Kali Linux Revealed

Mastering the Penetration Testing Distribution

Kali Linux Revealed

Mastering the Penetration Testing Distribution

by Raphaël Hertzog, Jim O'Gorman, and Mati Aharoni

OFFSEC PRESS

Kali Linux Revealed

Some sections of this book borrow content from the "Debian Administrator's Handbook, Debian Jessie from Discovery to Mastery" written by Raphaël Hertzog and Roland Mas, which is available here:

➨ `https://debian-handbook.info/browse/stable/`

For the purpose of the CC-BY-SA license, Kali Linux Revealed is an Adaptation of the Debian Administrator's Handbook.

ISBN: 978-0-9976156-0-9 (paperback)

```
Offsec Press
19701 Bethel Church Road, #103-253
Cornelius NC 28031
USA
www.offensive-security.com
```

Library of Congress Control Number: 2017905895

Printed in the United States of America.

Table of Contents

Preface

You have no idea how good you have it.

In 1998, I was an up-and-coming hacker, co-founding one of the earliest professional white hat hacking teams. We were kids, really, with dream jobs, paid to break into some of the most secure computer systems, networks, and buildings on the planet.

It sounds pretty sexy, but in reality, we spent most of our time hovering over a keyboard, armed with the digital tools of our trade. We wielded a sordid collection of programs, designed to map networks and locate targets; then scan, exploit, and pivot through them. In some cases, one of us (often Jim Chapple) would write custom tools to do wicked things like scan a Class A network (something no other tool could do, at the time), but most often we would use or modify tools written by the hacker community. In those pre-Google days, we frequented BugTraq, AstaLaVista, Packet Storm, w00w00, SecurityFocus, X-Force, and other resources to conduct research and build our arsenal.

Since we had limited time on each gig, we had to move quickly. That meant we couldn't spend a lot of time fiddling with tools. It meant we had to learn the core tools inside and out, and keep the ancillary ones on tap, just in case. It meant we had to have our tools well-organized, documented, and tested so there would be few surprises in the field. After all, if we didn't get in, we lost face with our clients and they would take our recommendations far less seriously.

Because of this, I spent a lot of time cataloging tools. When a tool was released or updated, I'd go through a routine. I had to figure out if it would run on the attack platform (some didn't), and whether it was worthwhile (some weren't); I had to update any scripts that relied on it, document it, and test it, including carrying over any changes made to the previous version.

Then, I would shake out all the tools and put them in directories based on their purpose during an assessment. I'd write wrapper scripts for certain tools, chain some tools together, and correlate all that into a separate CD that we could take into sensitive areas, when customers wouldn't let us take in attack machines or remove media from their labs.

This process was painful, but it was necessary. We knew that we had the ability to break into any network—if we applied our skills and expertise properly, stayed organized, and worked efficiently. Although remaining undefeated was a motivator, it was about providing a service to clients who *needed* us to break into networks, so they could plug gaps and move money toward critical-but-neglected information security programs.

We spent years sharpening our skills and expertise but we wouldn't have been successful without organization and efficiency. We would have failed if we couldn't put our hands on the proper tool when needed.

That's why I spent so much time researching, documenting, testing, and cataloging tools, and at the turn of the 21st Century, it was quickly becoming an overwhelming, full-time job. Thanks to the Internet, the worldwide attack surface exploded and the variety and number of attack tools increased exponentially, as did the workload required to maintain them.

Starting in 2004, the Internet exploded not only as a foundation for business but also as a social platform. Computers were affordable, more consumer-friendly and ubiquitous. Storage technology expanded from megabytes to gigabytes. Ethernet jumped from hundreds of kilobits to tens of megabits per second, and Internet connections were faster and cheaper than ever before. E-commerce was on the rise, social media sites like Facebook (2004) and Twitter (2006) came online and Google (1998) had matured to the point that anyone (including criminals) could find just about anything online.

Research became critical for teams like ours because we had to keep up with new attacks and toolsets. We responded to more computer crimes, and forensic work demanded that we tread lightly as we mucked through potential evidence. The concept of a *live CD* meant that we could perform live forensics on a compromised machine without compromising evidence.

Now our little team had to manage attack tools, forensic tools, and a sensitive area tool distribution; we had to keep up with all the latest attack and exploit methodologies; and we had to, you know, actually do what we were paid for—penetration tests, which were in high demand. Things were spinning out of control, and before long, we were spending less time in battle and much more time researching, sharpening our tools, and planning.

We were not alone in this struggle. In 2004, Mati "Muts" Aharoni, a hacker and security professional released "WHoppiX" (White Hat Knoppix), a live Linux CD that he billed as "the ultimate pen testing live CD," It included "all the exploits from SecurityFocus, Packet Storm and k-otik, Metasploit Framework 2.2, and much, much more."

I remember downloading WHoppiX and thinking it was a great thing to have around. I downloaded other live CDs, thinking that if I were ever in a real pinch, live CDs could save my bacon in the field. But I wasn't about to rely on WHoppiX or any other CD for real work. I didn't trust any of them to fulfill the majority of my needs; none of them felt right for my workflow; they were not full, installable distributions; and the moment I downloaded them they were out of date. An aged toolset is the kiss of death in our industry.

I simply added these CD images, despite their relatively massive size, to our arsenal and kept up the painful process of maintaining our "real" toolkit.

But despite my personal opinions at the time, and perhaps despite Muts' expectations, WHoppiX and its descendants had a seismic impact on his life, our industry, and our community.

In 2005, WHoppiX evolved into WHAX, with an expanded and updated toolset, based on "the more modular SLAX (Slackware) live CD." Muts and a growing team of volunteers from the hacker community seemed to realize that no matter how insightful they were, they could never anticipate all the growth and fluctuation of our industry and that users of their CD would have varied needs in the field. It was obvious that Muts and his team were actually using WHAX in the field, and they seemed dedicated to making it work. This was encouraging to me.

In 2006, Muts, Max Moser, and their teams consolidated Auditor Security Linux and WHAX into a single distribution called BackTrack. Still based on SLAX, BackTrack continued to grow, adding more tools, more frameworks, extended language support, extensive wireless support, a menu structure catering to both novice and pro users, and a heavily modified kernel. BackTrack became the leading security distribution, but many like me still used it as a backup for their "real tools."

By early 2009, Muts and his team had extended BackTrack significantly to BackTrack 4. Now a full-time job for Muts, BackTrack was no longer a live CD but a full-blown Ubuntu-based distribution leveraging the Ubuntu software repositories. The shift marked a serious evolution: BackTrack 4 had an update mechanism. In Muts' own words: "When syncing with our BackTrack repositories, you will regularly get security tool updates soon after they are released."

This was a turning point. The BackTrack team had tuned into the struggles facing pen testers, forensic analysts and others working in our industry. Their efforts would save us countless hours and provide a firm foundation, allowing us to get back into the fight and spend more time doing the important (and fun) stuff. As a result, the community responded by flocking to the forums and wiki; and by pitching in on the dev team. BackTrack was truly a community effort, with Muts still leading the charge.

BackTrack 4 had finally become an industrial-strength platform and I, and others like me, breathed a sigh of relief. We knew firsthand the "pain and sufferance" Muts and his team were bearing, because we had been there. As a result, many of us began using BackTrack as a primary foundation for our work. Yes, we still fiddled with tools, wrote our own code, and developed our own exploits and techniques; and we researched and experimented; but we did not spend all our time collecting, updating, validating, and organizing tools.

BackTrack 4 R1 and R2 were further revisions in 2010, leading to the ground-up rebuild of BackTrack 5 in 2011. Still based on Ubuntu, and picking up steam with every release, BackTrack was now a massive project that required a heroic volunteer and community effort but also funding. Muts launched Offensive Security (in 2006) not only to provide world-class training and penetration testing services but also to provide a vehicle to keep BackTrack development rolling, and ensure that BackTrack remained open-source and free to use.

BackTrack continued to grow and improve through 2012 (with R1, R2, and R3), maintaining an Ubuntu core and adding hundreds of new tools, including physical and hardware exploitation tools, VMware support, countless wireless and hardware drivers, and a multitude of stability improvements and bug fixes. However, after the release of R3, BackTrack development went relatively, and somewhat mysteriously, quiet.

There was some speculation in the industry. Some thought that BackTrack was getting "bought out", selling its soul to a faceless evil corporate overlord for a massive payout. Offensive Security was growing into one of the most respected training companies and a thought leader in our industry, and some speculated that its success had gobbled up and sidelined the key BackTrack developers. However, nothing could be farther from the truth.

In 2013, Kali Linux 1.0 was released. From the release notes: "After a year of silent development, Offensive Security is proud to announce the release and public availability of Kali Linux, the most advanced, robust, and stable penetration-testing distribution to date. Kali is a more mature, secure, and enterprise-ready version of BackTrack."

Kali Linux was not a mere rebranding of BackTrack. Sporting more than 600 completely repackaged tools, it was clearly an amazing toolset, but there was still more to it than that. Kali had been built, from the ground up, on a Debian core. To the uninformed, this might not seem like a big deal. But the ripple effects were staggering. Thanks to a massive repackaging effort, Kali users could download the source for every single tool; they could modify and rebuild a tool as needed, with only a few keystrokes. Unlike other mainstream operating systems of the day, Kali Linux synchronized with the Debian repositories four times a day, which meant Kali users could get wickedly current package updates and security fixes. Kali developers threw themselves into the fray, packaging and maintaining upstream versions of many tools so that users were constantly kept on the bleeding edge. Thanks to its Debian roots, Kali's users could bootstrap an installation or ISO directly from the repositories, which opened the door for completely customized Kali installations or massive enterprise deployments, which could be further automated and customized with preseed files. To complete the customization trifecta, Kali Users could modify the desktop environment, alter menus, change icons, and even replace windowing environments. A massive ARM development push opened the door for installation of Kali Linux on a wide range of hardware platforms including access points, single-board computers (Raspberry Pi, ODROID, BeagleBone, and CubieBoard, for example), and ARM-based Chromebook computers. And last but certainly not least, Kali Linux sported seamless minor and major upgrades, which meant devotees would never have to re-install customized Kali Linux setups.

The community took notice. In the first five days, 90,000 of us downloaded Kali 1.0.

This was just the beginning. In 2015, Kali 2.0 was released, followed by the 2016 rolling releases. In summary, "If Kali 1.0 was focused on building a solid infrastructure, then Kali 2.0 is focused on overhauling the user experience and maintaining updated packages and tool repositories."

The current version of Kali Linux is a rolling distribution, which marks the end of discrete versions. Now, users are up to date continuously and receive updates and patches as they are created. Core tools are updated more frequently thanks to an upstream version tagging system, groundbreaking accessibility improvements for the visually impaired have been implemented, and the Linux kernels are updated and patched to continue wireless 802.11 injection support. Software Defined Radio (SDR) and Near-Field Communication (NFC) tools add support for new fields of security testing. Full Linux encrypted disk installation and emergency self-destruct options are available,

thanks to LVM and LUKS respectively, USB persistence options have been added, allowing USB-based Kali installs to maintain changes between reboots, whether the USB drive is encrypted or not. Finally, the latest revisions of Kali opened the door for NetHunter, an open-source world-class operating system running on mobile devices based on Kali Linux and Android.

Kali Linux has evolved not only into the information security professional's platform of choice, but truly into an industrial-grade, world-class, mature, secure, and enterprise-ready operating system distribution.

Through the decade-long development process, Muts and his team, along with the tireless dedication of countless volunteers from the hacker community, have taken on the burden of streamlining and organizing our work environment, freeing us from much of the drudgery of our work and providing a secure and reliable foundation, allowing us to concentrate on driving the industry forward to the end goal of securing our digital world.

And interestingly, but not surprisingly, an amazing community has built up around Kali Linux. Each and every month, three to four hundred thousand of us download a version of Kali. We come together on the Kali forums, some forty-thousand strong, and three to four hundred of us at a time can be found on the Kali IRC channel. We gather at conferences and attend Kali Dojos to learn how to best leverage Kali from the developers themselves.

Kali Linux has changed the world of information security for the better, and Muts and his team have saved each of us countless hours of toil and frustration, allowing us to spend more time and energy driving the industry forward, together.

But despite its amazing acceptance, support, and popularity, Kali has never released an official manual. Well, now that has changed. I'm thrilled to have come alongside the Kali development team and specifically Mati Aharoni, Raphaël Hertzog, Devon Kearns, and Jim O'Gorman to offer this, the first in perhaps a series of official publications focused on Kali Linux. In this book, we will focus on the Kali Linux platform itself, and help you understand and maximize the usage of Kali from the ground up. We won't yet delve into the arsenal of tools contained in Kali Linux, but whether you're a veteran or an absolute n00b, this is the best place to start, if you're ready to dig in and get serious with Kali Linux. Regardless of how long you've been at the game, your decision to read this book connects you to the growing Kali Linux community, one of the oldest, largest, most active, and most vibrant in our industry.

On behalf of Muts and the rest of the amazing Kali team, congratulations on taking the first step to mastering Kali Linux!

Johnny Long

February 2017

Foreword

The sixteen high-end laptops ordered for your pentesting team just arrived, and you have been tasked to set them up—for tomorrow's offsite engagement. You install Kali and boot up one of the laptops only to find that it is barely usable. Despite Kali's cutting-edge kernel, the network cards and mouse aren't working, and the hefty NVIDIA graphics card and GPU are staring at you blankly, because they lack properly installed drivers. You sigh.

In Kali *Live mode*, you quickly type `lspci` into a console, then squint. You scroll through the hardware listing: "PCI bridge, USB controller, SATA controller. Aha! Ethernet and Network controllers." A quick Google search for their respective model numbers, cross referenced with the Kali kernel version, reveals that these cutting-edge drivers haven't reached the mainline kernel yet.

But all is not lost. A plan is slowly formulating in your head, and you thank the heavens for the *Kali Linux Revealed* book that you picked up a couple of weeks ago. You could use the Kali Live-Build system to create a custom Kali ISO, which would have the needed drivers baked into the installation media. In addition, you could include the NVIDIA graphics drivers as well as the CUDA libraries needed to get that beast of a GPU to talk nicely to hashcat, and have it purr while cracking password hashes at blistering speeds. Heck, you could even throw in a custom wallpaper with a Microsoft Logo on it, to taunt your team at work.

Since the hardware profiles for your installations are identical, you add a preseeded boot option to the ISO, so that your team can boot off a USB stick and have Kali installed with no user interaction—the installation takes care of itself, full disk encryption and all.

Perfect! You can now generate an updated version of Kali on demand, specifically designed and optimized for your hardware. You saved the day. Mission complete!

With the deluge of hardware hitting the market, this scenario is becoming more common for those of us who venture away from mainstream operating systems, in search of something leaner, meaner, or more suitable to our work and style.

This is especially applicable to those attracted to the security field, whether it be an alluring hobby, fascination, or line of work. As newcomers, they often find themselves stumped by the environment or the operating system. For many newcomers Kali is their first introduction to Linux.

We recognized this shift in our user base a couple of years back, and figured that we could help our community by creating a structured, introductory book that would guide users into the world

of security, while giving them all the Linux sophistication they would need to get started. And so, the Kali book was born—now available free over the Internet for the benefit of anyone interested in entering the field of security through Kali Linux.

As the book started taking shape, however, we quickly realized that there was untapped potential. This would be a great opportunity to go further than an introductory Kali Linux book and explore some of the more interesting and little-known features. Hence, the name of the book: *Kali Linux Revealed*.

By the end, we were chuffed with the result. The book answered all our requirements and I'm proud to say it exceeded our expectations. We came to the realization that we had inadvertently enlarged the book's potential user base. It was no longer intended only for newcomers to the security field, but also included great information for experienced penetration testers who needed to improve and polish their control of Kali Linux—allowing them to unlock the full potential of our distribution. Whether they were fielding a single machine or thousands across an enterprise, making minor configuration changes or completely customizing down to the kernel level, building their own repositories, touching the surface or delving deep into the amazing Debian package management system, *Kali Linux Revealed* provides the roadmap.

With your map in hand, on behalf of myself and the entire Kali Linux team, I wish you an exciting, fun, fruitful, and "revealing" journey!

Muts, February 2017

Introduction

Kali Linux is the world's most powerful and popular penetration testing platform, used by security professionals in a wide range of specializations, including penetration testing, forensics, reverse engineering, and vulnerability assessment. It is the culmination of years of refinement and the result of a continuous evolution of the platform, from WHoppiX to WHAX, to BackTrack, and now to a complete penetration testing framework leveraging many features of Debian GNU/Linux and the vibrant open source community worldwide.

Kali Linux has not been built to be a simple collection of tools, but rather a flexible framework that professional penetration testers, security enthusiasts, students, and amateurs can customize to fit their specific needs.

Why This Book?

Kali Linux is not merely a collection of various information security tools that are installed on a standard Debian base and preconfigured to get you up and running right away. To get the most out of Kali, it is important to have a thorough understanding of its powerful Debian GNU/Linux underpinnings (which support all those great tools) and learning how you can put them to use in your environment.

Although Kali is decidedly multi-purpose, it is primarily designed to aid in penetration testing. The objective of this book is not only to help you feel at home when you use Kali Linux, but also to help improve your understanding and streamline your experience so that when you are engaged in a penetration test and time is of the essence, you won't need to worry about losing precious minutes to install new software or enable a new network service. In this book, we will introduce you first to Linux, then we will dive deeper as we introduce you to the nuances specific to Kali Linux so you know exactly what is going on under the hood.

This is invaluable knowledge to have, particularly when you are trying to work under tight time constraints. It is not uncommon to require this depth of knowledge when you are getting set up, troubleshooting a problem, struggling to bend a tool to your will, parsing output from a tool, or leveraging Kali in a larger-scale environment.

Is This Book for You?

If you are eager to dive into the intellectually rich and incredibly fascinating field of information security, and have rightfully selected Kali Linux as a primary platform, then this book will help you in that journey. This book is written to help first-time Linux users, as well as current Kali users seeking to deepen their knowledge about the underpinnings of Kali, as well as those who have used Kali for years but who are looking to formalize their learning, expand their use of Kali, and fill in gaps in their knowledge.

In addition, this book can serve as a roadmap, technical reference, and study guide for those pursuing the Kali Linux Certified Professional certification.

General Approach and Book Structure

This book has been designed so that you can put your hands on Kali Linux right from the start. You don't have to read half of the book to get started. Every topic is covered in a very pragmatic manner, and the book is packed with samples and screenshots to help make the explanations more concrete.

In chapter 1, "About Kali Linux" [page 2], we define some basic terminology and explain the purpose of Kali Linux. In chapter 2, "Getting Started with Kali Linux" [page 14], we guide you step-by-step from the download of the ISO image to getting Kali Linux running on your computer. Next comes chapter 3, "Linux Fundamentals" [page 48] which supplies the basic knowledge that you need to know about any Linux system, such as its architecture, installation process, file system hierarchy, permissions, and more.

At this point, you have been using Kali Linux as live system for a while. With chapter 4, "Installing Kali Linux" [page 66] you will learn how to make a permanent Kali Linux installation (on your hard disk) and with chapter 5, "Configuring Kali Linux" [page 104] how to tweak it to your liking. As a regular Kali user, it is time to get familiar with the important resources available to Kali users: chapter 6, "Helping Yourself and Getting Help" [page 124] gives you the keys to deal with the unexpected problems that you will likely face.

With the basics well covered, the rest of the book dives into more advanced topics: chapter 7, "Securing and Monitoring Kali Linux" [page 150] gives you tips to ensure that your Kali Linux installation meets your security requirements. Next, chapter 8, "Debian Package Management" [page 170] explains how to leverage the full potential of the Debian packaging ecosystem. And in chapter 9, "Advanced Usage" [page 222], you learn how to create a fully customized Kali Linux ISO image. All those topics are even more relevant when you deploy Kali Linux at scale in an enterprise as documented in chapter 10, "Kali Linux in the Enterprise" [page 252].

The last chapter, chapter 11, "Introduction to Security Assessments" [page 280], makes the link between everything that you have learned in this book and the day-to-day work of security professionals.

Acknowledgments of Raphaël Hertzog

I would like to thank Mati Aharoni: in 2012, he got in touch with me because I was one out of dozens of Debian consultants and he wanted to build a successor to BackTrack that would be based on Debian. That is how I started to work on Kali Linux, and ever since I have enjoyed my journey in the Kali world.

Over the years, Kali Linux got closer to Debian GNU/Linux, notably with the switch to Kali Rolling, based on Debian Testing. Now most of my work, be it on Kali or on Debian, provides benefits to the entire Debian ecosystem. And this is exactly what keeps me so motivated to continue, day after day, month after month, year after year.

Working on this book is also a great opportunity that Mati offered me. It is not the same kind of work but it is equally rewarding to be able to help people and share with them my expertise of the Debian/Kali operating system. Building on my experience with the *Debian Administrator's Handbook*, I hope that my explanations will help you to get started in the fast-moving world of computer security.

I would also like to thank all the Offensive Security persons who were involved in the book: Jim O'Gorman (co-author of some chapters), Devon Kearns (reviewer), Ron Henry (technical editor), Joe Steinbach and Tony Cruse (project managers). And thank you to Johnny Long who joined to write the preface but ended up reviewing the whole book.

Acknowledgments of Jim O'Gorman

I would like to thank everyone involved in this project for their contributions, of which mine were only a small part. This book, much like Kali Linux itself was a collaborative project of many hands making light work. Special thanks to Raphaël, Devon, Mati, Johnny, and Ron for taking on the lion's share of the effort. Without them, this book would not have come together.

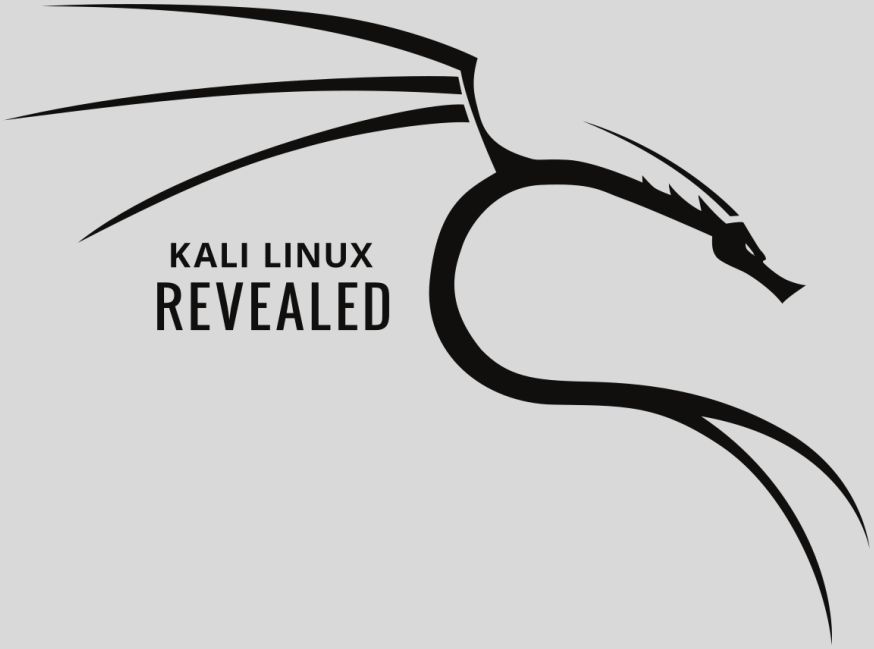
Acknowledgments of Mati Aharoni

It has been a few years since Kali Linux was first released, and since day one, I have always dreamt of publishing an official book which covers the Kali operating system as a whole. It is therefore a great privilege for me to finally see such a book making it out to the public. I would like to sincerely thank everyone involved in the creation of this project—including Jim, Devon, Johnny,

and Ron. A very special thanks goes to Raphaël for doing most of the heavy lifting in this book, and bringing in his extensive expertise to our group.

Keywords

Linux distribution
Debian derivative
Purpose
Features
Policies



Chapter

1 About Kali Linux

Contents

Kali Linux[1] is an enterprise-ready security auditing Linux distribution based on Debian GNU/Linux. Kali is aimed at security professionals and IT administrators, enabling them to conduct advanced penetration testing, forensic analysis, and security auditing.

What is a Linux Distribution?

Although it is commonly used as a name for the entire operating system, Linux is just the name of the kernel, a piece of software that handles interactions between the hardware and end-user applications.

The expression *Linux distribution*, on the other hand, refers to a complete operating system built on top of the Linux kernel, usually including an installation program and many applications, which are either pre-installed or packaged in an easily installable way.

Debian GNU/Linux[2] is a leading generic Linux distribution, known for its quality and stability. Kali Linux builds on the work of the Debian project and adds over 300 special-purpose packages of its own, all related to information security, particularly the field of penetration testing.

Debian is a free software project providing multiple versions of its operating system and we often use the term *distribution* to refer to a specific version of it, for example the Debian Stable or Debian Testing distributions. The same also applies to Kali Linux—with the Kali Rolling distribution, for example.

1.1. A Bit of History

The Kali Linux project began quietly in 2012, when Offensive Security decided that they wanted to replace their venerable BackTrack Linux project, which was manually maintained, with something that could become a genuine Debian derivative[3], complete with all of the required infrastructure and improved packaging techniques. The decision was made to build Kali on top of the Debian distribution because it is well known for its quality, stability, and wide selection of available software. That is why I (Raphaël) got involved in this project, as a Debian consultant.

The first release (version 1.0) happened one year later, in March 2013, and was based on Debian 7 "Wheezy", Debian's stable distribution at the time. In that first year of development, we packaged hundreds of pen-testing-related applications and built the infrastructure. Even though the number of applications is significant, the application list has been meticulously curated, dropping applications that no longer worked or that duplicated features already available in better programs.

During the two years following version 1.0, Kali released many incremental updates, expanding the range of available applications and improving hardware support, thanks to newer kernel releases. With some investment in continuous integration, we ensured that all important packages

[1] `https://www.kali.org`
[2] `https://www.debian.org`
[3] `https://wiki.debian.org/Derivatives/Census`

were kept in an installable state and that customized live images (a hallmark of the distribution) could always be created.

In 2015, when Debian 8 "Jessie" came out, we worked to rebase Kali Linux on top of it. While Kali Linux 1.x avoided the GNOME Shell (relying on GNOME Fallback instead), in this version we decided to embrace and enhance it: we added some GNOME Shell extensions to acquire missing features, most notably the Applications menu. The result of that work became Kali Linux 2.0, published in August 2015.

GNOME is Kali Linux's Default Desktop Environment

A desktop environment is a collection of graphical applications that share a common graphical toolkit and that are meant to be used together on user workstations. Desktop environments are generally not used in servers. They usually provide an application launcher, a file manager, a web browser, an email client, an office suite, etc.

GNOME[4] is one of the most popular desktop environments (together with KDE[5], Xfce[6], LXDE[7], MATE[8]) and is installed on the main ISO images provided by Kali Linux. If you dislike GNOME, it is easy to build a custom ISO image with the desktop environment of your choosing. Instructions to do so are covered later in this book in chapter 9, "Advanced Usage" [page 222].

