message:

```
ssh msfadmin@192.168.56.101
 Unable to negotiate with 192.168.56.101 port 22: no
 matching key type found. Their offer: ssh-rsa,ssh-dss
```

This error message is due to the fact that the cryptographic functions used by the SSH server in Metasploitable are so outdated that they are simply rejected by current SSH clients. But you can change that. To do so, add the following line to the `/etc/ssh_config` file in Kali Linux or on your host machine (not in the virtual machine with Metasploitable!):

```
# in /etc/ssh_config
...
# accept obsolete signature algorithms
HostKeyAlgorithms +ssh-rsa,ssh-dss
```

If you now run `ssh msfadmin@...` again, the connection will be established.

3.2.5 Hacking Metasploitable 2

After this preparatory work, the fun can start. Log into Kali Linux, for example, and try to hack Metasploitable 2 from there. Of course, you must assume that you do not know the msfadmin account and its password. In the following examples, we assume that the IP address of Metasploitable is 192.168.56.101 and that Kali and Metasploitable are on the same network.

If the IP address of Metasploitable is not known, the attacker first performs an address scan on the local network, very quickly, for example, with `arp-scan`. Provided that only Kali Linux and Metasploitable are on the same network, the scan will return only three addresses: that of the VirtualBox host (often `n.n.n.1`), the address of Kali Linux you know, and the address of Metasploitable you are looking for.

One possible way into the system starts with a port scan that determines which ports are open:

```
nmap -p0-65535 192.168.56.101

  Nmap scan report for 192.168.56.101
  Not shown: 65506 closed ports
  PORT      STATE SERVICE
  21/tcp    open  ftp
  22/tcp    open  ssh
  23/tcp open telnet
```