In parallel, we increased our efforts to ensure that Kali Linux always has the latest version of all pen-testing applications. Unfortunately, that goal was a bit at odds with the use of Debian Stable as a base for the distribution, because it required us to backport many packages. This is due to the fact that Debian Stable puts a priority on the stability of the software, often causing a long delay from the release of an upstream update to when it is integrated into the distribution. Given our investment in continuous integration, it was quite a natural move to rebase Kali Linux on top of Debian Testing so that we could benefit from the latest version of all Debian packages as soon as they were available. Debian Testing has a much more aggressive update cycle, which is more compatible with the philosophy of Kali Linux.

This is, in essence, the concept of Kali Rolling. While the rolling distribution has been available for quite a while, Kali 2016.1 was the first release to officially embrace the rolling nature of that distribution: when you install the latest Kali release, your system actually tracks the Kali Rolling distribution and *every single day you get new updates*. In the past, Kali releases were snapshots of the underlying Debian distribution with Kali-specific packages injected into it.

A rolling distribution has many benefits but it also comes with multiple challenges, both for those of us who are building the distribution and for the users who have to cope with a never-ending flow of updates and sometimes backwards-incompatible changes. This book aims to give you the knowledge required to deal with everything you may encounter while managing your Kali Linux installation.

[4] `https://www.gnome.org`
[5] `https://www.kde.org`
[6] `http://www.xfce.org`
[7] `http://lxde.org`
[8] `http://mate-desktop.org`

1.2. Relationship with Debian

The Kali Linux distribution is based on Debian Testing[9]. Therefore, most of the packages available in Kali Linux come straight from this Debian repository.

While Kali Linux relies heavily on Debian, it is also entirely independent in the sense that we have our own infrastructure and retain the freedom to make any changes we want.

1.2.1. The Flow of Packages

On the Debian side, the contributors are working every day on updating packages and uploading them to the Debian Unstable distribution. From there, packages migrate to the Debian Testing distribution once the most troublesome bugs have been taken out. The migration process also ensures that no dependencies are broken in Debian Testing. The goal is that Testing is always in a usable (or even releasable!) state.

Debian Testing's goals align quite well with those of Kali Linux so we picked it as the base. To add the Kali-specific packages in the distribution, we follow a two-step process.

First, we take Debian Testing and force-inject our own Kali packages (located in our *kali-dev-only* repository) to build the *kali-dev* repository. This repository will break from time to time: for instance, our Kali-specific packages might not be installable until they have been recompiled against newer libraries. In other situations, packages that we have forked might also have to be updated, either to become installable again, or to fix the installability of another package that depends on a newer version of the forked package. In any case, *kali-dev* is not for end-users.

kali-rolling is the distribution that Kali Linux users are expected to track and is built out of *kali-dev* in the same way that Debian Testing is built out of Debian Unstable. Packages migrate only when all dependencies can be satisfied in the target distribution.

1.2.2. Managing the Difference with Debian

As a design decision, we try to minimize the number of forked packages as much as possible. However, in order to implement some of Kali's unique features, some changes must be made. To limit the impact of these changes, we strive to send them upstream, either by integrating the feature directly, or by adding the required hooks so that it is straightforward to enable the desired features without further modifying the upstream packages themselves.

The Kali Package Tracker[10] helps us to keep track of our divergence with Debian. At any time, we can look up which package has been forked and whether it is in sync with Debian, or if an update

[9] `https://www.debian.org/releases/testing/`
[10] `http://pkg.kali.org/derivative/kali-dev/`

is required. All our packages are maintained in Git repositories[11] hosting a Debian branch and a Kali branch side-by-side. Thanks to this, updating a forked package is a simple two-step process: update the Debian branch and then merge it into the Kali branch.

While the number of forked packages in Kali is relatively low, the number of additional packages is rather high: in April 2017 there were almost 400. Most of these packages are free software complying with the Debian Free Software Guidelines[12] and our ultimate goal would be to maintain those packages within Debian whenever possible. That is why we strive to comply with the Debian Policy[13] and to follow the good packaging practices used in Debian. Unfortunately, there are also quite a few exceptions where proper packaging was nearly impossible to create. As a result of time being scarce, few packages have been pushed to Debian.

1.3. Purpose and Use Cases

While Kali's focus can be quickly summarized as "penetration testing and security auditing", there are many different tasks involved behind those activities. Kali Linux is built as a *framework*, because it includes many tools covering very different use cases (though they may certainly be used in combination during a penetration test).

For example, Kali Linux can be used on various types of computers: obviously on the laptops of penetration testers, but also on servers of system administrators wishing to monitor their network, on the workstations of forensic analysts, and more unexpectedly, on stealthy embedded devices, typically with ARM CPUs, that can be dropped in the range of a wireless network or plugged in the computer of target users. Many ARM devices are also perfect attack machines due to their small form factors and low power requirements. Kali Linux can also be deployed in the cloud to quickly build a farm of password-cracking machines and on mobile phones and tablets to allow for truly portable penetration testing.

But that is not all; penetration testers also need servers: to use collaboration software within a team of pen-testers, to set up a web server for use in phishing campaigns, to run vulnerability scanning tools, and other related activities.

Once you have booted Kali, you will quickly discover that Kali Linux's main menu is organized by theme across the various kind of tasks and activities that are relevant for pen-testers and other information security professionals as shown in Figure 1.1, "Kali Linux's Applications Menu" [page 6].

[11] `http://git.kali.org`
[12] `https://www.debian.org/social_contract`
[13] `https://www.debian.org/doc/debian-policy/`

Figure 1.1 *Kali Linux's Applications Menu*

These tasks and activities include:

- Information Gathering: Collecting data about the target network and its structure, identifying computers, their operating systems, and the services that they run. Identifying potentially sensitive parts of the information system. Extracting all sorts of listings from running directory services.
- Vulnerability Analysis: Quickly testing whether a local or remote system is affected by a number of known vulnerabilities or insecure configurations. Vulnerability scanners use databases containing thousands of signatures to identify potential vulnerabilities.
- Web Application Analysis: Identifying misconfigurations and security weaknesses in web applications. It is crucial to identify and mitigate these issues given that the public availability of these applications makes them ideal targets for attackers.
- Database Assessment: From SQL injection to attacking credentials, database attacks are a very common vector for attackers. Tools that test for attack vectors ranging from SQL injection to data extraction and analysis can be found here.
- Password Attacks: Authentication systems are always a go-to attack vector. Many useful tools can be found here, from online password attack tools to offline attacks against the encryption or hashing systems.
- Wireless Attacks: The pervasive nature of wireless networks means that they will always be a commonly attacked vector. With its wide range of support for multiple wireless cards, Kali is an obvious choice for attacks against multiple types of wireless networks.
- Reverse Engineering: Reverse engineering is an activity with many purposes. In support of offensive activities, it is one of the primary methods for vulnerability identification and

exploit development. On the defensive side, it is used to analyze malware employed in targeted attacks. In this capacity, the goal is to identify the capabilities of a given piece of tradecraft.

- Exploitation Tools: Exploiting, or taking advantage of a (formerly identified) vulnerability, allows you to gain control of a remote machine (or device). This access can then be used for further privilege escalation attacks, either locally on the compromised machine, or on other machines accessible on its local network. This category contains a number of tools and utilities that simplify the process of writing your own exploits.
- Sniffing & Spoofing: Gaining access to the data as they travel across the network is often advantageous for an attacker. Here you can find spoofing tools that allow you to impersonate a legitimate user as well as sniffing tools that allow you to capture and analyze data right off the wire. When used together, these tools can be very powerful.
- Post Exploitation: Once you have gained access to a system, you will often want to maintain that level of access or extend control by laterally moving across the network. Tools that assist in these goals are found here.
- Forensics: Forensic Linux live boot environments have been very popular for years now. Kali contains a large number of popular Linux-based forensic tools allowing you to do everything from initial triage, to data imaging, to full analysis and case management.
- Reporting Tools: A penetration test is only complete once the findings have been reported. This category contains tools to help collate the data collected from information-gathering tools, discover non-obvious relationships, and bring everything together in various reports.
- Social Engineering Tools: When the technical side is well-secured, there is often the possibility of exploiting human behavior as an attack vector. Given the right influence, people can frequently be induced to take actions that compromise the security of the environment. Did the USB key that the secretary just plugged in contain a harmless PDF? Or was it also a Trojan horse that installed a backdoor? Was the banking website the accountant just logged into the expected website or a perfect copy used for phishing purposes? This category contains tools that aid in these types of attacks.
- System Services: This category contains tools that allow you to start and stop applications that run in the background as system services.

1.4. Main Kali Linux Features

Kali Linux is a Linux distribution that contains its own collection of hundreds of software tools specifically tailored for their target users—penetration testers and other security professionals. It also comes with an installation program to completely setup Kali Linux as the main operating system on any computer.

This is pretty much like all other existing Linux distributions but there are other features that differentiate Kali Linux, many of which are tailored to the specific needs of penetration testers. Let's have a look at some of those features.

1.4.1. A Live System

Contrary to most Linux distributions, the main ISO image that you download is not simply dedicated to installing the operating system; it can also be used as a bootable live system. In other words, you can use Kali Linux without installing it, just by booting the ISO image (usually after having copied the image onto a USB key).

The live system contains the tools most commonly used by penetration testers so even if your day-to-day system is not Kali Linux, you can simply insert the disk or USB key and reboot to run Kali. However, keep in mind that the default configuration will not preserve changes between reboots. If you configure persistence with a USB key (see section 9.4, "Adding Persistence to the Live ISO with a USB Key" [page 239]), then you can tweak the system to your liking (modify config files, save reports, upgrade software, and install additional packages, for example), and the changes will be retained across reboots.

1.4.2. Forensics Mode

In general, when doing forensic work on a system, you want to avoid any activity that would alter the data on the analyzed system in any way. Unfortunately, modern desktop environments tend to interfere with this objective by trying to auto-mount any disk(s) they detect. To avoid this behavior, Kali Linux has a forensics mode that can be enabled from the boot menu: it will disable all such features.

The live system is particularly useful for forensics purposes, because it is possible to reboot any computer into a Kali Linux system without accessing or modifying its hard disks.

1.4.3. A Custom Linux Kernel

Kali Linux always provides a customized recent Linux kernel, based on the version in Debian Unstable. This ensures solid hardware support, especially for a wide range of wireless devices. The kernel is patched for wireless injection support since many wireless security assessment tools rely on this feature.

Since many hardware devices require up-to-date firmware files (found in `/lib/firmware/`), Kali installs them all by default—including the firmware available in Debian's non-free section. Those are not installed by default in Debian, because they are closed-source and thus not part of Debian proper.

1.4.4. Completely Customizable

Kali Linux is built by penetration testers for penetration testers but we understand that not everyone will agree with our design decisions or choice of tools to include by default. With this in mind, we always ensure that Kali Linux is easy to customize based on your own needs and preferences. To this end, we publish the live-build configuration used to build the official Kali images so you can customize it to your liking. It is very easy to start from this published configuration and implement various changes based on your needs thanks to the versatility of live-build.

Live-build includes many features to modify the installed system, install supplementary files, install additional packages, run arbitrary commands, and change the values pre-seeded to debconf.

1.4.5. A Trustable Operating System

Users of a security distribution rightfully want to know that it can be trusted and that it has been developed in plain sight, allowing anyone to inspect the source code. Kali Linux is developed by a small team of knowledgeable developers working transparently and following the best security practices: they upload signed source packages, which are then built on dedicated build daemons. The packages are then checksummed and distributed as part of a signed repository.

The work done on the packages can be fully reviewed through the packaging Git repositories[14] (which contain signed tags) that are used to build the Kali source packages. The evolution of each package can also be followed through the Kali package tracker[15].

1.4.6. Usable on a Wide Range of ARM Devices

Kali Linux provides binary packages for the armel, armhf, and arm64 ARM architectures. Thanks to the easily installable images provided by Offensive Security, Kali Linux can be deployed on many interesting devices, from smartphones and tablets to Wi-Fi routers and computers of various shapes and sizes.

1.5. Kali Linux Policies

While Kali Linux strives to follow the Debian policy whenever possible, there are some areas where we made significantly different design choices due to the particular needs of security professionals.

[14] `http://git.kali.org`
[15] `http://pkg.kali.org`

1.5.1. Single Root User by Default

Most Linux distributions encourage, quite sensibly, the use of a non-privileged account while running the system and the use of a utility like `sudo` when administrative privileges are needed. This is sound security advice, providing an extra layer of protection between the user and any potentially disruptive or destructive operating system commands or operations. This is especially true for multiple user systems, where user privilege separation is a requirement—misbehavior by one user can disrupt or destroy the work of many users.

Since many tools included in Kali Linux can only be executed with root privileges, this is the default Kali user account. Unlike other Linux distributions, you will not be prompted to create a non-privileged user when installing Kali. This particular policy is a major deviation from most Linux systems and tends to be very confusing for less experienced users. Beginners should be especially careful when using Kali since most destructive mistakes occur when operating with root privileges.

1.5.2. Network Services Disabled by Default

In contrast to Debian, Kali Linux disables any installed service that would listen on a public network interface by default, such as HTTP and SSH.

The rationale behind this decision is to minimize exposure during a penetration test when it is detrimental to announce your presence and risk detection because of unexpected network interactions.

You can still manually enable any services of your choosing by running `systemctl enable` *`service`*. We will get back to this in chapter 5, “Configuring Kali Linux” [page 104] later in this book.

1.5.3. A Curated Collection of Applications

Debian aims to be the universal operating system and puts very few limits on what gets packaged, provided that each package has a maintainer.

By way of contrast, Kali Linux does not package every penetration testing tool available. Instead, we aim to provide only the best freely-licensed tools covering most tasks that a penetration tester might want to perform.

Kali developers working as penetration testers drive the selection process and we leverage their experience and expertise to make enlightened choices. In some cases this is a matter of fact, but there are other, more difficult choices that simply come down to personal preference.

Here are some of the points considered when a new application gets evaluated:

- The usefulness of the application in a penetration testing context

- The unique functionality of the application's features
- The application's license
- The application's resource requirements

Maintaining an updated and useful penetration testing tool repository is a challenging task. We welcome tool suggestions within a dedicated category (*New Tool Requests*) in the Kali Bug Tracker[16]. New tool requests are best received when the submission is well-presented, including an explanation of why the tool is useful, how it compares to other similar applications, and so on.

1.6. Summary

In this chapter we have introduced you to Kali Linux, provided a bit of history, run through some of the primary features, and presented several use cases. We have also discussed some of the policies we have adopted when developing Kali Linux.

Summary Tips:

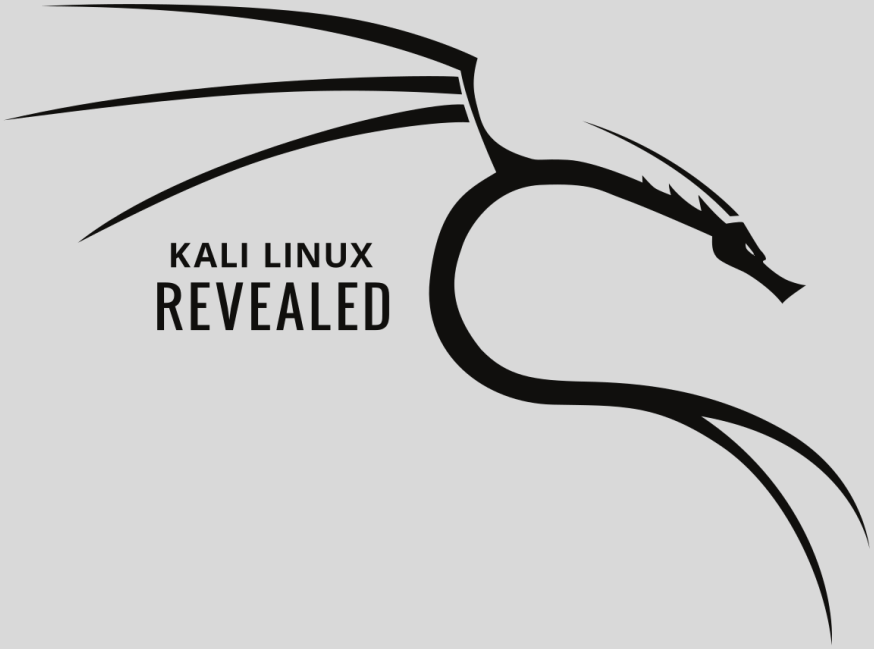
- Kali Linux[17] is an enterprise-ready security auditing Linux distribution based on Debian GNU/Linux. Kali is aimed at security professionals and IT administrators, enabling them to conduct advanced penetration testing, forensic analysis, and security auditing.
- Unlike most mainstream operating systems, Kali Linux is a rolling distribution, which means that *you will receive updates every single day*.
- The Kali Linux distribution is based on Debian Testing[18]. Therefore, most of the packages available in Kali Linux come straight from this Debian repository.
- While Kali's focus can be quickly summarized with "penetration testing and security auditing", there are several use cases including system administrators wishing to monitor their networks, forensic analysis, embedded device installations, wireless monitoring, installation on mobile platforms, and more.
- Kali's menus make it easy to get to tools for various tasks and activities including: vulnerability analysis, web application analysis, database assessment, password attacks, wireless attacks, reverse engineering, exploitation tools, sniffing and spoofing, post exploitation tools, forensics, reporting tools, social engineering tools, and system services.
- Kali Linux has many advanced features including: use as a live (non-installed) system, a robust and safe forensics mode, a custom Linux kernel, ability to completely customize the system, a trusted and secure base operating system, ARM installation capability, secure default network policies, and a curated set of applications.

In the next chapter, we will jump in and try out Kali Linux thanks to its live mode.

[16] `http://bugs.kali.org`
[17] `https://www.kali.org`
[18] `https://www.debian.org/releases/testing/`

Keywords

Download
ISO image
Live boot



Chapter

2

Getting Started with Kali Linux

Contents

Unlike some other operating systems, Kali Linux makes getting started easy, thanks to the fact that its disk images are *live ISOs*, meaning that you can boot the downloaded image without following any prior installation procedure. This means you can use the same image for testing, for use as a bootable USB or DVD-ROM image in a forensics case, or for installing as a permanent operating system on physical or virtual hardware.

Because of this simplicity, it is easy to forget that certain precautions must be taken. Kali users are often the target of those with ill intentions, whether state sponsored groups, elements of organized crime, or individual hackers. The open-source nature of Kali Linux makes it relatively easy to build and distribute fake versions, so it is essential that you get into the habit of downloading from original sources and verifying the integrity and the authenticity of your download. This is especially relevant to security professionals who often have access to sensitive networks and are entrusted with client data.

2.1. Downloading a Kali ISO Image

2.1.1. Where to Download

The only official source of Kali Linux ISO images is the Downloads section of the Kali website. Due to its popularity, numerous sites offer Kali images for download, but they should not be considered trustworthy and indeed may be infected with malware or otherwise cause irreparable damage to your system.

➨ `https://www.kali.org/downloads/`

The website is available over *HTTPS*, making it difficult to impersonate. Being able to carry out a man-in-the-middle attack is not sufficient as the attacker would also need a www.kali.org certificate signed by a Transport Layer Security (TLS) certificate authority that is trusted by the victim's browser. Because certificate authorities exist precisely to prevent this type of problem, they deliver certificates only to people whose identities have been verified and who have provided evidence that they control the corresponding website.

cdimage.kali.org The links found on the download page point to the `cdimage.kali.org` domain, which redirects to a mirror close to you, improving your transfer speed while reducing the burden on Kali's central servers.

A list of available mirrors can be found here:

➨ `http://cdimage.kali.org/README.mirrorlist`

2.1.2. What to Download

The official download page shows a short list of ISO images, as shown in Figure 2.1, "List of Images Offered for Download" [page 15].

Download Kali Linux Images

We generate fresh Kali Linux image files every few months, which we make available for download. This page provides the links to **download Kali Linux** in its latest official release. For a release history, check our Kali Linux Releases page. Please note: You can find unofficial, untested weekly releases at http://cdimage.kali.org/kali-weekly/.

Image Name	Download	Size	Version	sha256sum
Kali 64 bit	ISO | Torrent	2.6G	**2017.1**	49b1c5769b909220060dc4c0e11ae09d97a270a80d259e05773101df62e11e9d
Kali 32 bit	ISO | Torrent	2.7G	**2017.1**	501b3747e5ac7c698217392fe49ec21dacee277404500fc49d4a0ee82625aabe
Kali 64 bit Light	ISO | Torrent	0.8G	**2017.1**	5c0f6300bf9842b724df92cb20e4637f4561ffc03029cdcb21af3902442ae9b0
Kali 32 bit Light	ISO | Torrent	0.8G	**2017.1**	6c83101ecf8702c7d93d32562e822b639d5c577314b448e3b8330995e0f07e0f
Kali 64 bit e17	ISO | Torrent	2.4G	**2017.1**	ae293cf679f38a4f17d090a272ccb13d7619e66d4502374154186c12891fb99c
Kali 64 bit KDE	ISO | Torrent	2.7G	**2017.1**	839741fec378114ff068df3ec2dbed9d8e4fae613e690d50b25ce9cc1468104b
Kali 64 bit Mate	ISO | Torrent	2.6G	**2017.1**	3ea748aa8c5f50d80f020acdbca5f0398ee90242bb4413c12985e1865186ca9e
Kali 64 bit Xfce	ISO | Torrent	2.5G	**2017.1**	8a17c2f54850585760b9d32a22e26df9a28f395b401753fa0a9b298aef4c4593
Kali 64 bit LXDE	ISO | Torrent	2.5G	**2017.1**	35eae65aaaabba8188dfd963e45b7b4d76e0684e7721c7d232cf18320b7cae3b
Kali armhf	Image | Torrent	0.5G	**2017.1**	a75199aa8a3d7b64561bc03fcd6e3ff8b94743c8769eecfaa4b719f04f7cbb63
Kali armel	Image | Torrent	0.4G	**2017.1**	180414422196f0797c1ea5f3c18682bc4b3ced871cb3e874e90de52dd4af877c

Figure 2.1 *List of Images Offered for Download*

All disk images labeled 32- or 64-bit refer to images suitable for CPUs, found in most modern desktop and laptop computers. If you are downloading for use on a fairly modern machine, it most likely contains a 64-bit processor. If you are unsure, rest assured that all 64-bit processors can run 32-bit instructions. You can always download and run the 32-bit image. The reverse is not true, however. Refer to the sidebar for more detailed information.

If you are planning to install Kali on an embedded device, smartphone, Chromebook, access point, or any other device with an ARM processor, you must use the Linux *armel* or *armhf* images.

Is My CPU 32- or 64-bit?

Under Windows, you can find this information by running the *System Information* application (found in the Accessories > System Tools folder). On the System Summary screen, you can inspect the System Type field: it will contain "x64-based PC" for a 64-bit CPU or "x86-based PC" for a 32-bit CPU.

Under OS X/macOS, there is no standard application showing this information but you can still infer it from the output of the `uname -m` command run on the terminal. It will return `x86_64` on a system with a 64-bit kernel (which can only run on a 64-bit CPU) and on systems with a 32-bit kernel, it will return `i386` or something similar (`i486`, `i586`, or `i686`). Any 32-bit kernel can run on a 64-bit CPU, but since Apple controls the hardware and the software, it is unlikely you will find this configuration.

Under Linux, you can inspect the flags field in the `/proc/cpuinfo` virtual file. If it contains the `lm` attribute, then your CPU is a 64-bit; otherwise, it is a 32-bit. The following command line will tell you what kind of CPU you have:

```
$ grep -qP '^flags\s*:.*\blm\b' /proc/cpuinfo && echo 64-bit
    ➥ || echo 32-bit
64-bit
```

Now that you know whether you need a 32-bit or 64-bit image, there is only one step left: selecting the kind of image. The default Kali Linux image and the Kali Linux Light variant are both live ISOs that can be used to run the live system or to start the installation process. They differ only by the set of pre-installed applications. The default image comes with the GNOME desktop and a large collection of packages found to be appropriate for most penetration testers, while the light image comes with the Xfce desktop, (which is much less demanding on system resources), and a limited collection of packages, allowing you to choose only the apps you need. The remaining images use alternate desktop environments but come with the same large package collection as the main image.

Once you have decided on the image you need, you can download the image by clicking on "ISO" in the respective row. Alternatively, you can download the image from the BitTorrent peer-to-peer network by clicking on "Torrent," provided that you have a BitTorrent client associated with the .torrent extension.

While your chosen ISO image is downloading, you should take note of the checksum written in the sha256sum column. Once you have downloaded your image, use this checksum to verify that the downloaded image matches the one the Kali development team put online (see next section).

2.1.3. Verifying Integrity and Authenticity

Security professionals must verify the integrity of their tools to not only protect their data and networks but also those of their clients. While the Kali download page is TLS-protected, the actual download link points to an unencrypted URL that offers no protection against potential man-in-the-middle attacks. The fact that Kali relies on a network of external mirrors to distribute the

image means that you should not blindly trust what you download. The mirror you were directed to may have been compromised, or you might be the victim of an attack yourself.

To alleviate this, the Kali project always provides checksums of the images it distributes. But to make such a check effective, you must be sure that the checksum you grabbed is effectively the checksum published by the Kali Linux developers. You have different ways to ascertain this.

Relying on the TLS-Protected Website

When you retrieve the checksum from the TLS-protected download webpage, its origin is indirectly guaranteed by the X.509 certificate security model: the content you see comes from a web site that is effectively under the control of the person who requested the TLS certificate.

Now you should generate the checksum of your downloaded image and ensure that it matches what you recorded from the Kali website:

```
$ sha256sum kali-linux-2017.1-amd64.iso
49b1c5769b909220060dc4c0e11ae09d97a270a80d259e05773101df62e11e9d  kali-linux-2016.2-amd64.iso
```

If your generated checksum matches the one on the Kali Linux download page, you have the correct file. If the checksums differ, there is a problem, although this does not indicate a compromise or an attack; downloads occasionally get corrupted as they traverse the Internet. Try your download again, from another official Kali mirror, if possible (see "cdimage.kali.org" [page 14] for more information about available mirrors).

Relying on PGP's Web of Trust

If you don't trust HTTPS for authentication, you are a bit paranoid but rightfully so. There are many examples of badly managed certificate authorities that issued rogue certificates, which ended up being misused. You may also be the victim of a "friendly" man-in-the-middle attack implemented on many corporate networks, using a custom, browser-implanted trust store that presents fake certificates to encrypted websites, allowing corporate auditors to monitor encrypted traffic.

For cases like this, we also provide a GnuPG key that we use to sign the checksums of the images we provide. The key's identifiers and its fingerprints are shown here:

```
pub   rsa4096/0xED444FF07D8D0BF6 2012-03-05 [SC] [expires: 2018-02-02]
      Key fingerprint = 44C6 513A 8E4F B3D3 0875  F758 ED44 4FF0 7D8D 0BF6
uid                   [  full  ] Kali Linux Repository <devel@kali.org>
sub   rsa4096/0xA8373E18FC0D0DCB 2012-03-05 [E] [expires: 2018-02-02]
```

This key is part of a global *web of trust* because it has been signed at least by me (Raphaël Hertzog) and I am part of the web of trust due to my heavy GnuPG usage as a Debian developer.

The PGP/GPG security model is very unique. Anyone can generate any key with any identity, but you would only trust that key if it has been signed by another key that you already trust. When you sign a key, you certify that you met the holder of the key and that you know that the associated identity is correct. And you define the initial set of keys that you trust, which obviously includes your own key.

This model has its own limitations so you can opt to download Kali's public key over HTTPS (or from a keyserver) and just decide that you trust it because its fingerprint matches what we announced in multiple places, including just above in this book:

```
$ wget -q -O - https://www.kali.org/archive-key.asc | gpg --import
[ or ]
$ gpg --keyserver hkp://keys.gnupg.net --recv-key ED444FF07D8D0BF6
gpg: key 0xED444FF07D8D0BF6: public key "Kali Linux Repository <devel@kali.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
[...]
$ gpg --fingerprint 7D8D0BF6
[...]
      Key fingerprint = 44C6 513A 8E4F B3D3 0875  F758 ED44 4FF0 7D8D 0BF6
[...]
```

After you have retrieved the key, you can use it to verify the checksums of the distributed images. Let's download the file with the checksums (`SHA256SUMS`) and the associated signature file (`SHA256SUMS.gpg`) and verify the signature:

```
$ wget http://cdimage.kali.org/current/SHA256SUMS
[...]
$ wget http://cdimage.kali.org/current/SHA256SUMS.gpg
[...]
$ gpg --verify SHA256SUMS.gpg SHA256SUMS
gpg: Signature made Thu 16 Mar 2017 08:55:45 AM MDT
gpg:                using RSA key ED444FF07D8D0BF6
gpg: Good signature from "Kali Linux Repository <devel@kali.org>"
```

If you get that "Good signature" message, you can trust the content of the `SHA256SUMS` file and use it to verify the files you downloaded. Otherwise, there is a problem. You should review whether you downloaded the files from a legitimate Kali Linux mirror.

Note that you can use the following command line to verify that the downloaded file has the same checksum that is listed in `SHA256SUMS`, provided that the downloaded ISO file is in the same directory:

```
$ grep kali-linux-2017.1-amd64.iso SHA256SUMS | sha256sum -c
kali-linux-2017.1-amd64.iso: OK
```

If you don't get OK in response, then the file you have downloaded is different from the one released by the Kali team. It cannot be trusted and should not be used.

2.1.4. Copying the Image on a DVD-ROM or USB Key

Unless you want to run Kali Linux in a virtual machine, the ISO image is of limited use in and of itself. You must burn it on a DVD-ROM or copy it onto a USB key to be able to boot your machine into Kali Linux.

We won't cover how to burn the ISO image onto a DVD-ROM, as the process varies widely by platform and environment, but in most cases, right clicking on the `.iso` file will present a contextual menu item that executes a DVD-ROM burning application. Try it out!

> Warning
>
> 
>
> In this section, you will learn how to overwrite an arbitrary disk with a Kali Linux ISO image. Always double-check the target disk before launching the operation as a single mistake would likely cause complete data loss and possibly damage your setup beyond repair.

Creating a Bootable Kali USB Drive on Windows

As a prerequisite, you should download and install *Win32 Disk Imager*:

➨ `https://sourceforge.net/projects/win32diskimager/`

Plug your USB key into your Windows PC and note the drive designator associated to it (for example, "E:\").

Launch *Win32 Disk Imager* and choose the Kali Linux ISO file that you want to copy on the USB key. Verify that the letter of the device selected corresponds with that assigned to the USB key. Once you are certain that you have selected the correct drive, click the Write button and confirm that you want to overwrite the contents of the USB key as shown in Figure 2.2, "Win32 Disk Imager in action" [page 20].

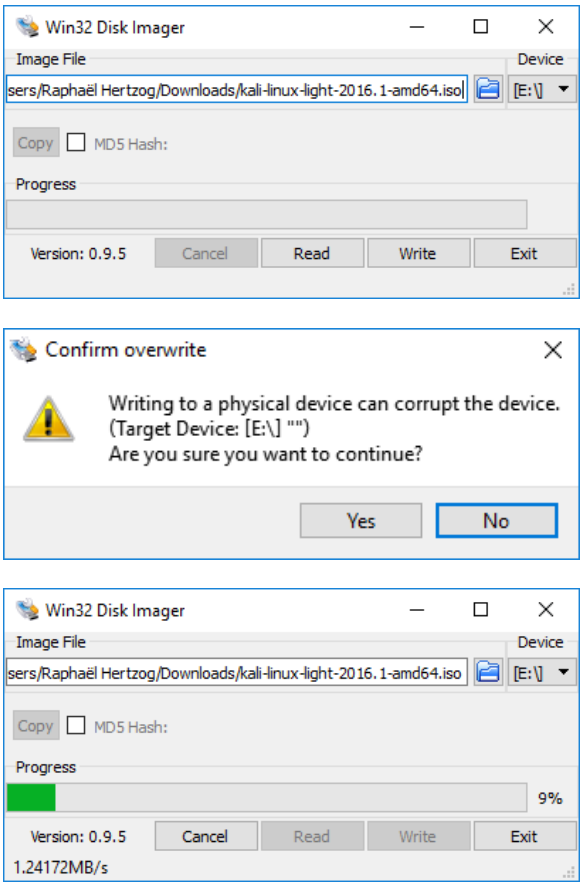


Figure 2.2 *Win32 Disk Imager in action*

Once the copy is completed, safely eject the USB drive from the Windows system. You can now use the USB device to boot Kali Linux.

Creating a Bootable Kali USB Drive on Linux

Creating a bootable Kali Linux USB key in a Linux environment is easy. The GNOME desktop environment, which is installed by default in many Linux distributions, comes with a *Disks* utility (in the *gnome-disk-utility* package, which is already installed in the stock Kali image). That program shows a list of disks, which refreshes dynamically when you plug or unplug a disk. When you select your USB key in the list of disks, detailed information will appear and will help you confirm that you selected the correct disk. Note that you can find its device name in the title bar as shown in Figure 2.3, “GNOME Disks” [page 21].

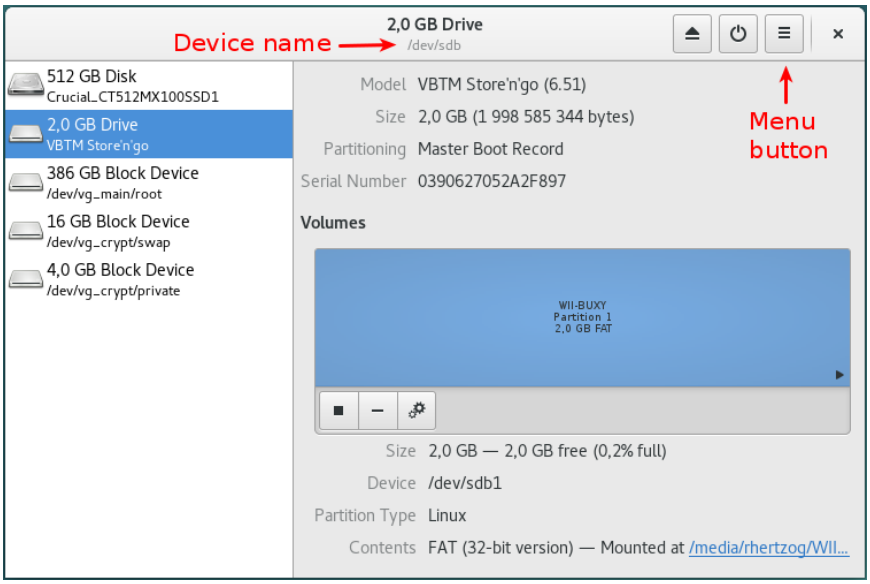


Figure 2.3 *GNOME Disks*

Click on the menu button and select Restore Disk Image... in the displayed pop-up menu. Select the ISO image that you formerly downloaded and click on Start Restoring... as shown in Figure 2.4, “Restore Disk Image Dialog” [page 21].

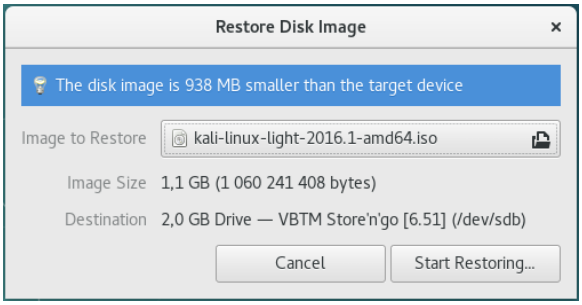


Figure 2.4 *Restore Disk Image Dialog*

Enjoy a cup of coffee while it finishes copying the image on the USB key (Figure 2.5, "Progression of the Image Restoration" [page 22]).

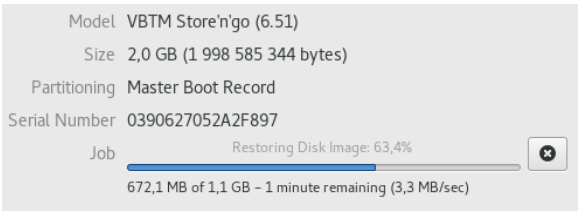


Figure 2.5 *Progression of the Image Restoration*

Create the Bootable USB Drive from the Command Line

Even though the graphical process is fairly straightforward, the operation is just as easy for command line users.

When you insert your USB key, the Linux kernel will detect it and assign it a name, which is printed in the kernel logs. You can find its name by inspecting the logs returned by `dmesg`.

```
$ dmesg
[...]
[234743.896134] usb 1-1.2: new high-speed USB device number 6 using ehci-pci
[234743.990764] usb 1-1.2: New USB device found, idVendor=08ec, idProduct=0020
[234743.990771] usb 1-1.2: New USB device strings: Mfr=1, Product=2,
     ➥ SerialNumber=3
[234743.990774] usb 1-1.2: Product: Store'n'go
[234743.990777] usb 1-1.2: Manufacturer: Verbatim
[234743.990780] usb 1-1.2: SerialNumber: 0390627052A2F897
[234743.991845] usb-storage 1-1.2:1.0: USB Mass Storage device detected
[234743.992017] scsi host7: usb-storage 1-1.2:1.0
[234744.993818] scsi 7:0:0:0: Direct-Access     VBTM     Store'n'go       6.51
     ➥ PQ: 0 ANSI: 0 CCS
[234744.994425] sd 7:0:0:0: Attached scsi generic sg1 type 0
[234744.995753] sd 7:0:0:0: [sdb] 3903487 512-byte logical blocks: (2.00 GB
     ➥ /1.86 GiB)
[234744.996663] sd 7:0:0:0: [sdb] Write Protect is off
[234744.996669] sd 7:0:0:0: [sdb] Mode Sense: 45 00 00 08
[234744.997518] sd 7:0:0:0: [sdb] No Caching mode page found
[234744.997524] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[234745.009375]  sdb: sdb1
[234745.015113] sd 7:0:0:0: [sdb] Attached SCSI removable disk
```

Now that you know that the USB key is available as `/dev/sdb`, you can proceed to copy the image with the `dd` command:

```
# dd if=kali-linux-light-2017.1-amd64.iso of=/dev/sdb
2070784+0 records in
2070784+0 records out
1060241408 bytes (1.1 GB, 1011 MiB) copied, 334.175 s, 3.2 MB/s
```

Note that you need root permissions for this operation to succeed and you should also ensure that the USB key is unused. That is, you should make sure that none of its partitions are mounted. The command also assumes that it is run while in the directory hosting the ISO image, otherwise the full path will need to be provided.

For reference, `if` stands for "input file" and `of` for "output file." The `dd` command reads data from the input file and writes it back to the output file. It does not show any progress information so you must be patient while it is doing its work (It is not unusual for the command to take more than half an hour!). Look at the write activity LED on the USB key if you want to double check that the command is working. The statistics shown above are displayed only when the command has completed. On OS X/macOS, you can also press CTRL+T during the operation to get statistical information about the copy including how much data has been copied.

Creating a Bootable Kali USB Drive on OS X/macOS

OS X/macOS is based on UNIX, so the process of creating a bootable Kali Linux USB drive is similar to the Linux procedure. Once you have downloaded and verified your chosen Kali ISO file, use `dd` to copy it over to your USB stick.

To identify the device name of the USB key, run `diskutil list` to list the disks available on your system. Next, insert your USB key and run the `diskutil list` command again. The second output should list an additional disk. You can determine the device name of the USB key by comparing the output from both commands. Look for a new line identifying your USB disk and note the `/dev/diskX` where *X* represents the disk ID.

You should make sure that the USB key is not mounted, which can be accomplished with an explicit unmount command (assuming `/dev/disk6` is the device name of the USB key):

```
$ diskutil unmount /dev/disk6
```

Now proceed to execute the `dd` command. This time, add a supplementary parameter — bs for block size. It defines the size of the block that is read from the input file and then written to the output file.

```
# dd if=kali-linux-light-2017.1-amd64.iso of=/dev/disk6 bs=1M
1011+0 records in
1011+0 records out
1060241408 bytes transferred in 327.061 secs (3242328 bytes/sec)
```

That's it. Your USB key is now ready and you can boot from it or use it to install Kali Linux.

Booting an Alternate Disk on OS X/macOS

To boot from an alternate drive on an OS X/macOS system, bring up the boot menu by pressing and holding the Option key immediately after powering on the device and selecting the drive you want to use.

For more information, see Apple's knowledge base[1].

[1] `http://support.apple.com/kb/ht1310`

2.2. Booting a Kali ISO Image in Live Mode

2.2.1. On a Real Computer

As a prerequisite, you need either a USB key prepared (as detailed in the previous section) or a DVD-ROM burned with a Kali Linux ISO image.

The BIOS/UEFI is responsible for the early boot process and can be configured through a piece of software called Setup. In particular, it allows users to choose which boot device is preferred. In this case, you want to select either the DVD-ROM drive or USB drive, depending on which device you have created.

Starting Setup usually involves pressing a particular key very soon after the computer is powered on. This key is often Del or Esc, and sometimes F2 or F10. Most of the time, the choice is briefly flashed onscreen when the computer powers on, before the operating system loads.

Once the BIOS/UEFI has been properly configured to boot from your device, booting Kali Linux is simply a matter of inserting the DVD-ROM or plugging in the USB drive and powering on the computer.

Disable Secure Boot While the Kali Linux images can be booted in UEFI mode, they do not support *secure boot*. You should disable that feature in Setup.

2.2.2. In a Virtual Machine

Virtual machines have multiple benefits for Kali Linux users. They are especially useful if you want to try out Kali Linux but aren't ready to commit to installing it permanently on your machine or if you have a powerful system and want to run multiple operating systems simultaneously. This is a popular choice for many penetration testers and security professionals who need to use the wide range of tools available in Kali Linux but still want to have full access to their primary operating system. This also provides them with the ability to archive or securely delete the virtual machine and any client data it may contain rather than reinstalling their entire operating system.

The snapshot features of virtualization software also make it easy to experiment with potentially dangerous operations, such as malware analysis, while allowing for an easy way out by restoring a previous snapshot.

There are many virtualization tools available for all major operating systems, including *VirtualBox®*, *VMware Workstation®*, *Xen*, *KVM*, and *Hyper-V* to name a few. Ultimately, you will use the one that best suits you but we will cover the two most frequently-used in a desktop context: *VirtualBox®* and *VMware Workstation Pro®*, both running on Windows 10. If you don't have corporate policy constraints or personal preference, our recommendation is that you try out VirtualBox first, as it is free, works well, is (mostly) open-source, and is available for most operating systems.

For the next sections, we will assume that you have already installed the appropriate virtualization tool and are familiar with its operation.

Preliminary Remarks

To fully benefit from virtualization, you should have a CPU with the appropriate virtualization features and they should not be disabled by the BIOS/UEFI. Double check for any "Intel® Virtualization Technology" and/or "Intel® VT-d Feature" options in the Setup screens.

You should also have a 64-bit host operating system, such as amd64 architecture for Debian-based Linux distributions, x86_64 architecture for RedHat-based Linux distributions, and Windows ... 64-bit for Windows.

If you lack any of the prerequisites, either the virtualization tool will not work properly or it will be restricted to running only 32-bit guest operating systems.

Since virtualization tools hook into the host operating system and hardware at a low level, there are often incompatibilities between them. Do not expect these tools to run well at the same time. Also, beware that professional versions of Windows come with *Hyper-V* installed and enabled, which might interfere with your virtualization tool of choice. To turn it off, execute "Turn windows features on or off" from Windows Settings.

After the initial installation, VirtualBox's main screen looks something like Figure 2.6, "VirtualBox's Start Screen" [page 26].

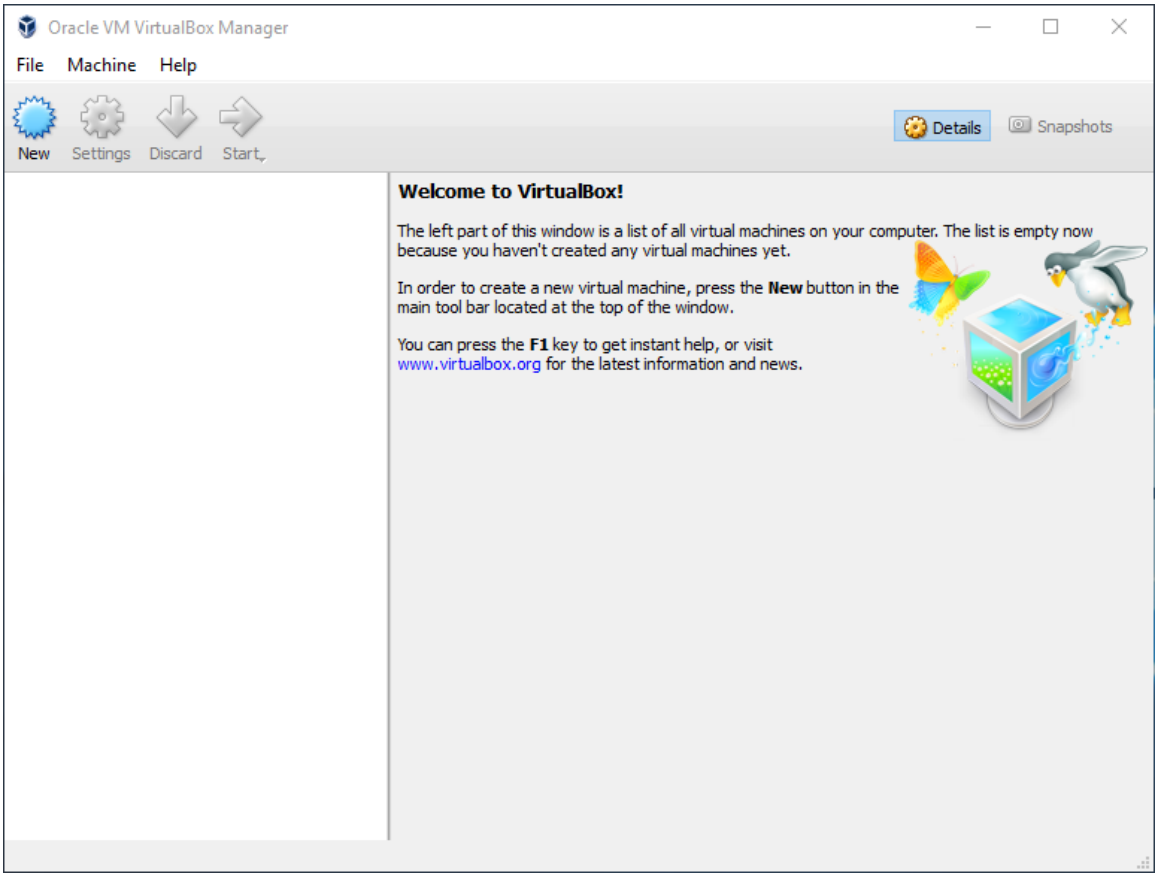


Figure 2.6 *VirtualBox's Start Screen*

Click on New (Figure 2.7, "Name and Operating System" [page 27]) to start a wizard that will guide you through the multiple steps required to input all the parameters of the new virtual machine.

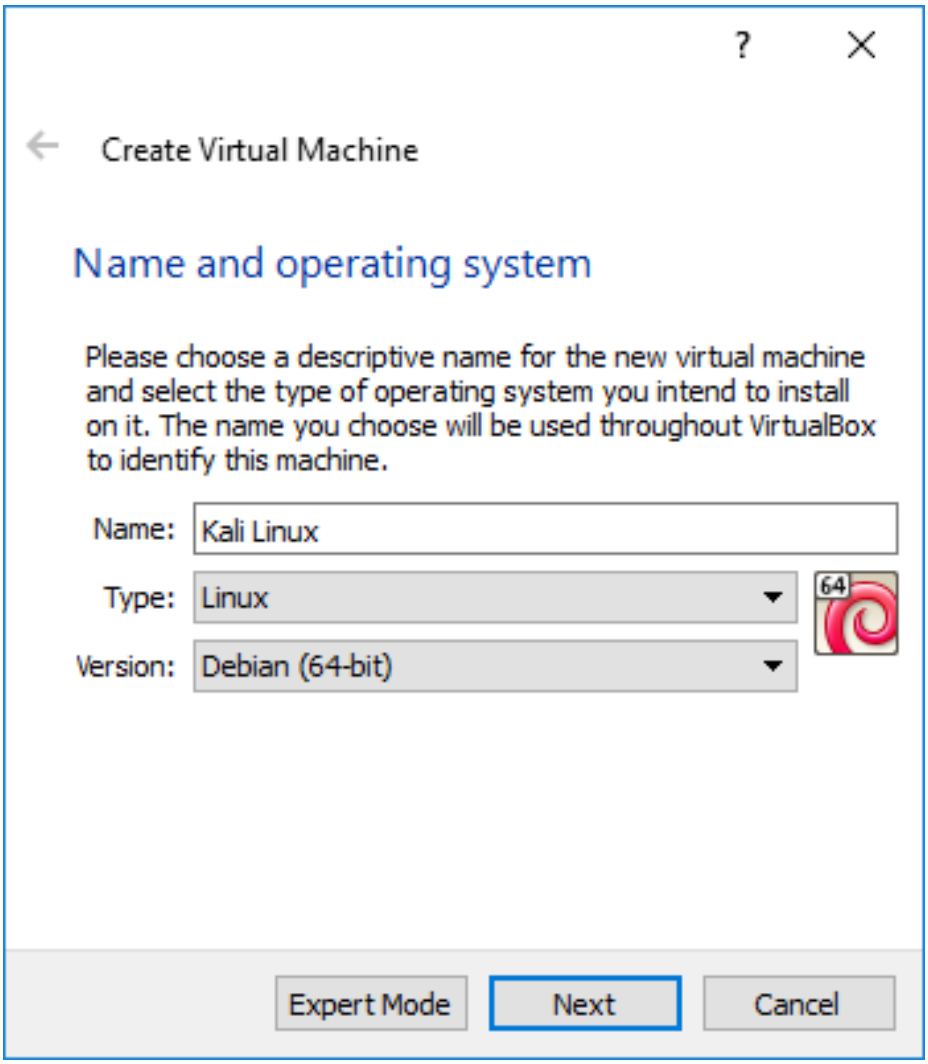


Figure 2.7 *Name and Operating System*

In the first step, shown in Figure 2.7, "Name and Operating System" [page 27], you must assign a name to your new virtual machine. Use "Kali Linux." You must also indicate what kind of operating system will be used. Since Kali Linux is based on Debian GNU/Linux, select Linux for the type and Debian (32-bit) or Debian (64-bit) for the version. Although any other Linux version will most likely work, this will help distinguish between the various virtual machines that you might have installed.

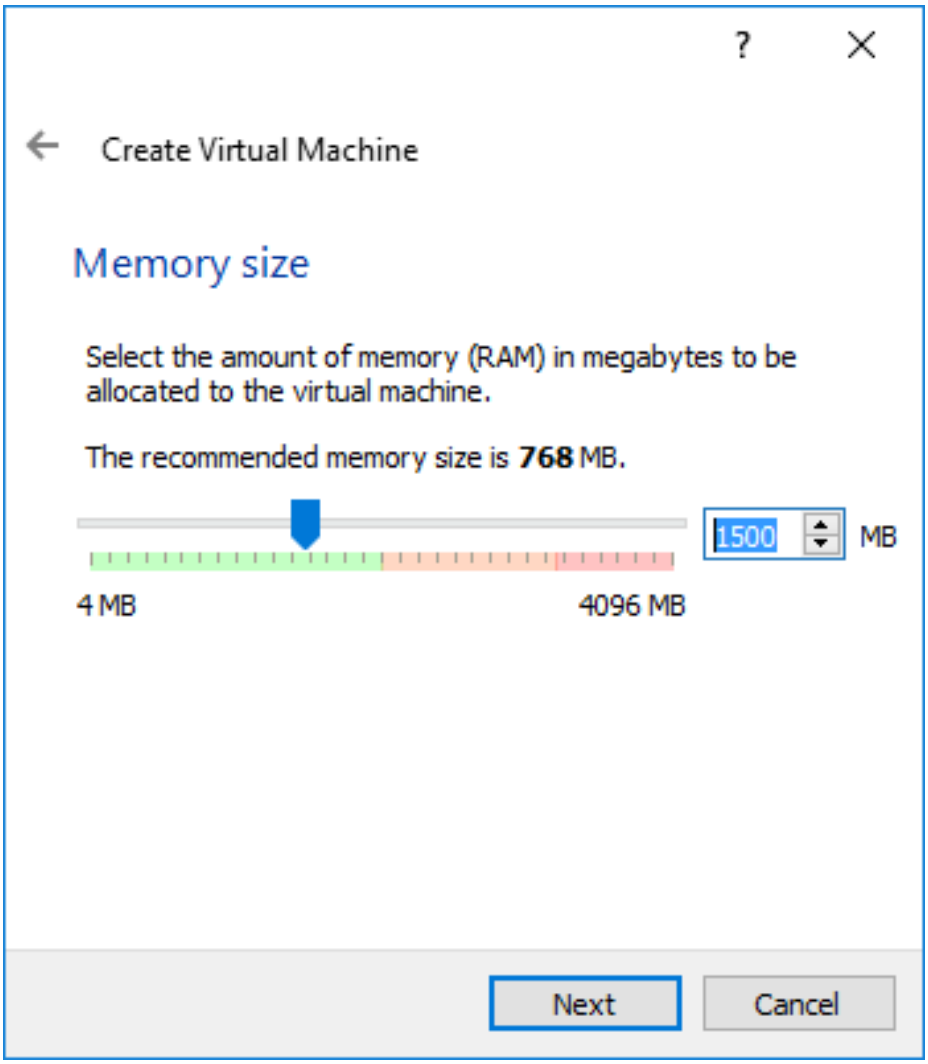


Figure 2.8 *Memory Size*

In the second step, you must decide how much memory to allocate to the virtual machine. While the recommended size of 768 MB is acceptable for a Debian virtual machine acting as a server, it is definitely not enough to run a Kali desktop system, especially not for a Kali Linux live system since the live system uses memory to store changes made to the file system. We recommend increasing the value to 1500 MB (Figure 2.8, "Memory Size" [page 28]) and highly recommend that you allocate no less than 2048 MB of RAM.

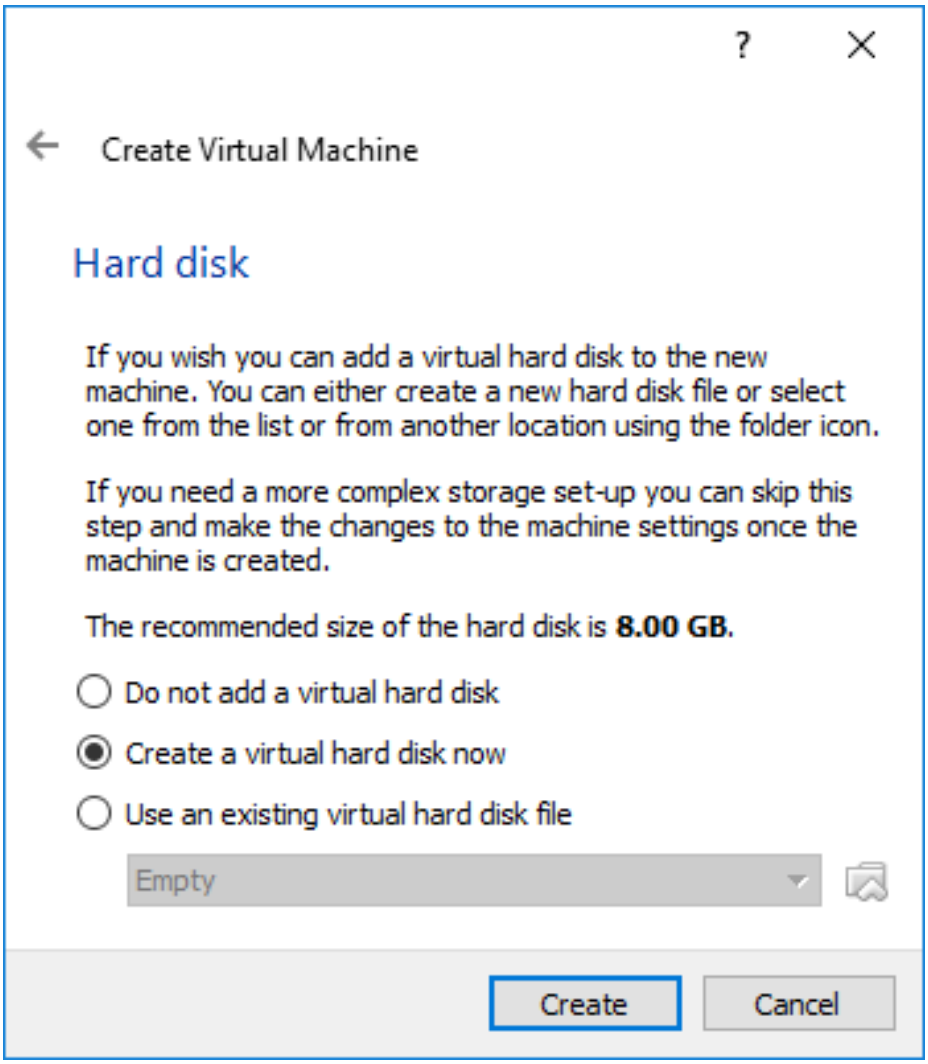


Figure 2.9 *Hard disk*

In the third step (shown in Figure 2.9, "Hard disk" [page 29]), you are prompted to choose a physical or virtual hard disk for your new virtual machine. Although a hard disk is not required to run Kali Linux as a live system, add one for when we demonstrate the installation procedure later, in chapter 4, "Installing Kali Linux" [page 66].

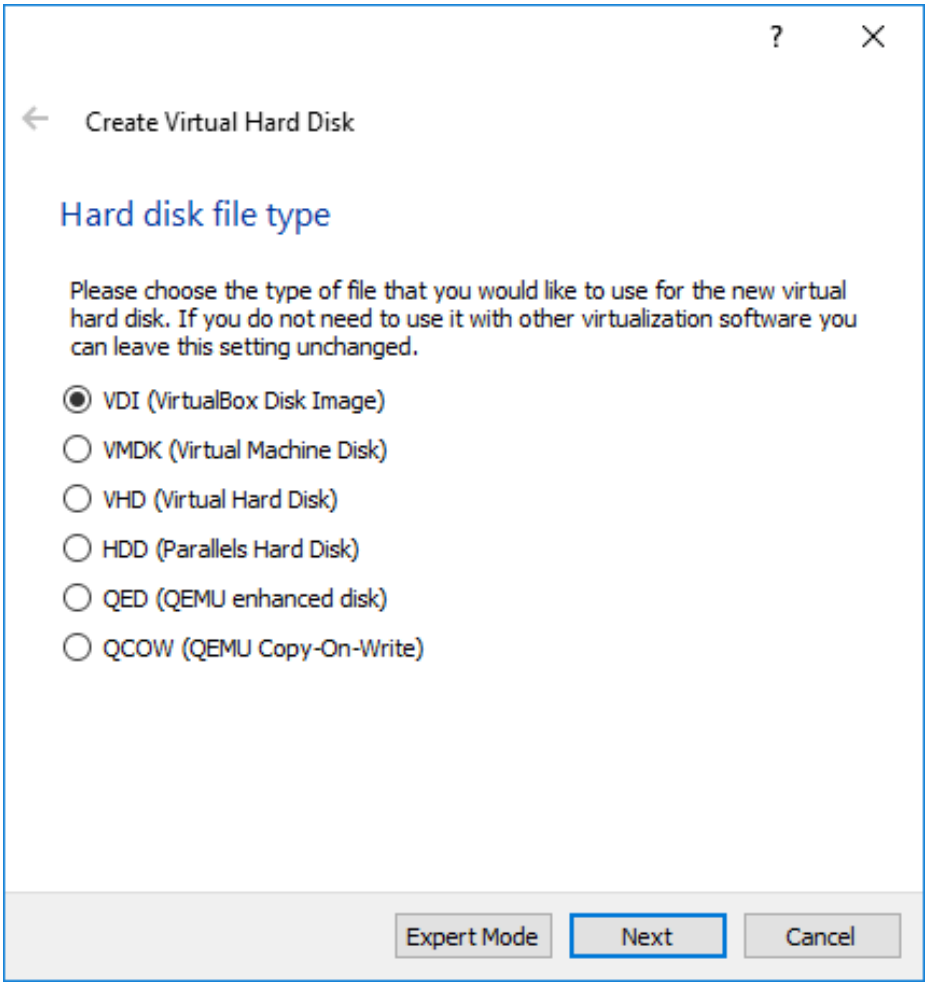


Figure 2.10 *Hard Disk File Type*

The content of the hard disk of the virtual machine is stored on the host machine as a file. VirtualBox is able to store the contents of the hard disk using multiple formats (shown in Figure 2.10, "Hard Disk File Type" [page 30]): the default (VDI) corresponds to VirtualBox's native format; VMDK is the format used by VMware; QCOW is the format used by QEMU. Keep the default value, because you don't have any reason to change it. The ability to use multiple formats is interesting mainly when you want to move a virtual machine from one virtualization tool to another.

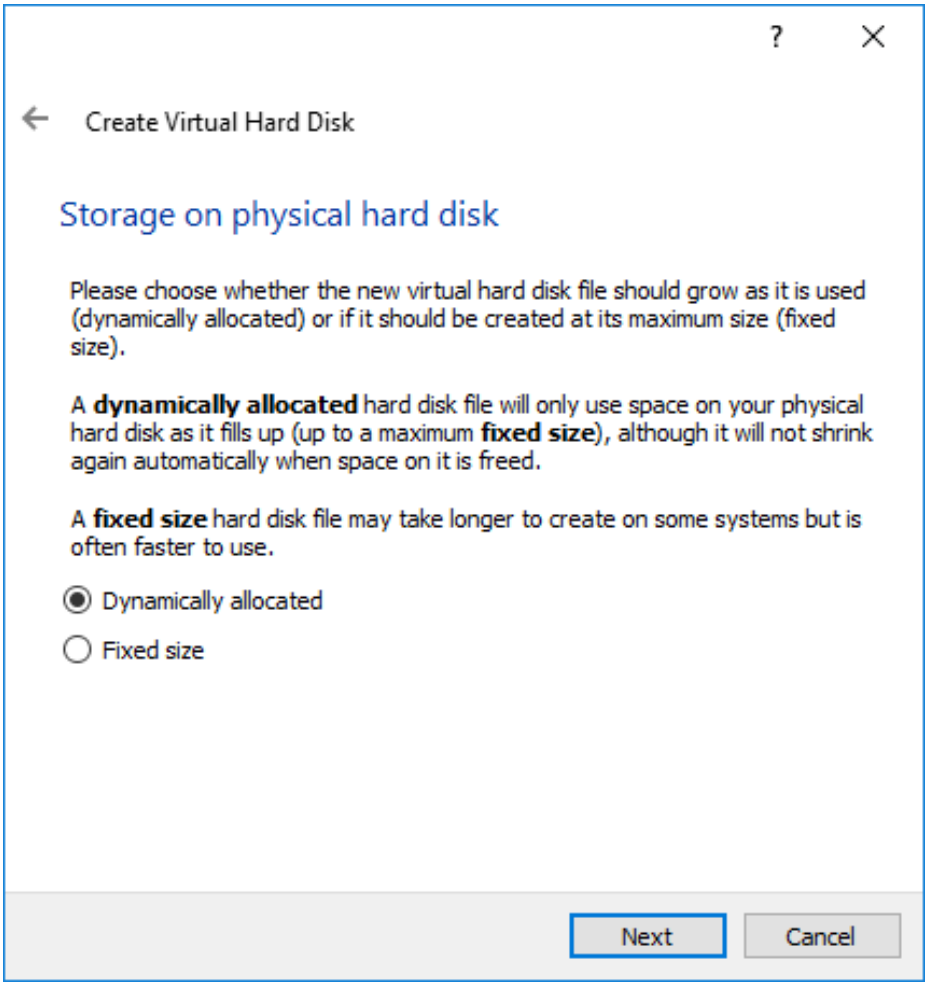


Figure 2.11 *Storage on Physical Hard Disk*

The explanation text in Figure 2.11, "Storage on Physical Hard Disk" [page 31] clearly describes the advantages and drawbacks of dynamic and fixed disk allocation. In this example, we accept the default selection (Dynamically allocated), since we are using a laptop with SSD disks. We don't want to waste space and won't need the extra bit of performance as the machine is already quite fast to begin with.

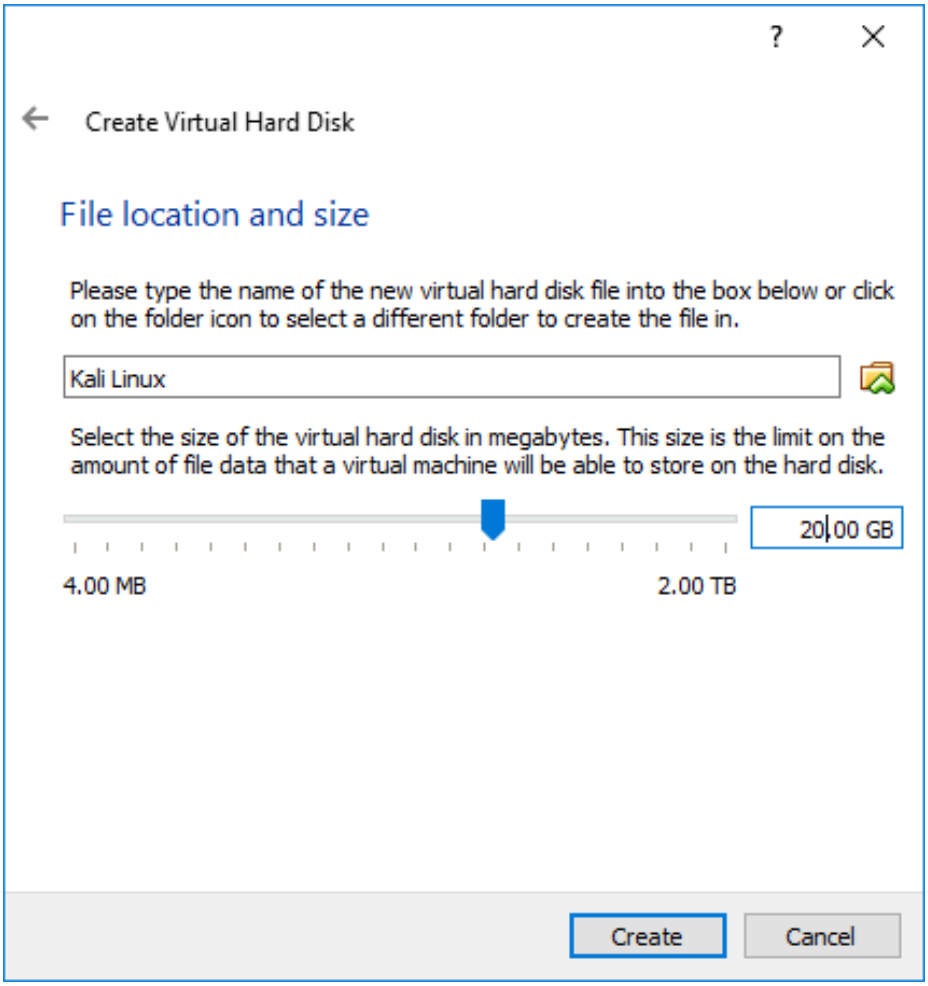


Figure 2.12 *File Location and Size*

The default hard disk size of 8 GB shown in Figure 2.12, "File Location and Size" [page 32] is not enough for a standard installation of Kali Linux, so increase the size to 20 GB. You can also tweak the name and the location of the disk image. This can be handy when you don't have enough space on your hard disk, allowing you to store the disk image on an external drive.

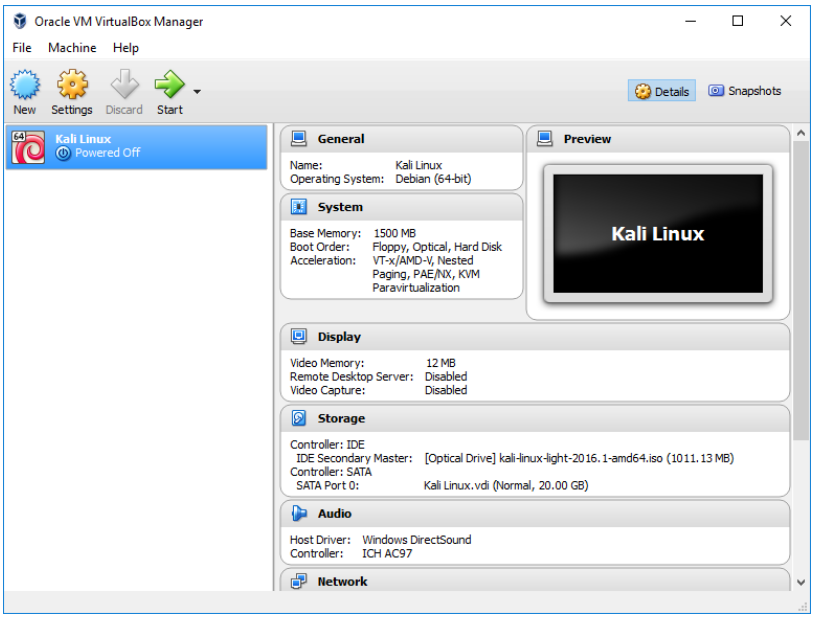


Figure 2.13 *The New Virtual Machine Appears in the List*

The virtual machine has been created but you can't really run it yet, because there is no operating system installed. You also have some settings to tweak. Click on Settings on the VM Manager screen and let's review some of the most useful settings.

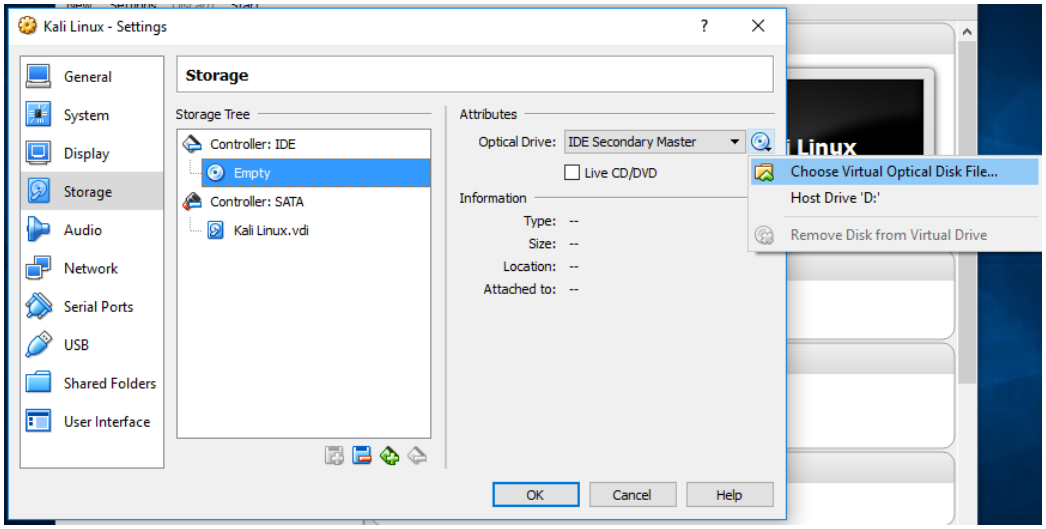


Figure 2.14 *Storage Settings*

In the Storage screen (Figure 2.14, "Storage Settings" [page 33]), you should associate the Kali Linux ISO image with the virtual CD/DVD-ROM reader. First, select the CD-ROM drive in the Storage Tree list and then click on the small CD-ROM icon on the right to display a contextual menu where you can Choose Virtual Optical Disk File....

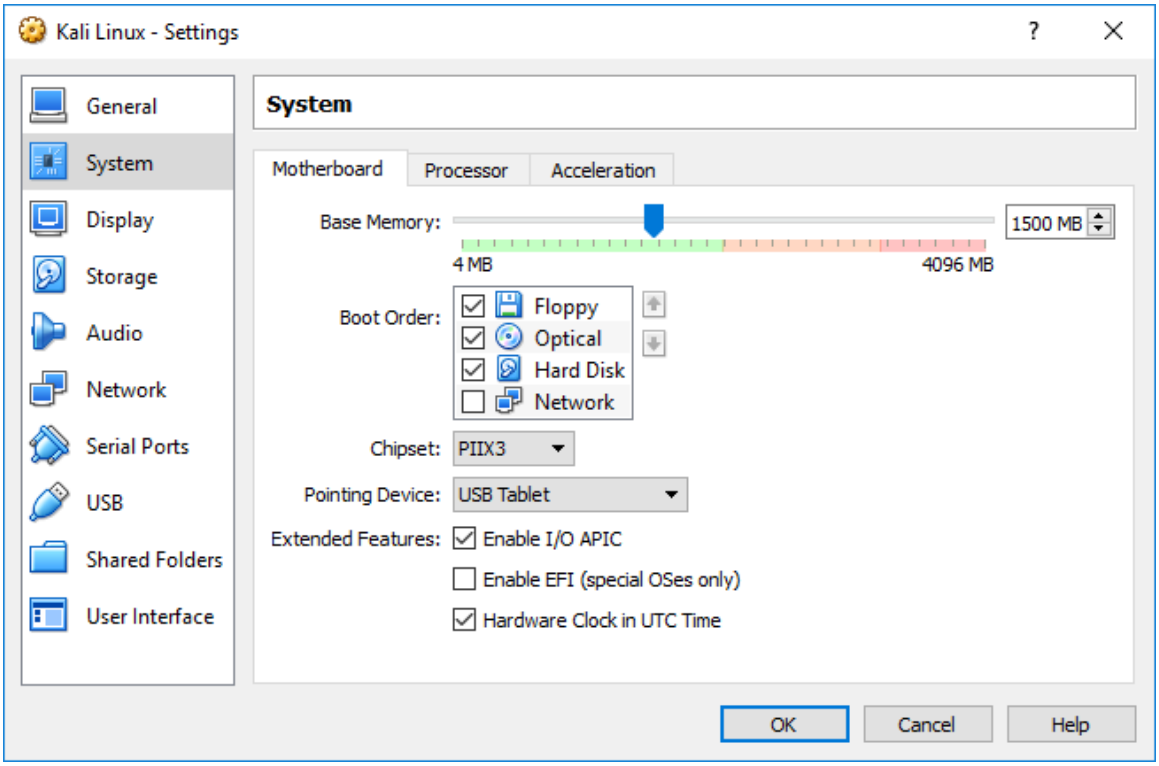


Figure 2.15 *System Settings: Motherboard*

In the System screen (Figure 2.15, "System Settings: Motherboard" [page 34]), you will find a Motherboard tab. Make sure that the boot order indicates that the system will first try to boot from any optical device before trying a hard disk. This is also the tab where you can alter the amount of memory allocated to the virtual machine, should the need arise.

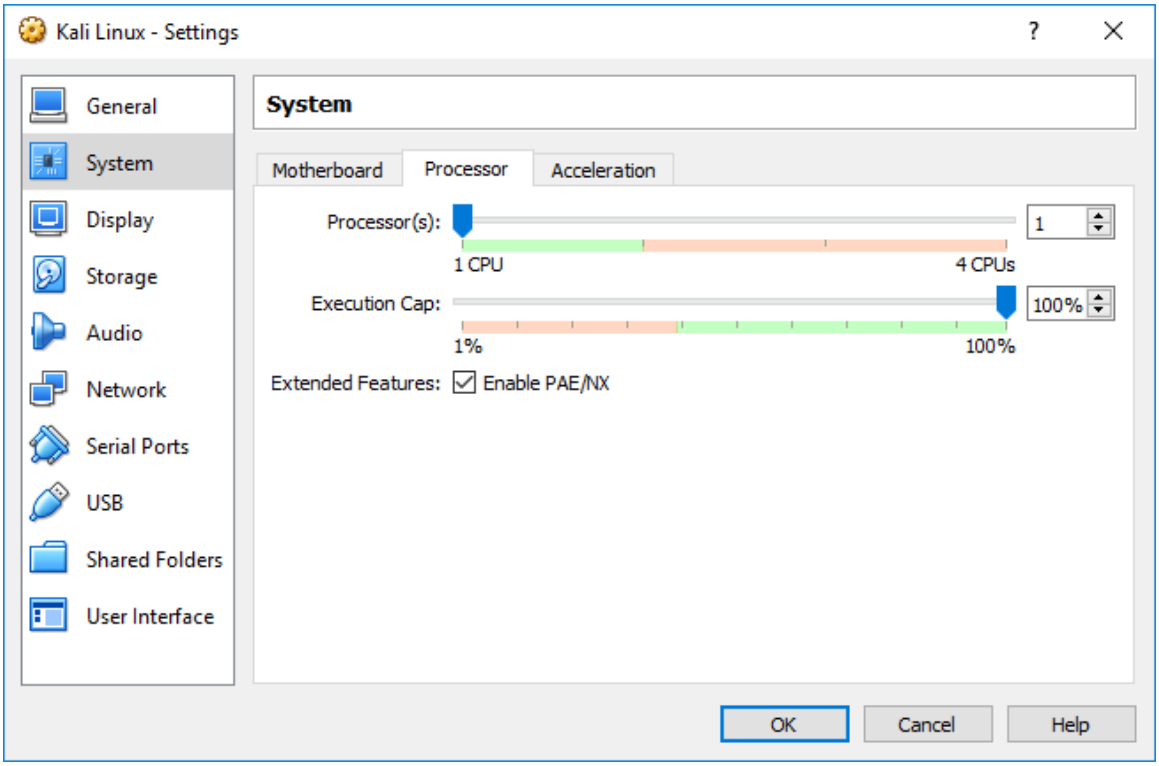


Figure 2.16 *System Settings: Processor*

In the same screen but on the "Processor" tab (Figure 2.16, "System Settings: Processor" [page 35]), you can adjust the number of processors assigned to the virtual machine. Most importantly, if you use a 32-bit image, enable PAE/NX or the Kali image will not boot since the default kernel variant used by Kali for i386 (aptly named "686-pae") is compiled in a way that requires Physical Address Extension (PAE) support in the CPU.

There are many other parameters that can be configured, like the network setup (defining how the traffic on the network card is handled), but the above changes are sufficient to be able to boot a working Kali Linux live system. Finally, click Boot and the VM should boot properly, as shown in Figure 2.17, "Kali Linux Boot Screen in VirtualBox" [page 36]. If not, carefully review all settings and try again.

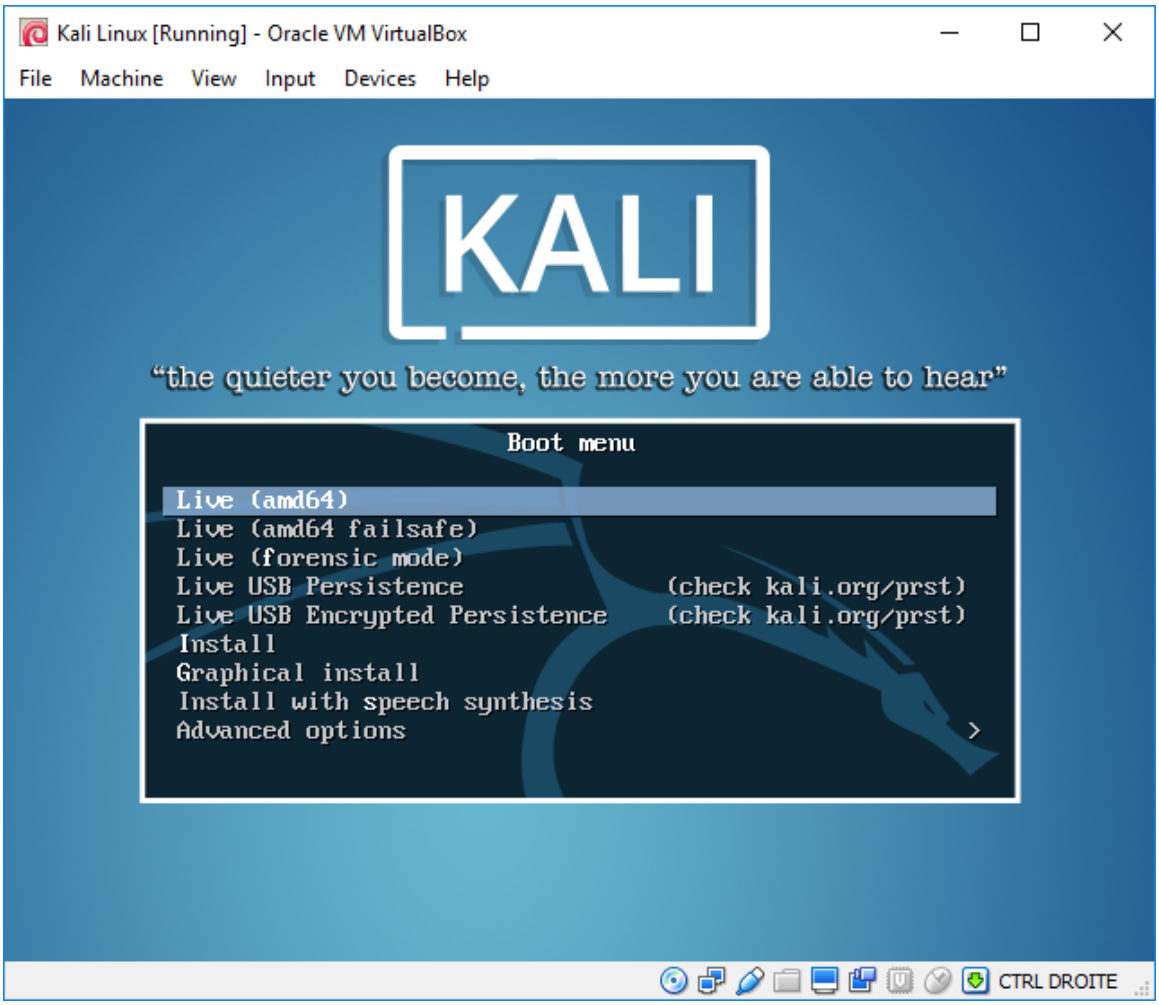


Figure 2.17 *Kali Linux Boot Screen in VirtualBox*

VMware

VMware Workstation Pro is very similar to *VirtualBox* in terms of features and user interface, because they are both designed primarily for desktop usage, but the setup process for a new virtual machine is a bit different.

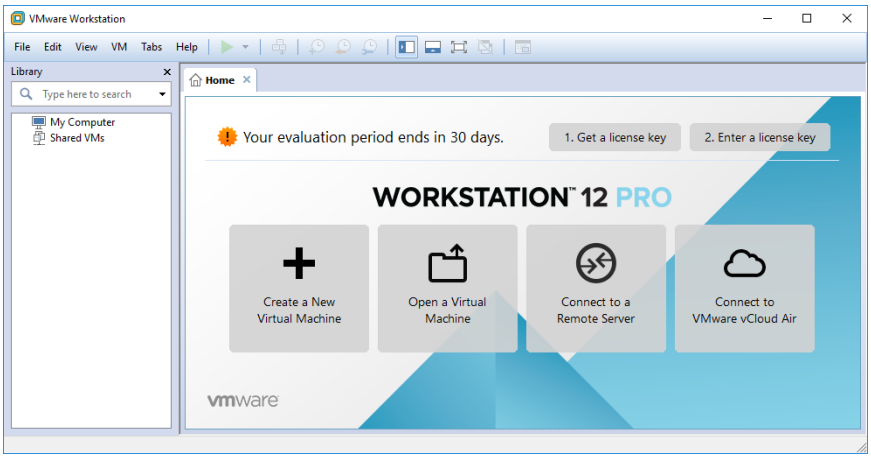


Figure 2.18 *VMware Start Screen*

The initial screen, shown in Figure 2.18, "VMware Start Screen" [page 37], displays a big Create a New Virtual Machine button that starts a wizard to guide you through the creation of your virtual machine.

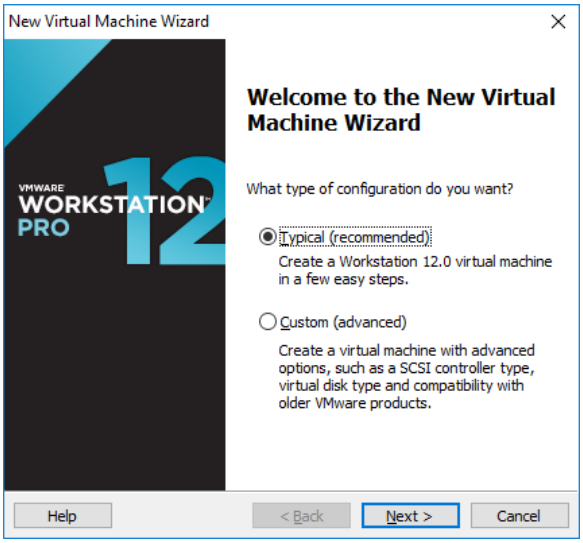


Figure 2.19 *New virtual Machine Wizard*

In the first step, you must decide whether you want to be presented with advanced settings during the setup process. In this example, there are no special requirements so choose the typical installation, as shown in Figure 2.19, "New virtual Machine Wizard" [page 37].

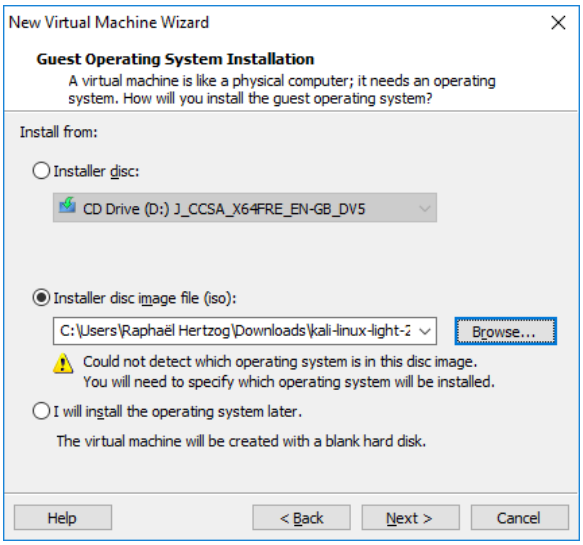


Figure 2.20 *Guest Operating System Installation*

The wizard assumes that you want to install the operating system immediately and asks you to select the ISO image containing the installation program (Figure 2.20, "Guest Operating System Installation" [page 38]). Select "Installer disc image file (iso)" and click on Browse to select the image file.

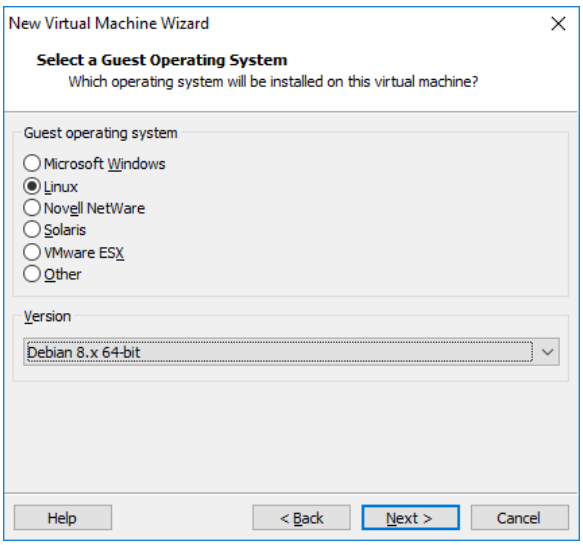


Figure 2.21 *Select a Guest Operating System*

When the operating system (OS) cannot be detected from the selected ISO image, the wizard asks you which guest OS type you intend to run. You should select “Linux” for the OS and “Debian 8.x” for the version, as shown in Figure 2.21, “Select a Guest Operating System” [page 38].

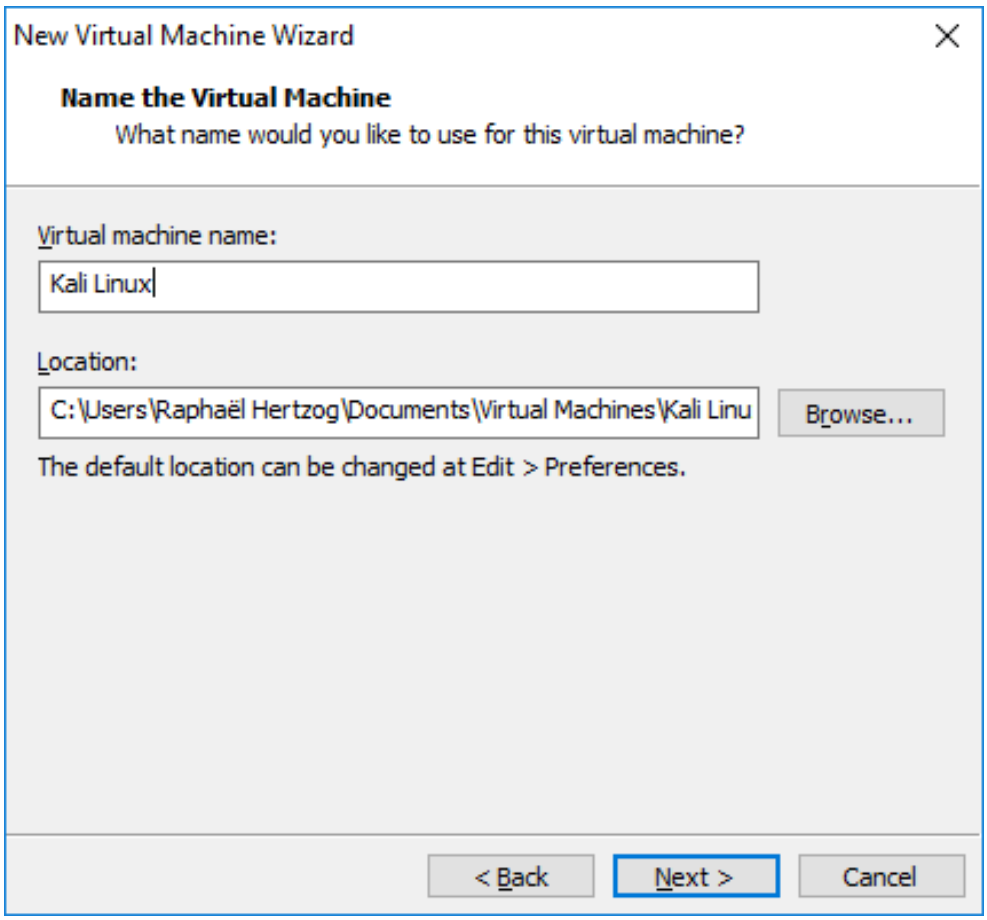


Figure 2.22 *Name the Virtual Machine*

Choose ”Kali Linux” as the name of the new virtual machine (Figure 2.22, “Name the Virtual Machine” [page 39]). As with VirtualBox, you also have the option to store the VM files in an alternate location.

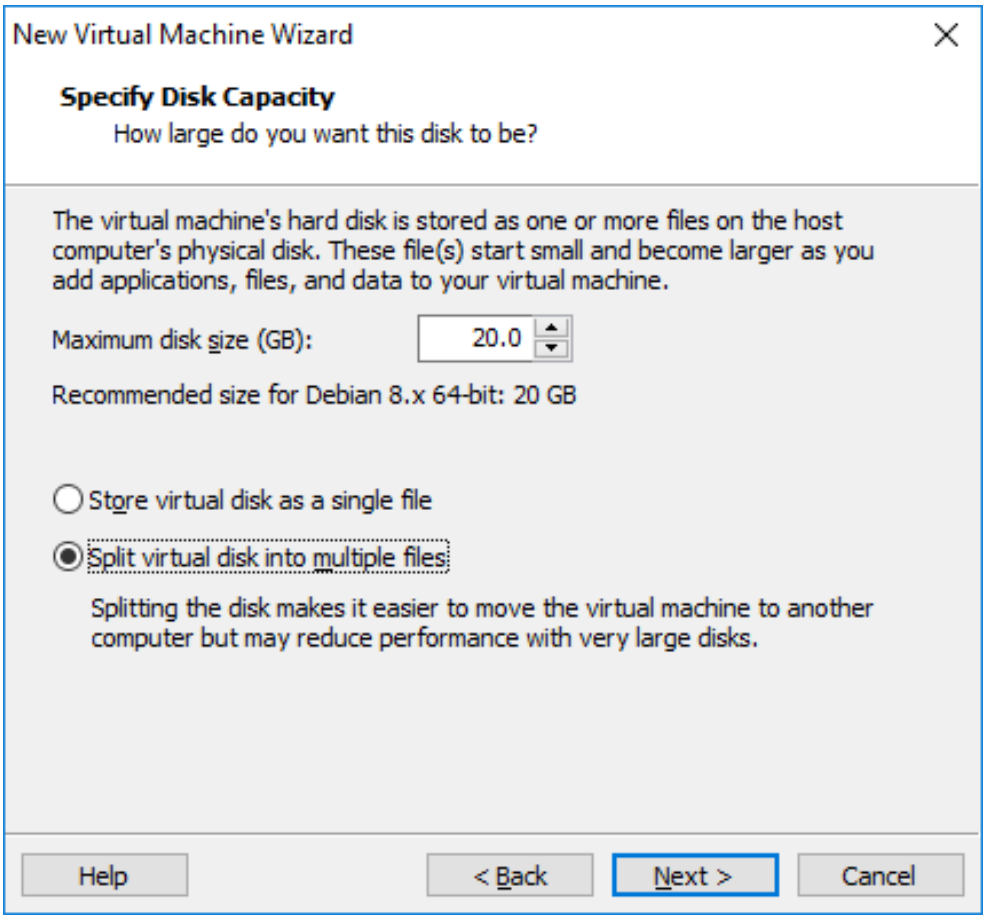


Figure 2.23 *Specify Disk Capacity*

The default hard disk size of 20 GB (Figure 2.23, "Specify Disk Capacity" [page 40]) is usually sufficient but you can adjust it here depending on your expected needs. As opposed to VirtualBox, which can use a single file of varying size, VMware has the ability to store the disk's content over multiple files. In both cases, the goal is to conserve the host's disk space.

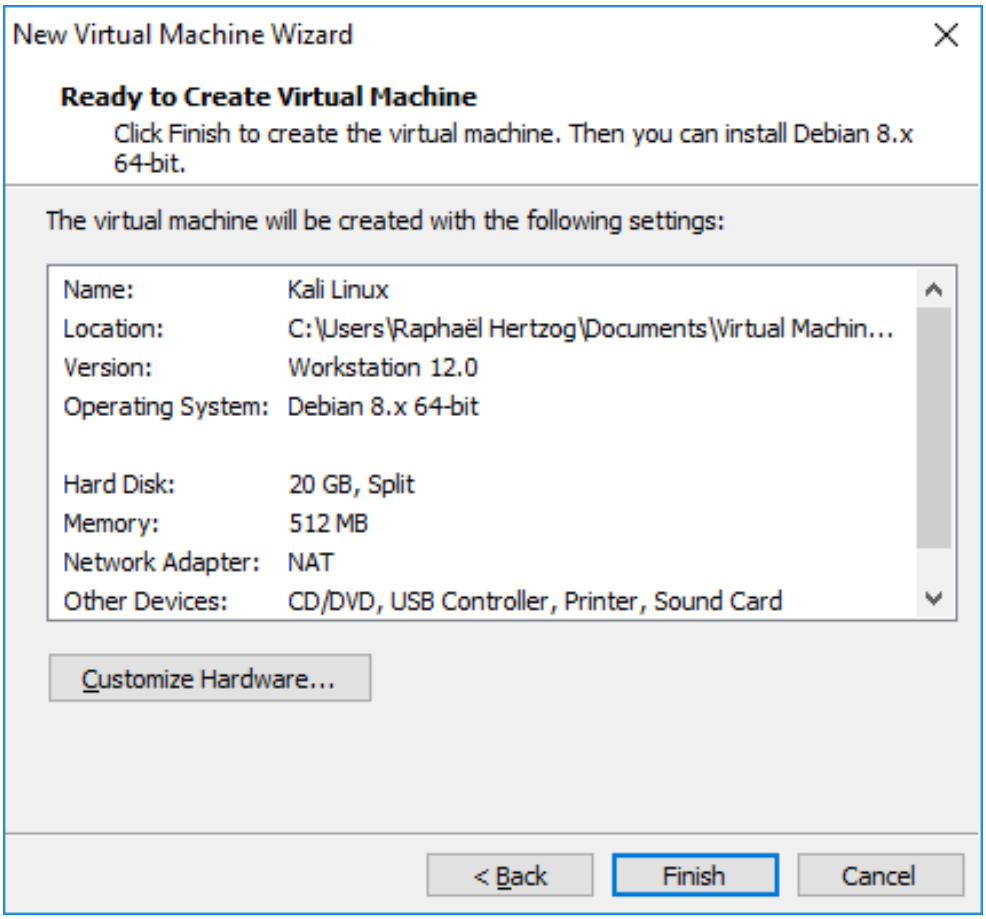


Figure 2.24 *Ready to Create Virtual Machine*

VMware Workstation is now configured to create the new virtual machine. It displays a summary of the choices made so that you can double-check everything before creating the machine. Notice that the wizard opted to allocate only 512 MB of RAM to the virtual machine, which is not enough so click on Customize Hardware... (Figure 2.24, "Ready to Create Virtual Machine" [page 41]) and tweak the Memory setting, as shown in Figure 2.25, "Configure Hardware Window" [page 42].

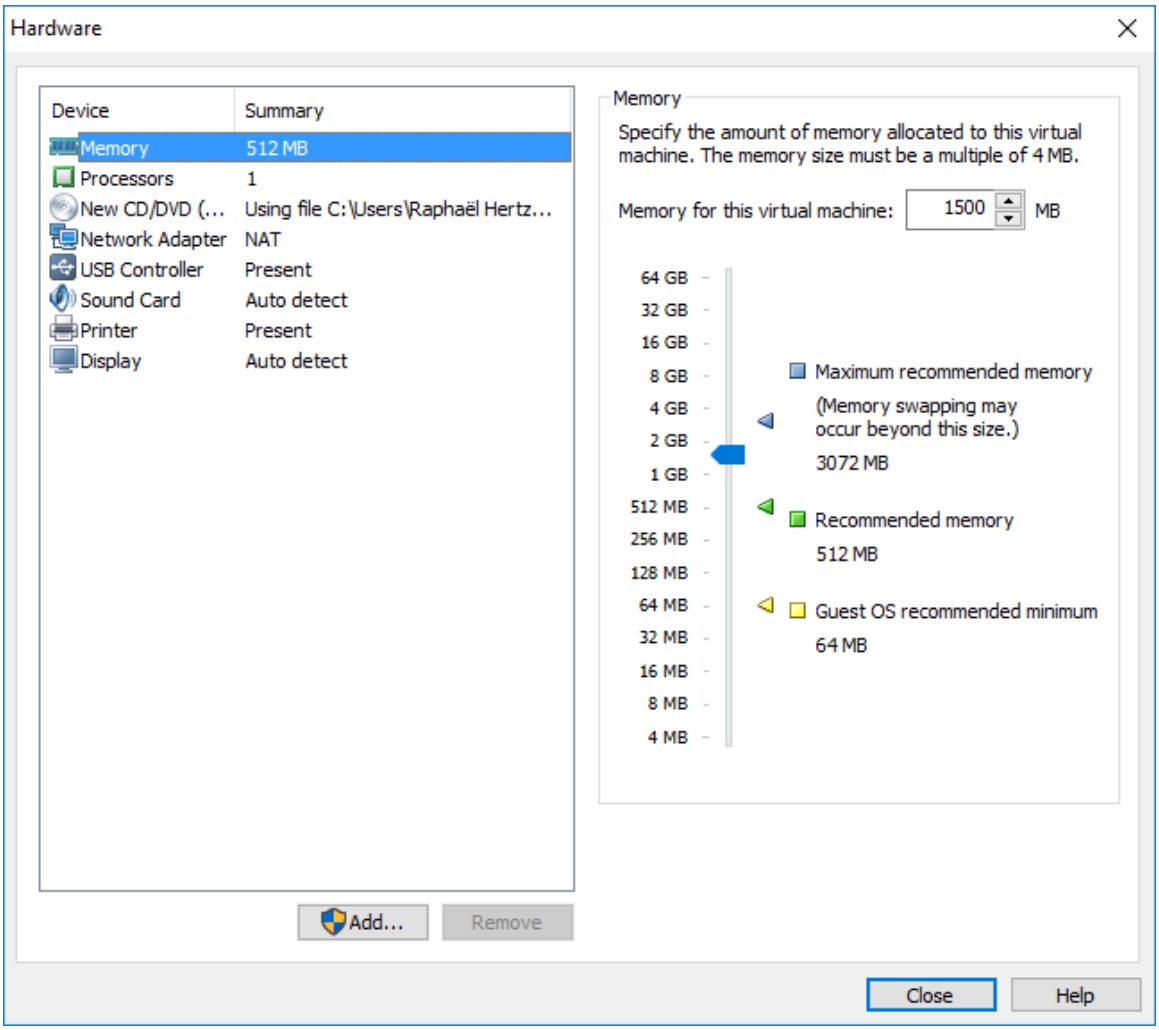


Figure 2.25 *Configure Hardware Window*

After a last click on Finish (Figure 2.24, “Ready to Create Virtual Machine” [page 41]), the virtual machine is now configured and can be started by clicking ”Power on this virtual machine” as shown in Figure 2.26, “Kali Linux Virtual Machine Ready” [page 43].

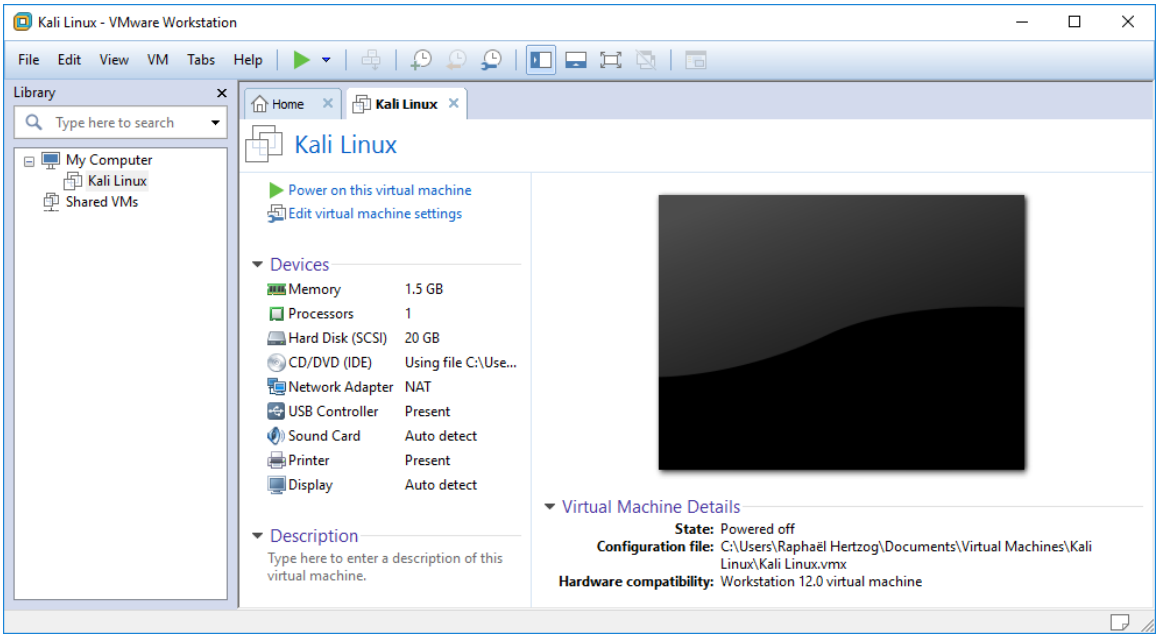


Figure 2.26 *Kali Linux Virtual Machine Ready*

2.3. Summary

In this chapter, you learned about the various Kali Linux ISO images, learned how to verify and download them, and learned how to create bootable USB disks from them on various operating systems. We also discussed how to boot the USB disks and reviewed how to configure the BIOS and startup settings on various hardware platforms so that the USB disks will boot.

Summary Tips:

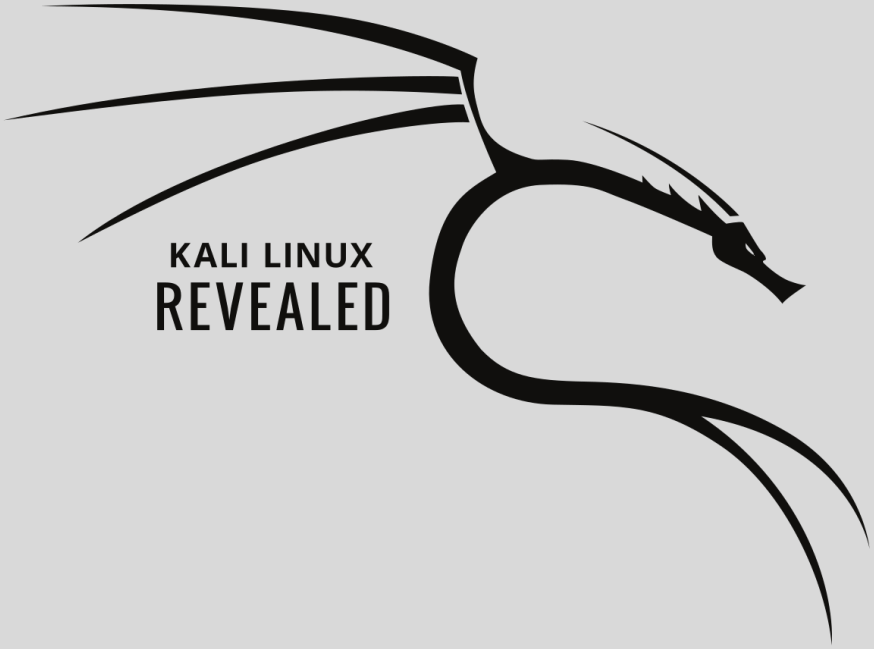
- www.kali.org is the only official download site for Kali ISOs. Do not download them from any other site, because those downloads could contain malware.
- Always validate the sha256sum of your downloads with the `sha256sum` command to ensure the integrity of your ISO download. If it doesn't match, try the download again or use a different source.
- You must write the Kali Linux ISO image to a bootable media if you want to boot it on a physical machine. Use *Win32 Disk Imager* on Windows, the Disks utility on Linux, or the `dd` command on Mac OS X/macOS. Be *very careful* when writing the image. Selecting the wrong disk could permanently damage data on your machine.
- Configure the BIOS/UEFI setup screens on a PC or hold the Option key on OS X/macOS to allow the machine to boot from the USB drive.

- Virtual machine programs like *VirtualBox* and *VMware Workstation Pro* are especially useful if you want to try out Kali Linux but aren't ready to commit to installing it permanently on your machine or if you have a powerful system and want to run multiple operating systems simultaneously.

Now that you have a working installation of Kali Linux, it is time to delve into some Linux fundamentals that are required for basic and advanced operation of Kali. If you are a moderate to advanced Linux user, consider skimming the next chapter.

Keywords

Linux kernel
User space
Command line
bash
Filesystem Hierarchy
Unix commands



Chapter

3 Linux Fundamentals

Contents

Before you can master Kali Linux, you must be at ease with a generic Linux system. Linux proficiency will serve you well, because a large percentage of web, email, and other Internet services run on Linux servers.

In this section, we strive to cover the basics of Linux, but we assume that you already know about computer systems in general, including components such as the CPU, RAM, motherboard, and hard disk, as well as device controllers and their associated connectors.

3.1. What Is Linux and What Is It Doing?

The term "Linux" is often used to refer to the entire operating system, but in reality, Linux is the operating system kernel, which is started by the boot loader, which is itself started by the BIOS/UEFI. The kernel assumes a role similar to that of a conductor in an orchestra—it ensures coordination between hardware and software. This role includes managing hardware, processes, users, permissions, and the file system. The kernel provides a common base to all other programs on the system and typically runs in *ring zero*, also known as *kernel space*.

The User Space

We use the term *user space* to lump together everything that happens outside of the kernel.

Among the programs running in user space are many core utilities from the GNU project[1], most of which are meant to be run from the command line. You can use them in scripts to automate many tasks. Refer to section 3.4, "Useful Commands" [page 56] for more information about the most important commands.

Let's quickly review the various tasks handled by the Linux kernel.

3.1.1. Driving Hardware

The kernel is tasked, first and foremost, with controlling the computer's hardware components. It detects and configures them when the computer powers on, or when a device is inserted or removed (for example, a USB device). It also makes them available to higher-level software, through a simplified programming interface, so applications can take advantage of devices without having to address details such as which extension slot an option board is plugged into. The programming interface also provides an abstraction layer; this allows video-conferencing software, for example, to use a webcam regardless of its maker and model. The software can use the *Video for Linux* (V4L) interface and the kernel will translate function calls of the interface into actual hardware commands needed by the specific webcam in use.

The kernel exports data about detected hardware through the `/proc/` and `/sys/` virtual file systems. Applications often access devices by way of files created within `/dev/`. Specific files rep-

[1] `http://www.gnu.org`

resent disk drives (for instance, `/dev/sda`), partitions (`/dev/sda1`), mice (`/dev/input/mouse0`), keyboards (`/dev/input/event0`), sound cards (`/dev/snd/*`), serial ports (`/dev/ttyS*`), and other components.

There are two types of device files: *block* and *character*. The former has characteristics of a block of data: It has a finite size, and you can access bytes at any position in the block. The latter behaves like a flow of characters. You can read and write characters, but you cannot seek to a given position and change arbitrary bytes. To find out the type of a given device file, inspect the first letter in the output of `ls -l`. It is either b, for block devices, or c, for character devices:

```
$ ls -l /dev/sda /dev/ttyS0
brw-rw---- 1 root disk    8,  0 Mar 21 08:44 /dev/sda
crw-rw---- 1 root dialout 4, 64 Mar 30 08:59 /dev/ttyS0
```

As you might expect, disk drives and partitions use block devices, whereas mouse, keyboard, and serial ports use character devices. In both cases, the programming interface includes device-specific commands that can be invoked through the *ioctl* system call.

3.1.2. Unifying File Systems

File systems are a prominent aspect of the kernel. Unix-like systems merge all the file stores into a single hierarchy, which allows users and applications to access data by knowing its location within that hierarchy.

The starting point of this hierarchical tree is called the root, represented by the "`/`" character. This directory can contain named subdirectories. For instance, the home subdirectory of `/` is called `/home/`. This subdirectory can, in turn, contain other subdirectories, and so on. Each directory can also contain files, where the data will be stored. Thus, `/home/buxy/Desktop/hello.txt` refers to a file named hello.txt stored in the Desktop subdirectory of the buxy subdirectory of the home directory, present in the root. The kernel translates between this naming system and the storage location on a disk.

Unlike other systems, Linux possesses only one such hierarchy, and it can integrate data from several disks. One of these disks becomes the root, and the others are *mounted* on directories in the hierarchy (the Linux command is called `mount`). These other disks are then available under the *mount points*. This allows storing users' home directories (traditionally stored within `/home/`) on a separate hard disk, which will contain the buxy directory (along with home directories of other users). Once you mount the disk on `/home/`, these directories become accessible at their usual locations, and paths such as `/home/buxy/Desktop/hello.txt` keep working.

There are many file system formats, corresponding to many ways of physically storing data on disks. The most widely known are *ext2*, *ext3*, and *ext4*, but others exist. For instance, *VFAT* is the filesystem that was historically used by DOS and Windows operating systems. Linux's support for VFAT allows hard disks to be accessible under Kali as well as under Windows. In any case, you must prepare a file system on a disk before you can mount it and this operation is known as *formatting*.

Commands such as `mkfs.ext3` (where `mkfs` stands for *MaKe FileSystem*) handle formatting. These commands require, as a parameter, a device file representing the partition to be formatted (for instance, `/dev/sda1`, the first partition on the first drive). This operation is destructive and should be run only once, unless you want to wipe a filesystem and start fresh.

There are also network filesystems such as NFS, which do not store data on a local disk. Instead, data are transmitted through the network to a server that stores and retrieves them on demand. Thanks to the file system abstraction, you don't have to worry about how this disk is connected, since the files remain accessible in their usual hierarchical way.

3.1.3. Managing Processes

A process is a running instance of a program, which requires memory to store both the program itself and its operating data. The kernel is in charge of creating and tracking processes. When a program runs, the kernel first sets aside some memory, loads the executable code from the file system into it, and then starts the code running. It keeps information about this process, the most visible of which is an identification number known as the *process identifier* (PID).

Like most modern operating systems, those with Unix-like kernels, including Linux, are capable of multi-tasking. In other words, they allow the system to run many processes at the same time. There is actually only one running process at any one time, but the kernel divides CPU time into small slices and runs each process in turn. Since these time slices are very short (in the millisecond range), they create the appearance of processes running in parallel, although they are active only during their time interval and idle the rest of the time. The kernel's job is to adjust its scheduling mechanisms to keep that appearance, while maximizing global system performance. If the time slices are too long, the application may not appear as responsive as desired. Too short, and the system loses time by switching tasks too frequently. These decisions can be refined with process priorities, where high-priority processes will run for longer periods and with more frequent time slices than low-priority processes.

Multi-Processor Systems (and Variants)

The limitation described above, of only one process running at a time, doesn't always apply: the actual restriction is that there can be only one running process *per processor core*. Multi-processor, multi-core, or *hyper-threaded* systems allow several processes to run in parallel. The same time-slicing system is used, though, to handle cases where there are more active processes than available processor cores. This is not unusual: a basic system, even a mostly idle one, almost always has tens of running processes.

The kernel allows several independent instances of the same program to run, but each is allowed to access only its own time slices and memory. Their data thus remain independent.

3.1.4. Rights Management

Unix-like systems support multiple users and groups and allow control of permissions. Most of the time, a process is identified by the user who started it. That process is only permitted to take actions permitted for its owner. For instance, opening a file requires the kernel to check the process identity against access permissions (for more details on this particular example, see section 3.4.4, “Managing Rights” [page 57]).

3.2. The Command Line

By “command line”, we mean a text-based interface that allows you to enter commands, execute them, and view the results. You can run a terminal (a textual screen within the graphical desktop, or the text console itself outside of any graphical interface) and a command interpreter inside it (*the shell*).

3.2.1. How To Get a Command Line

When your system is working properly, the easiest way to access the command line is to run a terminal in your graphical desktop session.

Figure 3.1 *Starting GNOME Terminal*

For instance, on a default Kali Linux system, GNOME Terminal can be started from the list of favorite applications. You can also type "terminal" while in the Activities screen (the one that gets activated when you move the mouse to the top-left corner) and click on the correct application icon that appears (Figure 3.1, "Starting GNOME Terminal" [page 51]).

In the event that your graphical interface is broken, you can still get a command line on virtual consoles (up to six of them can be accessible through the six key combinations of CTRL+ALT+F1 through CTRL+ALT+F6 — the CTRL key can be omitted if you are already in text mode, outside of Xorg or Wayland's graphical interface). You get a very basic login screen where you enter your login and password before being granted access to the command line with its shell:

```
Kali GNU/Linux Rolling kali-rolling tty3
kali-rolling login: root
Password:
Last login: Fir Mar 25 12:30:05 EDT 2016 from 192.168.122.1 on pts/2
Linux kali-rolling 4.4.0-kali1-amd4 #1 SMP Debian 4.4.6-1kali1 (2016-03-18) 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.
root@kali-rolling:~#
```

The program handling your input and executing your commands is called a *shell* (or a command-line interpreter). The default shell provided in Kali Linux is *Bash* (it stands for *Bourne Again SHell*). The trailing "$" or "#" character indicates that the shell is awaiting your input. It also indicates whether Bash recognizes you as a normal user (the former case with the dollar) or as a super user (the latter case with the hash).

3.2.2. Command Line Basics: Browsing the Directory Tree and Managing Files

This section only provides a brief overview of the covered commands, all of which have many options not described here, so please refer to the abundant documentation available in their respective manual pages. In penetration tests, you will most often receive shell access to a system after a successful exploit, rather than a graphical user interface. Proficiency with the command line is essential for your success as a security professional.

Once a session is open, the `pwd` command (which stands for *print working directory*) displays your current location in the filesystem. The current directory is changed with the `cd` *`directory`* command (`cd` is for *change directory*). When you don't specify the target directory, you are taken to your home directory. When you use `cd -`, you go back to the former working directory (the one in use before the last `cd` call). The parent directory is always called `..` (two dots), whereas the

current directory is also known as . (one dot). The `ls` command allows *listing* the contents of a directory. If you don't provide parameters, `ls` operates on the current directory.

```
$ pwd
/home/buxy
$ cd Desktop
$ pwd
/home/buxy/Desktop
$ cd .
$ pwd
/home/buxy/Desktop
$ cd ..
$ pwd
/home/buxy
$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

You can create a new directory with `mkdir` *`directory`*, and remove an existing (empty) directory with `rmdir` *`directory`*. The `mv` command allows *moving* and renaming files and directories; *removing* a file is achieved with `rm` *`file`*, and copying a file is done with `cp` *`source-file target-file`*.

```
$ mkdir test
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public    test
$ mv test new
$ ls
Desktop    Downloads  new       Public     Videos
Documents  Music      Pictures  Templates
$ rmdir new
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public
```

The shell executes each command by running the first program of the given name that it finds in a directory listed in the PATH environment variable. Most often, these programs are in `/bin`, `/sbin`, `/usr/bin`, or `/usr/sbin`. For example, the `ls` command is found in `/bin/ls`; the `which` command reports the location of a given executable. Sometimes the command is directly handled by the shell, in which case, it is called a shell built-in command (`cd` and `pwd` are among those); the `type` command lets you query the type of each command.

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
$ which ls
/bin/ls
```

```
$ type rm
rm is /bin/rm
$ type cd
cd is a shell builtin
```

Note the usage of the `echo` command, which simply displays a string on the terminal. In this case, it is used to print the contents of an environment variable since the shell automatically substitutes variables with their values before executing the command line.

Environment Variables

Environment variables allow storage of global settings for the shell or various other programs. They are contextual but inheritable. For example, each process has its own set of environment variables (they are contextual). Shells, like login shells, can declare variables, which will be passed down to other programs they execute (they are inheritable).

These variables can be defined system-wide in `/etc/profile` or per-user in `~/.profile` but variables that are not specific to command line interpreters are better put in `/etc/environment`, since those variables will be injected into all user sessions thanks to a Pluggable Authentication Module (PAM) – even when no shell is executed.

3.3. The File System

3.3.1. The Filesystem Hierarchy Standard

As with other Linux distributions, Kali Linux is organized to be consistent with the *Filesystem Hierarchy Standard* (FHS), allowing users of other Linux distributions to easily find their way around Kali. The FHS defines the purpose of each directory. The top-level directories are described as follows.

- `/bin/`: basic programs
- `/boot/`: Kali Linux kernel and other files required for its early boot process
- `/dev/`: device files
- `/etc/`: configuration files
- `/home/`: user's personal files
- `/lib/`: basic libraries
- `/media/*`: mount points for removable devices (CD-ROM, USB keys, and so on)
- `/mnt/`: temporary mount point
- `/opt/`: extra applications provided by third parties
- `/root/`: administrator's (root's) personal files

- `/run/`: volatile runtime data that does not persist across reboots (not yet included in the FHS)
- `/sbin/`: system programs
- `/srv/`: data used by servers hosted on this system
- `/tmp/`: temporary files (this directory is often emptied at boot)
- `/usr/`: applications (this directory is further subdivided into `bin`, `sbin`, `lib` according to the same logic as in the root directory) Furthermore, `/usr/share/` contains architecture-independent data. The `/usr/local/` directory is meant to be used by the administrator for installing applications manually without overwriting files handled by the packaging system (`dpkg`).
- `/var/`: variable data handled by daemons. This includes log files, queues, spools, and caches.
- `/proc/` and `/sys/` are specific to the Linux kernel (and not part of the FHS). They are used by the kernel for exporting data to user space.

3.3.2. The User's Home Directory

The contents of a user's home directory are not standardized but there are still a few noteworthy conventions. One is that a user's home directory is often referred to by a tilde ("~"). That is useful to know because command interpreters automatically replace a tilde with the correct directory (which is stored in the HOME environment variable, and whose usual value is `/home/`*`user`*`/`).

Traditionally, application configuration files are often stored directly under your home directory, but the filenames usually start with a dot (for instance, the `mutt` email client stores its configuration in `~/.muttrc`). Note that filenames that start with a dot are hidden by default; the `ls` command only lists them when the -a option is used and graphical file managers need to be explicitly configured to display hidden files.

Some programs also use multiple configuration files organized in one directory (for instance, `~/.ssh/`). Some applications (such as the Firefox web browser) also use their directory to store a cache of downloaded data. This means that those directories can end up consuming a lot of disk space.

These configuration files stored directly in your home directory, often collectively referred to as *dotfiles*, have long proliferated to the point that these directories can be quite cluttered with them. Fortunately, an effort led collectively under the FreeDesktop.org umbrella has resulted in the XDG Base Directory Specification, a convention that aims at cleaning up these files and directories. This specification states that configuration files should be stored under `~/.config`, cache files under `~/.cache`, and application data files under `~/.local` (or subdirectories thereof). This convention is slowly gaining traction.

Graphical desktops usually have shortcuts to display the contents of the `~/Desktop/` directory (or whatever the appropriate translation is for systems not configured in English).

Finally, the email system sometimes stores incoming emails into a `~/Mail/` directory.

3.4. Useful Commands

3.4.1. Displaying and Modifying Text Files

The `cat` *`file`* command (intended to *concatenate* files to the standard output device) reads a file and displays its contents on the terminal. If the file is too big to fit on a screen, you can use a pager such as `less` (or `more`) to display it page by page.

The `editor` command starts a text editor (such as Vi or Nano) and allows creating, modifying, and reading text files. The simplest files can sometimes be created directly from the command interpreter thanks to redirection: *`command`* `>`*`file`* creates a file named *file* containing the output of the given command. *`command`* `>>`*`file`* is similar except that it appends the output of the command to the file rather than overwriting it.

```
$ echo "Kali rules!" > kali-rules.txt
$ cat kali-rules.txt
Kali rules!
$ echo "Kali is the best!" >> kali-rules.txt
$ cat kali-rules.txt
Kali rules!
Kali is the best!
```

3.4.2. Searching for Files and within Files

The `find` *`directory criteria`* command searches for files in the hierarchy under *directory* according to several criteria. The most commonly used criterion is `-name` *filename*, which allows searching for a file by name. You can also use common wildcards such as "*" in the filename search.

```
$ find /etc -name hosts
/etc/hosts
/etc/avahi/hosts
$ find /etc -name "hosts*"
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/avahi/hosts
```

The `grep` *`expression files`* command searches the contents of the files and extracts lines matching the regular expression. Adding the `-r` option enables a recursive search on all files contained in the directory. This allows you to look for a file when you only know a part of its contents.

3.4.3. Managing Processes

The `ps aux` command lists the processes currently running and helps to identify them by showing their PID. Once you know the *PID* of a process, the `kill -signal pid` command allows you to send it a signal (if you own the process). Several signals exist; most commonly used are TERM (a request to terminate gracefully) and KILL (a forced kill).

The command interpreter can also run programs in the background if the command is followed by "&". By using the ampersand, you resume control of the shell immediately even though the command is still running (hidden from view as a background process). The `jobs` command lists the processes running in the background; running `fg %job-number` (for *foreground*) restores a job to the foreground. When a command is running in the foreground (either because it was started normally, or brought back to the foreground with `fg`), the Control+Z key combination pauses the process and resumes control of the command line. The process can then be restarted in the background with `bg %job-number` (for *background*).

3.4.4. Managing Rights

Linux is a multi-user system so it is necessary to provide a permissions system to control the set of authorized operations on files and directories, which includes all the system resources and devices (on a Unix system, any device is represented by a file or directory). This principle is common to all Unix-like systems.

Each file or directory has specific permissions for three categories of users:

- Its owner (symbolized by `u`, as in user)
- Its owner group (symbolized by `g`, as in group), representing all the members of the group
- The others (symbolized by `o`, as in other)

Three types of rights can be combined:

- reading (symbolized by `r`, as in read);
- writing (or modifying, symbolized by `w`, as in write);
- executing (symbolized by `x`, as in eXecute).

In the case of a file, these rights are easily understood: read access allows reading the content (including copying), write access allows changing it, and execute access allows running it (which will only work if it is a program).

setuid and setgid executables

Two particular rights are relevant to executable files: `setuid` and `setgid` (symbolized with the letter "s"). Note that we frequently speak of bit, since each of these boolean values can be represented by a 0 or a 1. These two rights allow any user to execute the program with the rights of the owner or the group, respectively. This mechanism grants access to features requiring higher level permissions than those you would usually have.

Since a `setuid` root program is systematically run under the super-user identity, it is very important to ensure it is secure and reliable. Any user who manages to subvert a setuid root program to call a command of their choice could then impersonate the root user and have all rights on the system. Penetration testers regularly search for these types of files when they gain access to a system as a way of escalating their privileges.

A directory is handled differently from a file. Read access gives the right to consult the list of its contents (files and directories); write access allows creating or deleting files; and execute access allows crossing through the directory to access its contents (for example, with the `cd` command). Being able to cross through a directory without being able to read it gives the user permission to access the entries therein that are known by name, but not to find them without knowing their exact name.

SECURITY

`setgid` directory and *sticky bit*

The `setgid` bit also applies to directories. Any newly-created item in such directories is automatically assigned the owner group of the parent directory, instead of inheriting the creator's main group as usual. Because of this, you don't have to change your main group (with the `newgrp` command) when working in a file tree shared between several users of the same dedicated group.

The *sticky bit* (symbolized by the letter "t") is a permission that is only useful in directories. It is especially used for temporary directories where everybody has write access (such as `/tmp/`): it restricts deletion of files so that only their owner or the owner of the parent directory can delete them. Lacking this, everyone could delete other users' files in `/tmp/`.

Three commands control the permissions associated with a file:

- `chown` *`user file`* changes the owner of the file

TIP

Changing the user and group

Frequently you want to change the group of a file at the same time that you change the owner. The `chown` command has a special syntax for that: `chown` *`user:group file`*

- `chgrp` *`group file`* alters the owner group
- `chmod` *`rights file`* changes the permissions for the file

There are two ways of representing rights. Among them, the symbolic representation is probably the easiest to understand and remember. It involves the letter symbols mentioned above. You can define rights for each category of users (u/g/o), by setting them explicitly (with =), by adding

(+), or subtracting (-). Thus the u=rwx,g+rw,o-r formula gives the owner read, write, and execute rights, adds read and write rights for the owner group, and removes read rights for other users. Rights not altered by the addition or subtraction in such a command remain unmodified. The letter a, for all, covers all three categories of users, so that a=rx grants all three categories the same rights (read and execute, but not write).

The (octal) numeric representation associates each right with a value: 4 for read, 2 for write, and 1 for execute. We associate each combination of rights with the sum of the three figures, and a value is assigned to each category of users, in the usual order (owner, group, others).

For instance, the `chmod 754 file` command will set the following rights: read, write and execute for the owner (since 7 = 4 + 2 + 1); read and execute for the group (since 5 = 4 + 1); read-only for others. The 0 means no rights; thus `chmod 600 file` allows for read and write permissions for the owner, and no rights for anyone else. The most frequent right combinations are 755 for executable files and directories, and 644 for data files.

To represent special rights, you can prefix a fourth digit to this number according to the same principle, where the setuid, setgid, and sticky bits are 4, 2, and 1, respectively. The command `chmod 4754` will associate the setuid bit with the previously described rights.

Note that the use of octal notation only allows you to set all the rights at once on a file; you cannot use it to add a new right, such as read access for the group owner, since you must take into account the existing rights and compute the new corresponding numerical value.

The octal representation is also used with the `umask` command, which is used to restrict permissions on newly created files. When an application creates a file, it assigns indicative permissions, knowing that the system automatically removes the rights defined with `umask`. Enter `umask` in a shell; you will see a mask such as `0022`. This is simply an octal representation of the rights to be systematically removed (in this case, the write rights for the group and other users).

If you give it a new octal value, the `umask` command modifies the mask. Used in a shell initialization file (for example, `~/.bash_profile`), it will effectively change the default mask for your work sessions.

TIP

Recursive operation

Sometimes we have to change rights for an entire file tree. All the commands above have a `-R` option to operate recursively in sub-directories.

The distinction between directories and files sometimes causes problems with recursive operations. That is why the "X" letter has been introduced in the symbolic representation of rights. It represents a right to execute which applies only to directories (and not to files lacking this right). Thus, `chmod -R a+X directory` will only add execute rights for all categories of users (a) for all of the sub-directories and files for which at least one category of user (even if their sole owner) already has execute rights.

3.4.5. Getting System Information and Logs

The `free` command displays information on memory; *disk free* (`df`) reports on the available disk space on each of the disks mounted in the file system. Its `-h` option (for *human readable*) converts the sizes into a more legible unit (usually mebibytes or gibibytes). In a similar fashion, the `free` command supports the `-m` and `-g` options, and displays its data either in mebibytes or in gibibytes, respectively.

```
$ free
              total        used        free      shared  buff/cache   available
Mem:        2052944      661232      621208       10520      770504     1359916
Swap:             0           0           0
$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             1014584        0   1014584   0% /dev
tmpfs             205296     8940    196356   5% /run
/dev/vda1       30830588 11168116  18073328  39% /
tmpfs            1026472      456   1026016   1% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs            1026472        0   1026472   0% /sys/fs/cgroup
tmpfs             205296       36    205260   1% /run/user/132
tmpfs             205296       24    205272   1% /run/user/0
```

The `id` command displays the identity of the user running the session along with the list of groups they belong to. Since access to some files or devices may be limited to group members, checking available group membership may be useful.

```
$ id
uid=1000(buxy) gid=1000(buxy) groups=1000(buxy),27(sudo)
```

The `uname -a` command returns a single line documenting the kernel name (Linux), the hostname, the kernel release, the kernel version, the machine type (an architecture string such as x86_64), and the name of the operating system (GNU/Linux). The output of this command should usually be included in bug reports as it clearly defines the kernel in use and the hardware platform you are running on.

```
$ uname -a
Linux kali 4.9.0-kali3-amd64 #1 SMP Debian 4.9.18-1kali1 (2017-04-04) x86_64 GNU/Linux
```

All these commands provide run-time information, but often you need to consult logs to understand what happened on your computer. In particular, the kernel emits messages that it stores in a ring buffer whenever something interesting happens (such as a new USB device being inserted, a failing hard disk operation, or initial hardware detection on boot). You can retrieve the kernel logs with the `dmesg` command.

Systemd's journal also stores multiple logs (stdout/stderr output of daemons, syslog messages, kernel logs) and makes it easy to query them with `journalctl`. Without any arguments, it just dumps all the available logs in a chronological way. With the `-r` option, it will reverse the order so that newer messages are shown first. With the `-f` option, it will continuously print new log entries as they are appended to its database. The `-u` option can limit the messages to those emitted by a specific systemd unit (ex: `journalctl -u ssh.service`).

3.4.6. Discovering the Hardware

The kernel exports many details about detected hardware through the `/proc/` and `/sys/` virtual filesystems. Several tools summarize those details. Among them, `lspci` (in the *pciutils* package) lists PCI devices, `lsusb` (in the *usbutils* package) lists USB devices, and `lspcmcia` (in the *pcmciautils* package) lists PCMCIA cards. These tools are very useful for identifying the exact model of a device. This identification also allows more precise searches on the web, which in turn, lead to more relevant documents. Note that the *pciutils* and *usbutils* packages are already installed on the base Kali system but *pcmciautils* must be installed with `apt install pcmciautils`. We will discuss more about package installation and management in a later chapter.

Example 3.1 *Example of information provided by `lspci` and `lsusb`*

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection (rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

These programs have a -v option that lists much more detailed (but usually unnecessary) information. Finally, the `lsdev` command (in the *procinfo* package) lists communication resources used by devices.

The `lshw` program is a combination of the above programs and displays a long description of the hardware discovered in a hierarchical manner. You should attach its full output to any report about hardware support problems.

3.5. Summary

In this section, we took a whirlwind tour of the Linux landscape. We discussed the kernel and user space, reviewed many common Linux shell commands, discussed processes and how to manage them, reviewed user and group security concepts, discussed the FHS, and toured some of the most common directories and files found on Kali Linux.

Summary Tips:

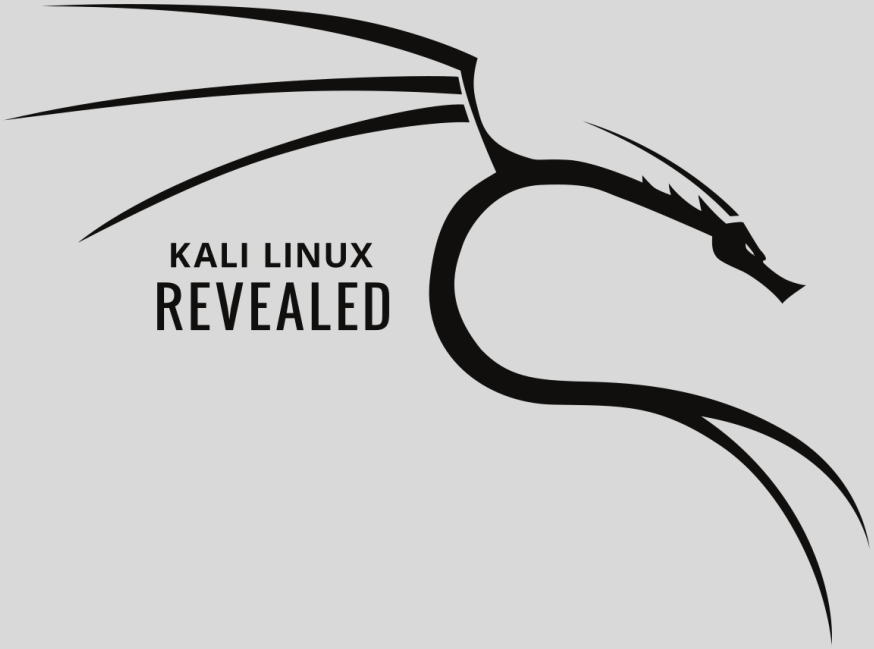
- Linux is often used to refer to the entire operating system but in reality Linux itself is the operating system kernel that is started by the boot loader, which is itself started by the BIOS/UEFI.
- User space refers to everything that happens outside of the kernel. Among the programs running in user space, there are many core utilities from the GNU project[2], most of which are meant to be run from the command line (a text-based interface that allows you to enter commands, execute them, and view the results). A shell executes your commands within that interface.
- Common commands include: `pwd` (print working directory), `cd` (change directory), `ls` (list file or directory contents), `mkdir` (make directory), `rmdir` (remove directory), `mv`, `rm`, and `cp` (move, remove, or copy file or directory respectively), `cat` (concatenate or show file), `less/more` (show files a page at a time), `editor` (start a text editor), `find` (locate a file or directory), `free` (display memory information), `df` (show disk free space), `id` display the identity of the user along with the list of groups they belong to), `dmesg` (review kernel logs), and `journalctl` (show all available logs).
- You can inspect the hardware on a Kali system with several commands: `lspci` (list PCI devices), `lsusb` (list USB devices), and `lspcmcia` lists PCMCIA cards.
- A process is a running instance of a program, which requires memory to store both the program itself and its operating data. You can manage processes with commands like: `ps` (show processes), `kill` (kill processes), `bg` (send process to background), `fg` (bring background process to foreground), and `jobs` (show background processes).
- Unix-like systems are multi-user. They support multiple users and groups and allow control over actions, based on permissions. You can manage file and directory rights with several commands, including: `chmod` (change permissions), `chown` (change owner), and `chgrp` (change group).
- As with other professional Linux distributions, Kali Linux is organized to be consistent with the *Filesystem Hierarchy Standard* (FHS), allowing users coming from other Linux distributions to easily find their way around Kali.
- Traditionally, application configuration files are stored under your home directory, in hidden files or directories starting with a period (or dot).

[2] `http://www.gnu.org`

Now that you have a handle on Linux fundamentals, let's get Kali Linux set up and running.

Keywords

Installation
Unattended installation
ARM devices
Troubleshooting



Chapter

4

Installing Kali Linux

Contents

In this chapter, we will focus on the Kali Linux installation process. First, we will discuss the minimum installation requirements (section 4.1, "Minimal Installation Requirements" [page 66]) to ensure that your real or virtual system is well-configured to handle the type of installation that you will pursue. Then we will go through each step of the installation process (section 4.2, "Step by Step Installation on a Hard Drive" [page 66]) for a plain installation, as well as for a more secure installation involving a fully encrypted file system. We will also discuss *preseeding*, which allows unattended installations (section 4.3, "Unattended Installations" [page 91]) by providing predetermined answers to installation questions. We will also show you how to install Kali Linux on various ARM devices (section 4.4, "ARM Installations" [page 94]), which expands Kali's capabilities far beyond the desktop. Finally, we will show you what to do in the rare case of an installation failure (section 4.5, "Troubleshooting Installations" [page 95]), so you can work through the issue and successfully finish a tough install.

4.1. Minimal Installation Requirements

The installation requirements for Kali Linux vary depending on what you would like to install. On the low end, you can set up Kali as a basic Secure Shell (SSH) server with no desktop, using as little as 128 MB of RAM (512 MB recommended) and 2 GB of disk space. On the higher end, if you opt to install the default GNOME desktop and the *kali-linux-full* meta-package, you should really aim for at least 2048 MB of RAM and 20 GB of disk space.

Besides the RAM and hard disk requirements, your computer needs to have a CPU supported by at least one of the amd64, i386, armel, armhf, or arm64 architectures.

4.2. Step by Step Installation on a Hard Drive

In this section, we assume that you have a bootable USB drive or DVD (see section 2.1.4, "Copying the Image on a DVD-ROM or USB Key" [page 19] for details on how to prepare such a drive) and that you booted from it to start the installation process.

4.2.1. Plain Installation

First, we will take a look at a standard Kali installation, with an unencrypted file system.

Booting and Starting the Installer

Once the BIOS has begun booting from the USB drive or DVD-ROM, the Isolinux boot loader menu appears, as shown in Figure 4.1, "Boot Screen" [page 67]. At this stage, the Linux kernel is not yet loaded; this menu allows you to choose the kernel to boot and enter optional parameters to be transferred to it in the process.

For a standard installation, you only need to choose Install or Graphical Install (with the arrow keys), then press the Enter key to initiate the remainder of the installation process.

Each menu entry hides a specific boot command line, which can be configured as needed by pressing the Tab key before validating the entry and booting.

Figure 4.1 *Boot Screen*

Once booted, the installation program guides you step-by-step through the process. We will take a look at each of these steps in detail. We will cover installation from a standard Kali Linux DVD-ROM; installations from a `mini.iso` may look slightly different. We will also address graphical mode installation, but the only difference from classic text-mode installation is the appearance. The versions pose identical questions and present identical options.

Selecting the Language

As shown in Figure 4.2, "Selecting the Language" [page 68], the installation program begins in English but the first step allows you to choose the language that will be used for the rest of the installation process. This language choice is also used to define more relevant default choices in subsequent stages (notably the keyboard layout).

Navigating with the Keyboard

Some steps in the installation process require you to enter information. These screens have several areas that may gain focus (text entry area, checkboxes, list of choices, OK and Cancel buttons), and the Tab key allows you to move from one to another.

In graphical installation mode, you can use the mouse as you would normally on an installed graphical desktop.

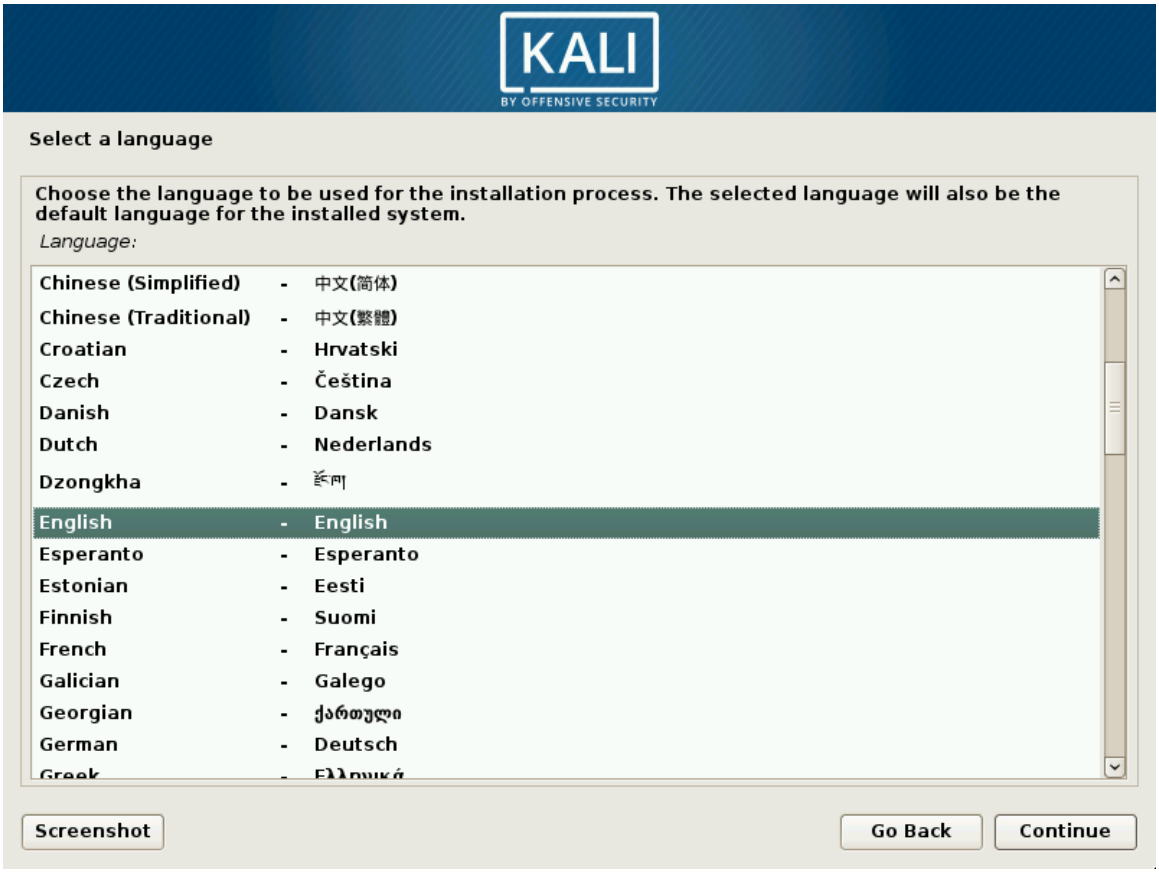


Figure 4.2 *Selecting the Language*

The second step (Figure 4.3, "Selecting the Country" [page 69]) consists in choosing your country. Combined with the language, this information enables the installation program to offer the most appropriate keyboard layout. This will also influence the configuration of the time zone. In the United States, a standard QWERTY keyboard is suggested and the installer presents a choice of appropriate time zones.

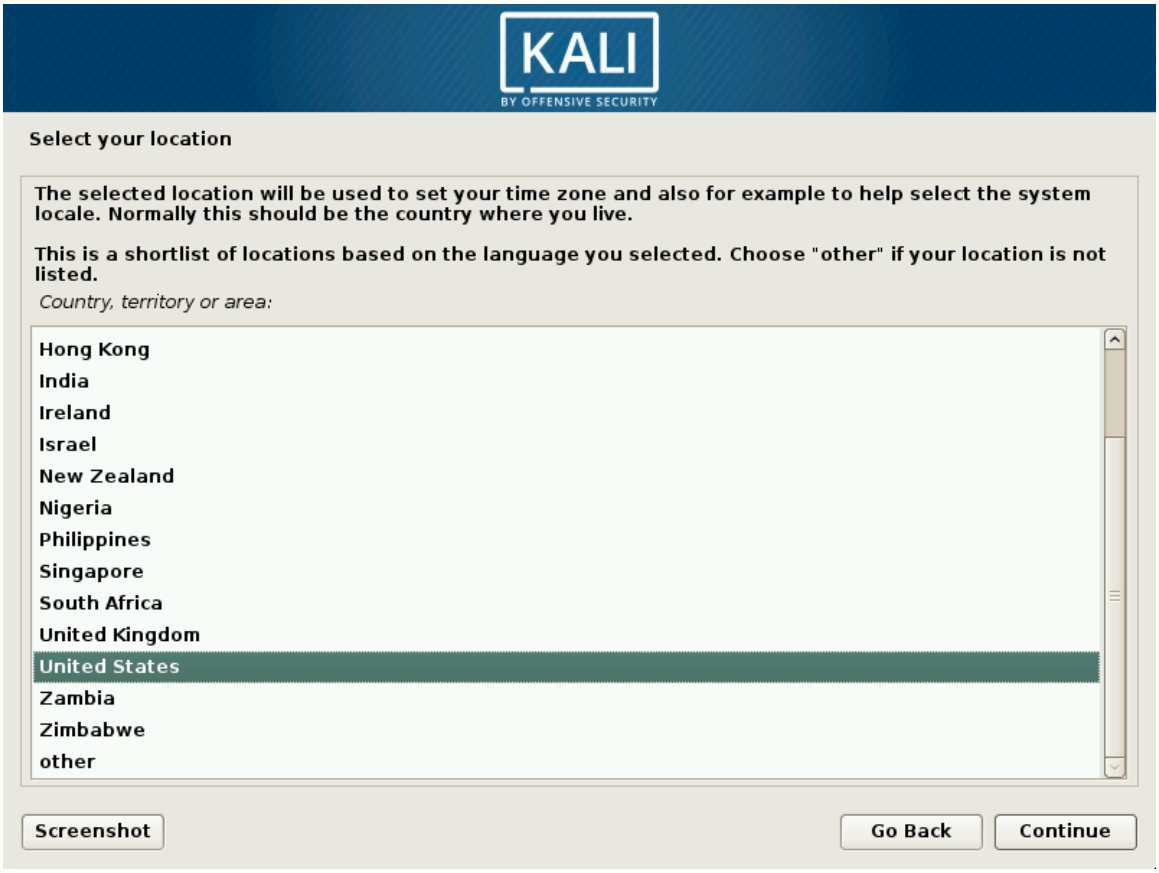


Figure 4.3 *Selecting the Country*

Selecting the Keyboard Layout

The proposed American English keyboard corresponds to the usual QWERTY layout as shown in Figure 4.4, "Choice of Keyboard" [page 70].

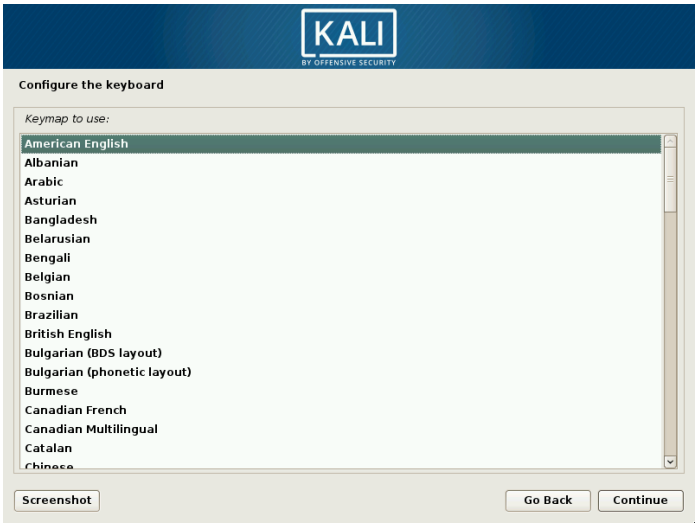


Figure 4.4 *Choice of Keyboard*

Detecting Hardware

In the vast majority of cases, the hardware detection step is completely automatic. The installer detects your hardware and tries to identify the boot device used in order to access its content. It loads the modules corresponding to the various hardware components detected and then mounts the boot device in order to read it. The previous steps were completely contained in the boot image included on the boot device, a file of limited size and loaded into memory by the bootloader when booting from the boot device.

Loading Components

With the contents of the boot device now available, the installer loads all the files necessary to continue with its work. This includes additional drivers for the remaining hardware (especially the network card), as well as all the components of the installation program.

Detecting Network Hardware

In this step, the installer will try to automatically identify the network card and load the corresponding module. If automatic detection fails, you can manually select the module to load. If all else fails, you can load a specific module from a removable device. This last solution is usually only needed if the appropriate driver is not included in the standard Linux kernel, but available elsewhere, such as the manufacturer's website.

This step must absolutely be successful for network installations (such as those done when booting from a `mini.iso`), since the Debian packages must be loaded from the network.

Configuring the Network

In order to automate the process as much as possible, the installer attempts an automatic network configuration using dynamic host configuration protocol (DHCP) (for IPv4 and IPv6) and ICMPv6's Neighbor Discovery Protocol (for IPv6), as shown in Figure 4.5, "Network Autoconfiguration" [page 71].

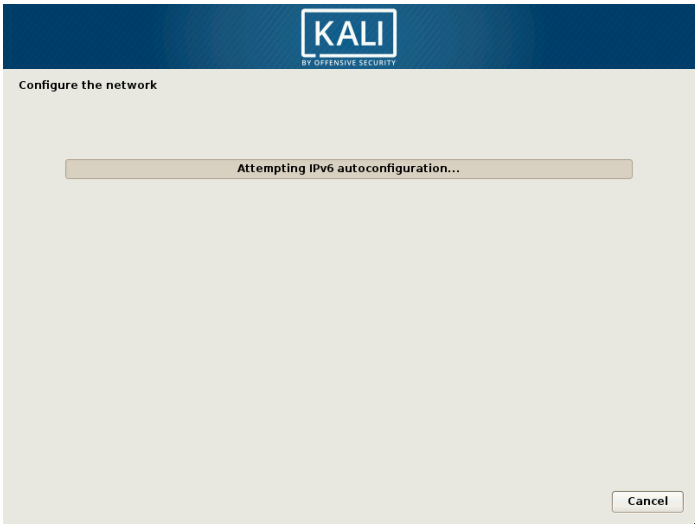


Figure 4.5 *Network Autoconfiguration*

If the automatic configuration fails, the installer offers more choices: try again with a normal DHCP configuration, attempt DHCP configuration by declaring the name of the machine, or set up a static network configuration.

This last option requires an IP address, a subnet mask, an IP address for a potential gateway, a machine name, and a domain name.

Configuration without DHCP

If the local network is equipped with a DHCP server that you do not wish to use because you prefer to define a static IP address for the machine during installation, you can add the **`netcfg/use_dhcp=false`** option when booting. You just need to edit the desired menu entry by pressing the Tab key and adding the desired option before pressing the Enter key.

Root Password

The installer prompts for a password (Figure 4.6, "Root Password" [page 72]) since it automatically creates a super-user root account. The installer also asks for a confirmation of the password to prevent any input error which would later be difficult to adjust.

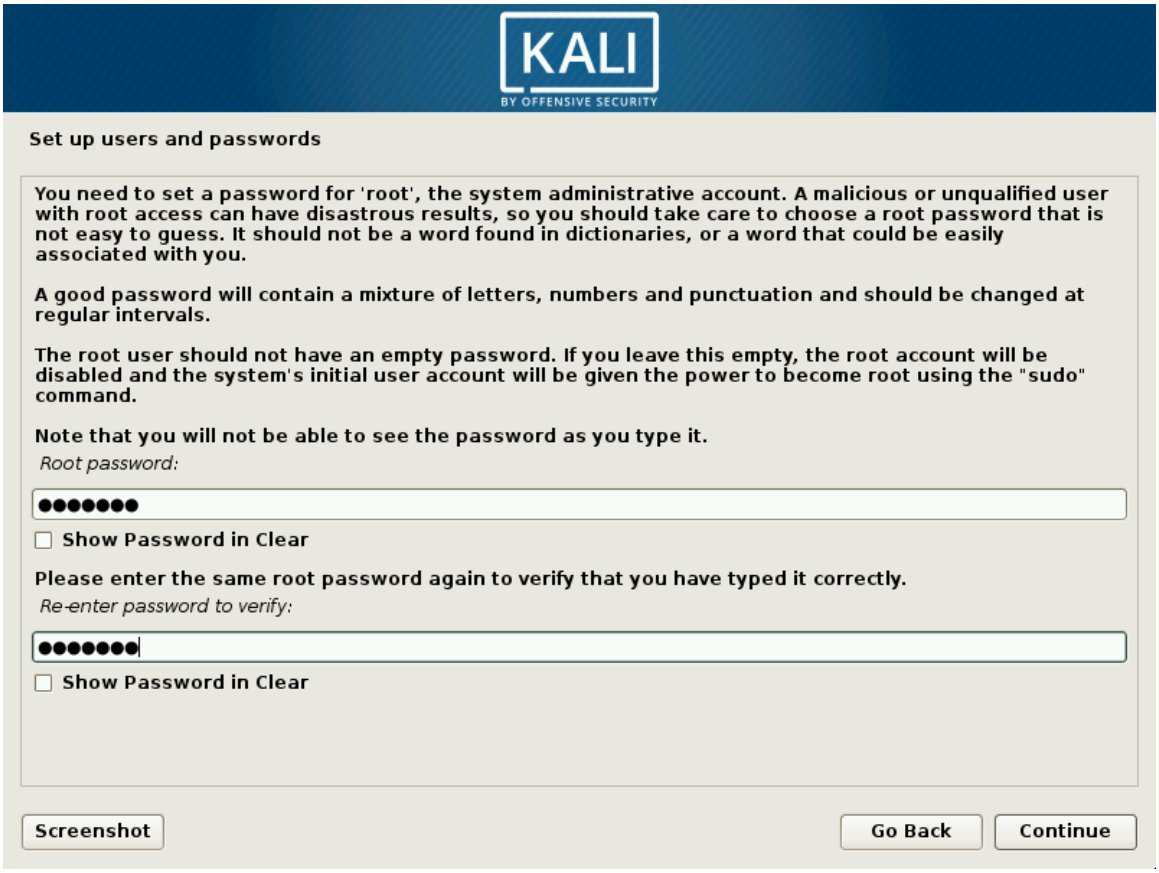


Figure 4.6 *Root Password*

The Administrator Password

The root user's password should be long (eight characters or more) and impossible to guess, since attackers target Internet-connected computers and servers with automated tools, attempting to log in with obvious passwords. Sometimes attackers leverage dictionary attacks, using many combinations of words and numbers as passwords. Avoid using the names of children or parents and dates of birth, because these are easily guessed.

These remarks are equally applicable to other user passwords but the consequences of a compromised account are less drastic for users without administrative rights.

If you are lacking inspiration, don't hesitate to use a password generator, such as `pwgen` (found in the package of the same name, which is already included in the base Kali installation).

Configuring the Clock

If the network is available, the system's internal clock will be updated from a network time protocol (NTP) server. This is beneficial because it ensures timestamps on logs will be correct from the first boot.

If your country spans multiple timezones, you will be asked to select the timezone that you want to use, as shown in Figure 4.7, "Timezone Selection" [page 73].

Figure 4.7 *Timezone Selection*

Detecting Disks and Other Devices

This step automatically detects the hard drives on which Kali may be installed, each of which will be presented in the next step: partitioning.

Partitioning

Partitioning is an indispensable step in installation, which consists of dividing the available space on the hard drives into discrete sections (*partitions*) according to the intended function of the computer and those partitions. Partitioning also involves choosing the file systems to be used. All of these decisions will have an influence on performance, data security, and server administration.

The partitioning step is traditionally difficult for new users. However, the Linux file systems and partitions, including virtual memory (or *swap* partitions) must be defined as they form the foundation of the system. This task can become complicated if you have already installed another operating system on the machine and you want the two to coexist. In this case, you must make sure not to alter its partitions, or if need be, resize them without causing damage.

To accommodate more common (and simpler) partition schemes, most users will prefer the *Guided* mode that recommends partition configurations and provides suggestions each step of the way. More advanced users will appreciate the *Manual* mode, which allows for more advanced configurations. Each mode shares certain capabilities.

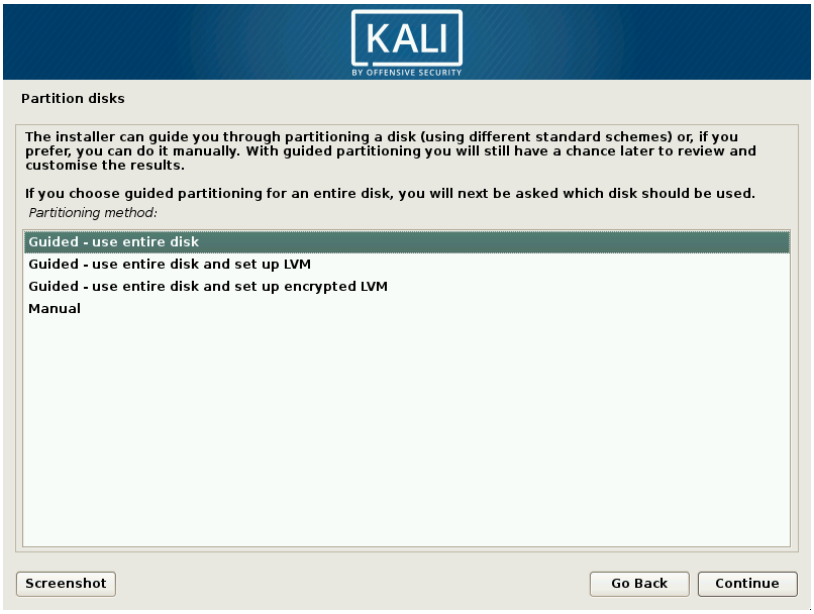


Figure 4.8 *Choice of Partitioning Mode*

Guided Partitioning The first screen in the partitioning tool (Figure 4.8, "Choice of Partitioning Mode" [page 74]) presents entry points for the guided and manual partitioning modes. "Guided - use entire disk" is the simplest and most common partition scheme, which will allocate an entire disk to Kali Linux.

The next two selections use Logical Volume Manager (LVM) to set up logical (instead of physical), optionally encrypted, partitions. We will discuss LVM and encryption later in this chapter.

Finally, the last choice initiates manual partitioning, which allows for more advanced partitioning schemes, such as installing Kali Linux alongside other operating systems. We will discuss manual mode in the next section.

In this example, we will allocate an entire hard disk to Kali, so we select "Guided - use entire disk" to proceed to the next step.

The next screen (shown in Figure 4.9, "Disk to Use for Guided Partitioning" [page 75]) allows you to choose the disk where Kali will be installed by selecting the corresponding entry (for example, "Virtual disk 1 (vda) - 32.2 GB Virtio Block Device"). Once selected, guided partitioning will continue. This option will erase all of the data on this disk, so choose wisely.

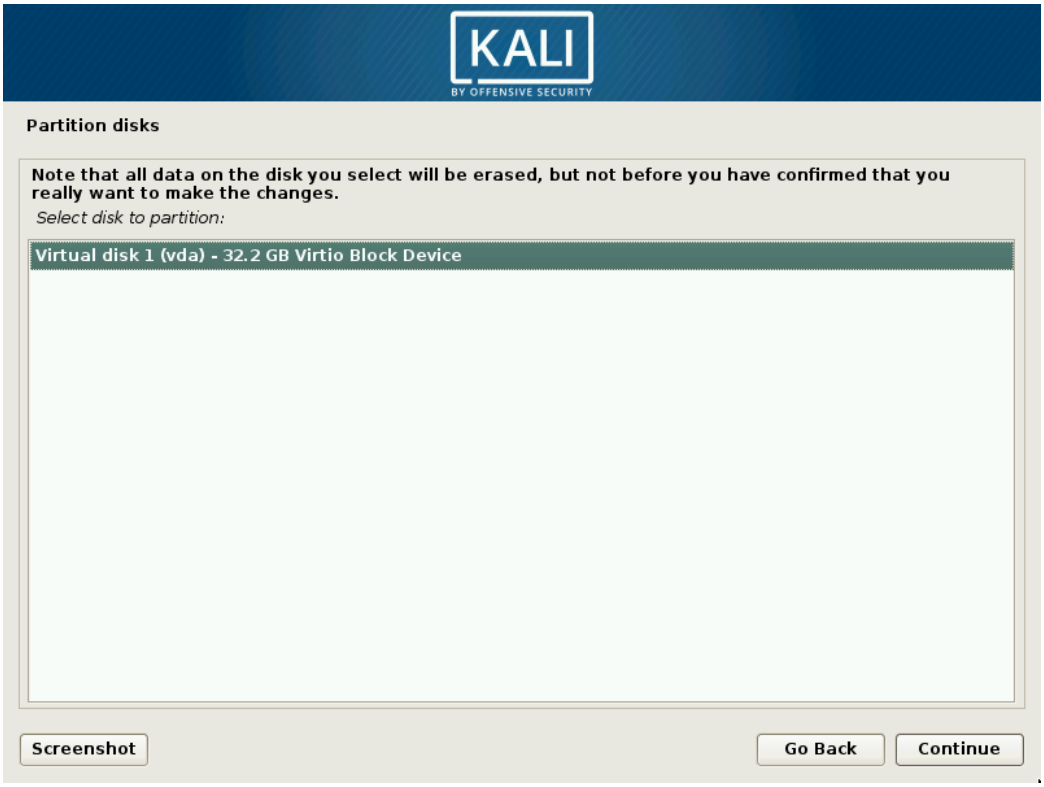


Figure 4.9 *Disk to Use for Guided Partitioning*

Next, the guided partitioning tool offers three partitioning methods, which correspond to different usages, as shown in Figure 4.10, “Guided Partition Allocation” [page 76].

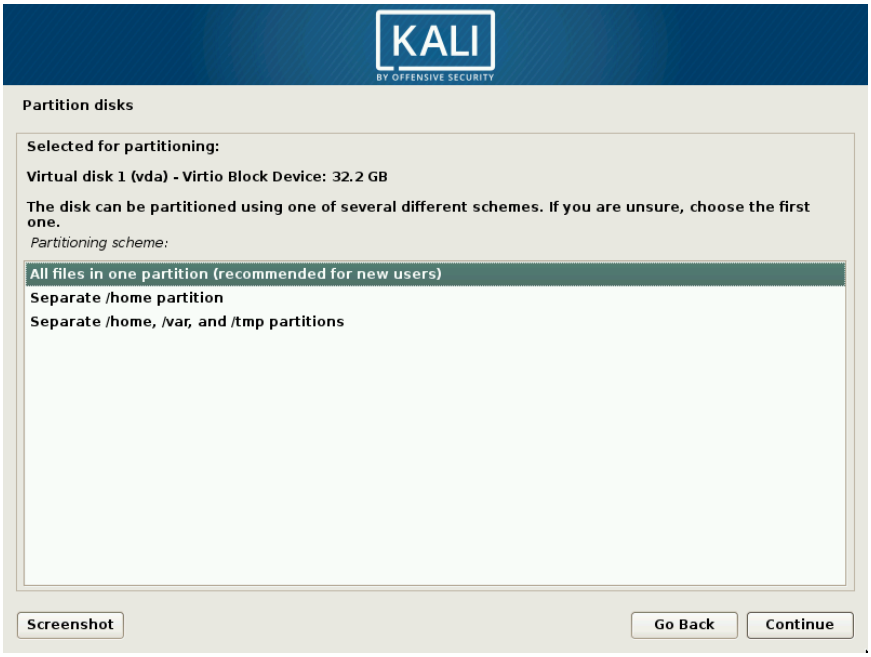


Figure 4.10 *Guided Partition Allocation*

The first method is called “All files in one partition.” The entire Linux system tree is stored in a single file system, corresponding to the root (”/”) directory. This simple and robust partitioning scheme works perfectly well for personal or single-user systems. Despite the name, two partitions will actually be created: the first will house the complete system, the second the virtual memory (or “swap”).

The second method, “Separate `/home/` partition,” is similar, but splits the file hierarchy in two: one partition contains the Linux system (`/`), and the second contains “home directories” (meaning user data, in files and subdirectories available under `/home/`). One benefit to this method is that it is easy to preserve the users’ data if you have to reinstall the system.

The last partitioning method, called “Separate `/home`, `/var`, and `/tmp` partitions,” is appropriate for servers and multi-user systems. It divides the file tree into many partitions: in addition to the root (`/`) and user accounts (`/home/`) partitions, it also has partitions for server software data (`/var/`), and temporary files (`/tmp/`). One benefit to this method is that end users cannot lock up the server by consuming all available hard drive space (they can only fill up `/tmp/` and `/home/`). At the same time, daemon data (especially logs) can no longer clog up the rest of the system.

After choosing the type of partition, the installer presents a summary of your selections on the screen as a partition map (Figure 4.11, "Validating Partitioning" [page 77]). You can modify each partition individually by selecting a partition. For example, you could choose another file system if the standard (*ext4*) isn't appropriate. In most cases, however, the proposed partitioning is reasonable and you can accept it by selecting "Finish partitioning and write changes to disk." It may go without saying, but choose wisely as this will erase the contents of the selected disk.

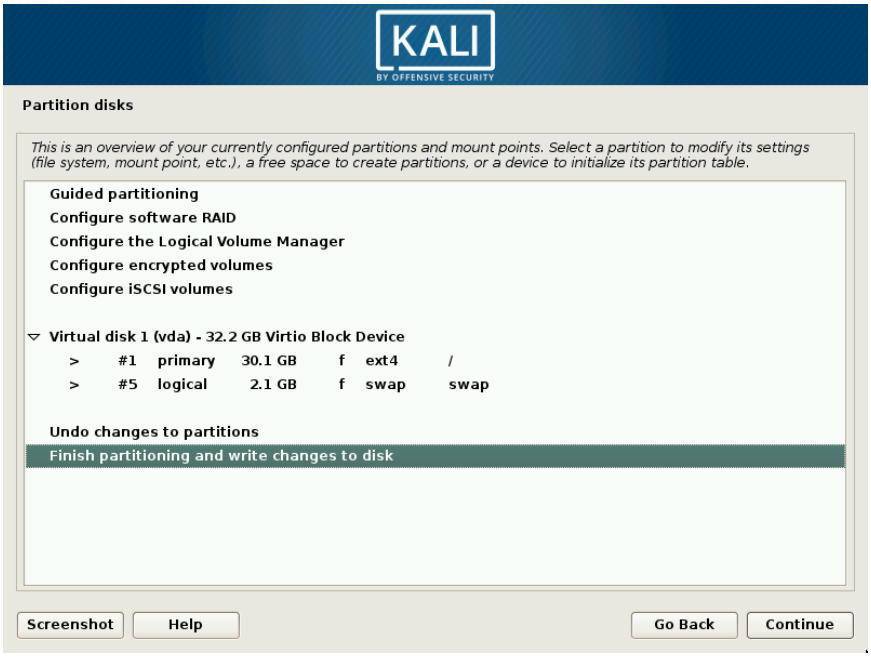


Figure 4.11 *Validating Partitioning*

Manual Partitioning Selecting Manual at the main "Partition disks" screen (Figure 4.8, "Choice of Partitioning Mode" [page 74]) permits greater flexibility, allowing you to choose more advanced configurations and specifically dictate the purpose and size of each partition. For example, this mode allows you to install Kali alongside other operating systems, enable a software-based redundant array of independent disks (RAID) to protect data from hard disk failures, and safely resize existing partitions without losing data, among other things.

Shrinking a Windows Partition

To install Kali Linux alongside an existing operating system (Windows or other), you will need available, unused hard drive space for the partitions dedicated to Kali. In most cases, this means shrinking an existing partition and reusing the freed space.

If you are using the manual partitioning mode, the installer can shrink a Windows partition quite easily. You only need to choose the Windows partition and enter its new size (this works the same with both FAT and NTFS partitions).

If you are a less experienced user working on a system with existing data, please be very careful with this setup method as it is very easy to make mistakes that could lead to data loss.

The first screen in the manual installer is actually the same as the one shown in Figure 4.11, "Validating Partitioning" [page 77], except that it doesn't include any new partitions to create. It is up to you to add those.

First, you will see an option to enter "Guided partitioning" followed by several configuration options. Next, the installer will show the available disks, their partitions, and any possible free space that has not yet been partitioned. You can select each displayed element and press the Enter key to interact with it, as usual.

If the disk is entirely new, you might have to create a partition table. You can do this by selecting the disk. Once done, you should see free space available within the disk.

To make use of this free space, you should select it and the installer will offer you two ways to create partitions in that space.

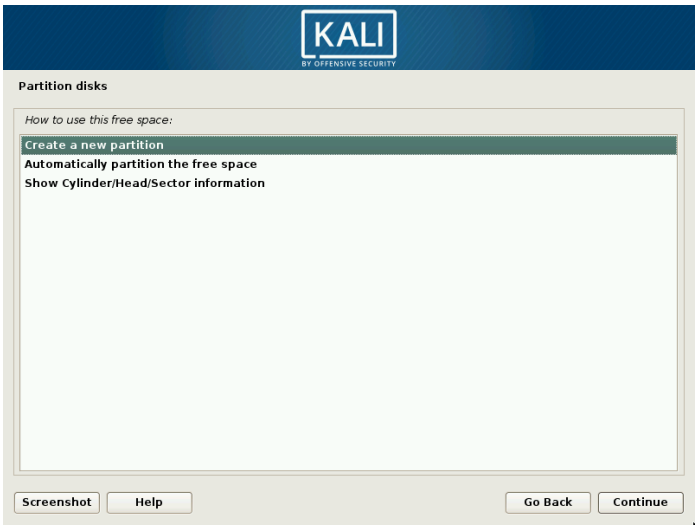


Figure 4.12 *Creating Partitions in the Free Space*

The first entry will create a single partition with the characteristics (including the size) of your choice. The second entry will use all the free space and will create multiple partitions in it with the help of the guided partitioning wizard (see section 4.2.1.12.1, "Guided Partitioning" [page 75]). This option is particularly interesting when you want to install Kali alongside another operating system but when you don't want to micro-manage the partition layout. The last entry will show the cylinder/head/sector numbers of the start and of the end of the free space.

When you select to "Create a new partition," you will enter into the meat of the manual partitioning sequence. After selecting this option, you will be prompted for a partition size. If the disk

uses an MSDOS partition table, you will be given the option to create a primary or logical partition. (Things to know: You can only have four primary partitions but many more logical partitions. The partition containing `/boot`, and thus the kernel, must be a primary one, logical partitions reside in an extended partition, which consumes one of the four primary partitions.) Then you should see the generic partition configuration screen:

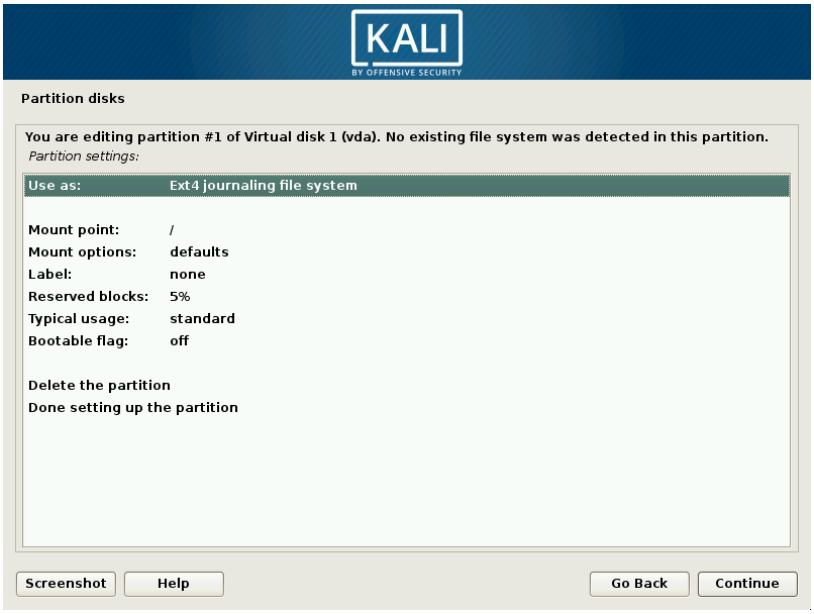


Figure 4.13 *Partition Configuration Screen*

To summarize this step of manual partitioning, let's take a look at what you can do with the new partition. You can:

- Format it and include it in the file tree by choosing a mount point. The mount point is the directory that will house the contents of the file system on the selected partition. Thus, a partition mounted at `/home/` is traditionally intended to contain user data, while "`/`" is known as the *root* of the file tree, and therefore the root of the partition that will actually host the Kali system.

- Use it as a *swap partition*. When the Linux kernel lacks sufficient free memory, it will store inactive parts of RAM in a special swap partition on the hard disk. The virtual memory subsystem makes this transparent to applications. To simulate the additional memory, Windows uses a swap (paging) file that is directly contained in a file system. Conversely, Linux uses a partition dedicated to this purpose, hence the term swap partition.

- Make it into a “physical volume for encryption” to protect the confidentiality of data on certain partitions. This case is automated in the guided partitioning. See section 4.2.2, “Installation on a Fully Encrypted File System” [page 85] for more information.
- Make it a “physical volume for LVM” (not covered in this book). Note that this feature is used by the guided partitioning when you set up encrypted partitions.
- Use it as a RAID device (not covered in this book).
- Choose not to use the partition, and leave it unchanged.

When finished, you can either back out of manual partitioning by selecting “Undo changes to partitions” or write your changes to the disk by selecting “Finish partitioning and write changes to disk” from the manual installer screen (Figure 4.11, “Validating Partitioning” [page 77]).

Copying the Live Image

This next step, which doesn’t require any user interaction, copies the contents of the live image to the target file system, as shown in Figure 4.14, “Copying the Data from the Live Image” [page 80].

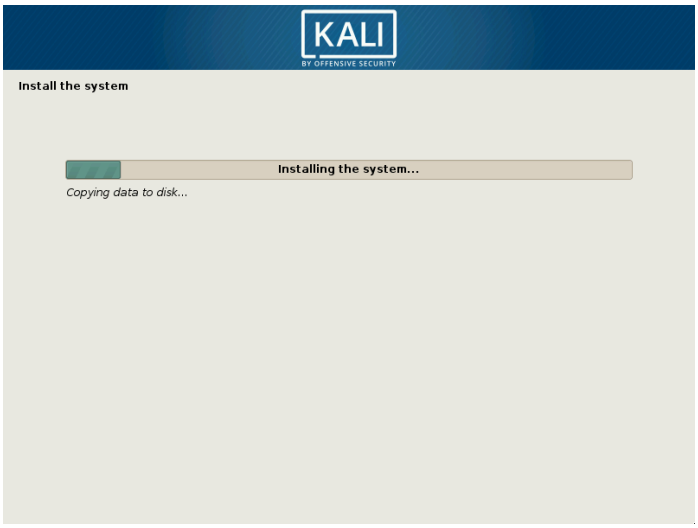


Figure 4.14 *Copying the Data from the Live Image*

Configuring the Package Manager (apt)

In order to be able to install additional software, APT needs to be configured and told where to find Debian packages. In Kali, this step is mostly non-interactive as we force the mirror to be http.kali.org. You just have to confirm whether you want to use this mirror (Figure 4.15, "Use a Network Mirror?" [page 81]). If you don't use it, you won't be able to install supplementary packages with `apt` unless you configure a package repository later.

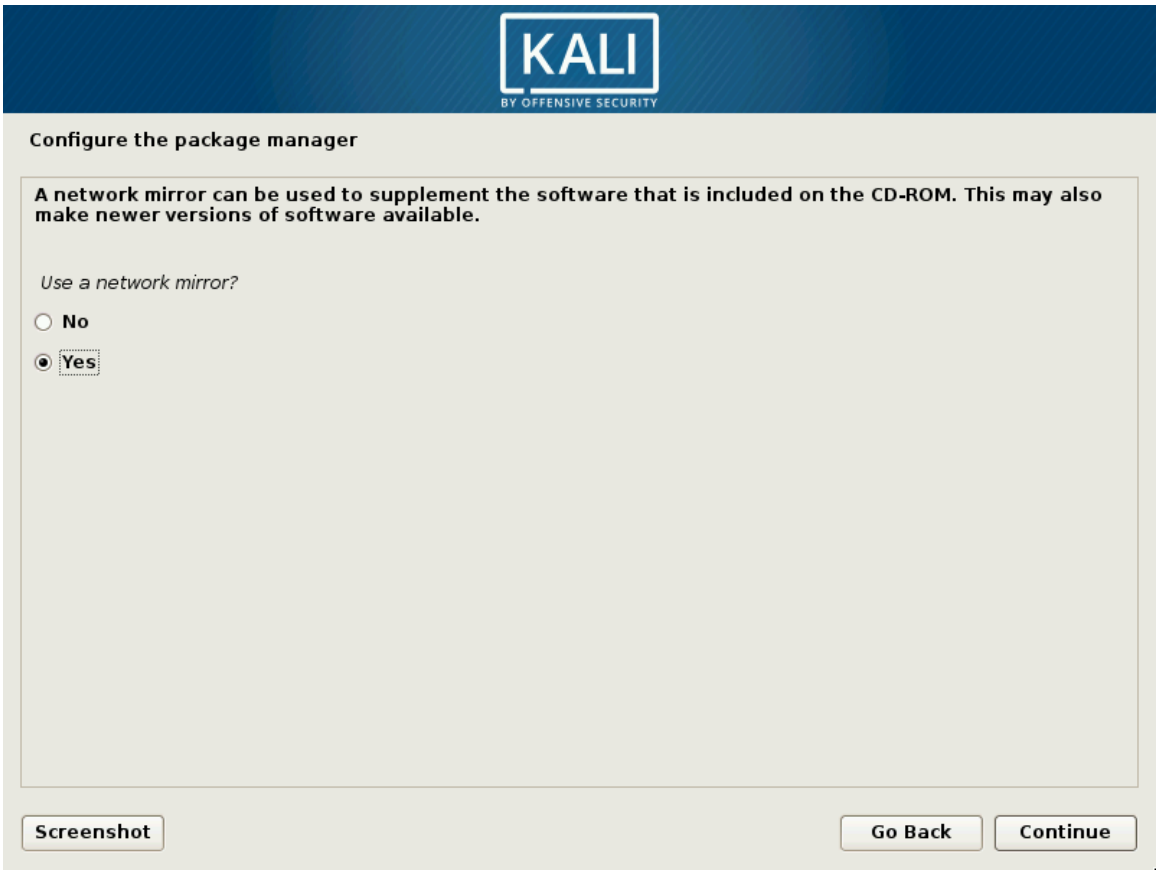


Figure 4.15 *Use a Network Mirror?*

If you want to use a local mirror instead of http.kali.org, you can pass its name on the kernel command line (at boot-time) with a syntax like this: mirror/http/hostname=*my.own.mirror*.

Finally, the program proposes to use an *HTTP proxy* as shown in Figure 4.16, "Use an HTTP Proxy" [page 82]. An HTTP proxy is a server that forwards HTTP requests for network users. It sometimes helps to speed up downloads by keeping a copy of files that have been transferred through it (we then speak of a caching proxy). In some cases, it is the only means of accessing an external web server; in such cases the installer will only be able to download the Debian packages if you properly fill in this field during installation. If you do not provide a proxy address, the installer will attempt to connect directly to the Internet.

Figure 4.16 *Use an HTTP Proxy*

Next, the `Packages.xz` and `Sources.xz` files will be automatically downloaded to update the list of packages recognized by APT.

Installing the GRUB Boot Loader

The boot loader is the first program started by the BIOS. This program loads the Linux kernel into memory and then executes it. The boot loader often offers a menu that allows you to choose the kernel to load or the operating system to boot.

Due to its technical superiority, GRUB is the default boot loader installed by Debian: it works with most file systems and therefore doesn't require an update after each installation of a new kernel, since it reads its configuration during boot and finds the exact position of the new kernel.

You should install GRUB to the Master Boot Record (MBR) unless you already have another Linux system installed that knows how to boot Kali Linux. As noted in Figure 4.17, "Install the GRUB Boot Loader on a Hard Disk" [page 83], modifying the MBR will make unrecognized operating systems that depend on it unbootable until you fix GRUB's configuration.

Figure 4.17 *Install the GRUB Boot Loader on a Hard Disk*

In this step (Figure 4.18, "Device for Boot Loader Installation" [page 84]), you must select which device GRUB will be installed on. This should be your current boot drive.

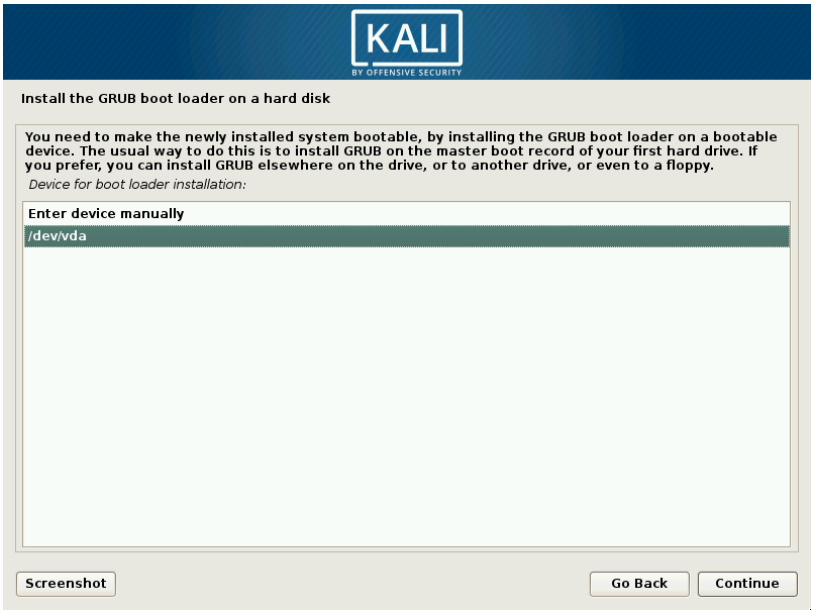


Figure 4.18 *Device for Boot Loader Installation*

By default, the boot menu proposed by GRUB shows all the installed Linux kernels, as well as any other operating systems that were detected. This is why you should accept the offer to install it in the Master Boot Record. Keeping older kernel versions preserves the ability to boot the system if the most recently installed kernel is defective or poorly adapted to the hardware. We thus recommend that you keep a few older kernel versions installed.

Beware: The Boot Loader and Dual Boot

This phase in the installation process detects the operating systems that are already installed on the computer and will automatically add corresponding entries in the boot menu. However, not all installation programs do this.

In particular, if you install (or reinstall) Windows thereafter, the boot loader will be erased. Kali will still be on the hard drive, but will no longer be accessible from the boot menu. You would then have to start the Kali installer with the **`rescue/enable=true`** parameter on the kernel command line to reinstall the boot loader. This operation is described in detail in the Debian installation manual.

➨ `http://www.debian.org/releases/stable/amd64/ch08s07.html`

Finishing the Installation and Rebooting

Now that installation is complete, the program asks you to remove the DVD-ROM from the reader (or unplug your USB drive) so that your computer can boot into your new Kali system after the installer restarts the system (Figure 4.19, "Installation Complete" [page 85]).

Finally, the installer will do some cleanup work, like removing packages that are specific to creating the live environment.

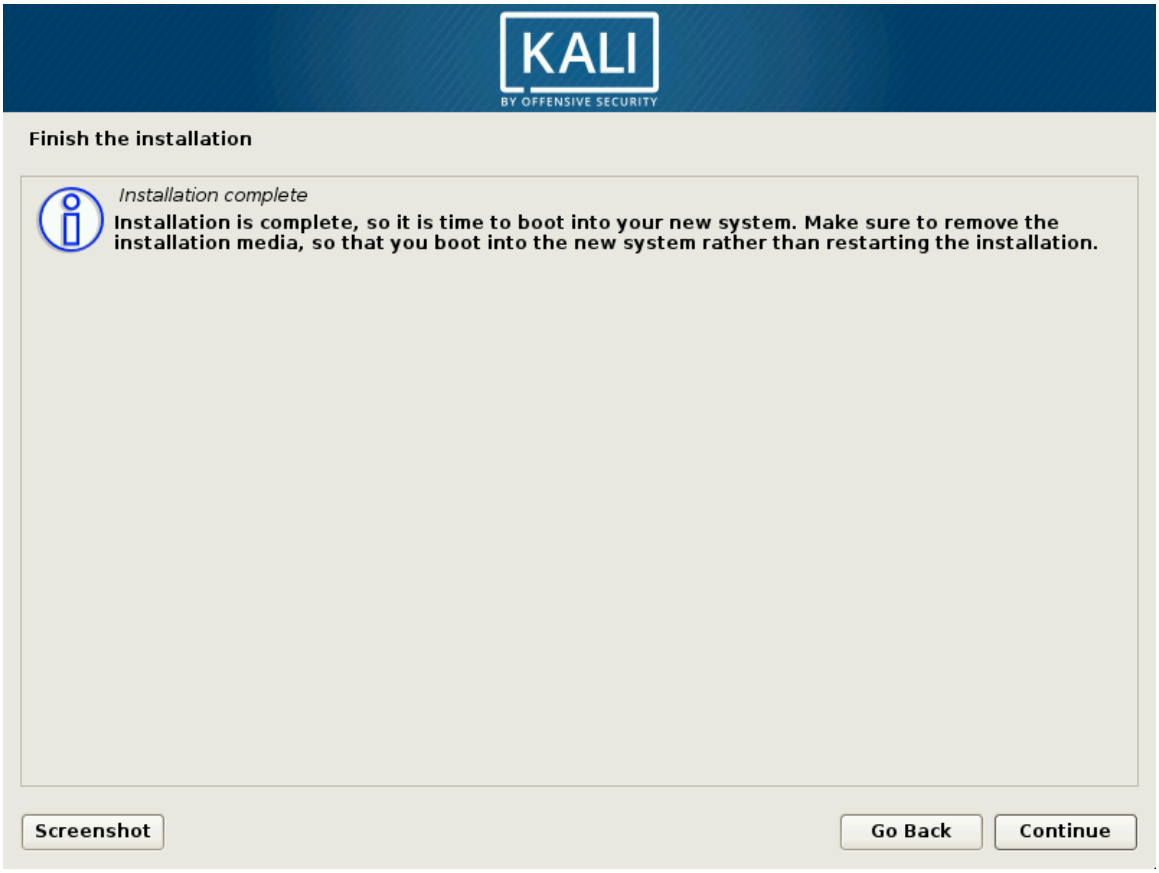


Figure 4.19 *Installation Complete*

4.2.2. Installation on a Fully Encrypted File System

To guarantee the confidentiality of your data, you can set up encrypted partitions. This will protect your data if your laptop or hard drive is lost or stolen. The partitioning tool can help you in this process, both in guided and manual mode.

The guided partitioning mode will combine the use of two technologies: Linux Unified Key Setup (LUKS) for encrypting partitions and Logical Volume Management (LVM) for managing storage dynamically. Both features can also be set up and configured through manual partitioning mode.

Introduction to LVM

Let's discuss LVM first. Using LVM terminology, a *virtual partition* is a logical volume, which is part of a *volume group*, or an association of several physical volumes. Physical volumes are real partitions (or virtual partitions exported by other abstractions, such as a software RAID device or an encrypted partition).

With its lack of distinction between "physical" and "logical" partitions, LVM allows you to create "virtual" partitions that span several disks. The benefits are twofold: the size of the partitions is no longer limited by individual disks but by their cumulative volume, and you can resize existing partitions at any time, such as after adding an additional disk.

This technique works in a very simple way: each volume, whether physical or logical, is split into blocks of the same size, which LVM correlates. The addition of a new disk will cause the creation of a new physical volume providing new blocks that can be associated to any volume group. All of the partitions in the volume group can then take full advantage of the additional allocated space.

Introduction to LUKS

To protect your data, you can add an encryption layer underneath your file system of choice. Linux (and more particularly the *dm-crypt* driver) uses the device mapper to create the virtual partition (whose contents are protected) based on an underlying partition that will store the data in an encrypted form (thanks to LUKS). LUKS standardizes the storage of the encrypted data as well as meta-information that indicates the encryption algorithms used.

Encrypted Swap Partition	When an encrypted partition is used, the encryption key is stored in memory (RAM), and when hibernating, a laptop will copy the key, along with other contents of RAM, to the hard disk's swap partition. Since anyone with access to the swap file (including a technician or a thief) could extract the key and decrypt your data, the swap file must be protected with encryption. Because of this, the installer will warn you if you try to use an encrypted partition alongside an unencrypted swap partition.

Setting Up Encrypted Partitions

The installation process for encrypted LVM is the same as a standard installation except for the partitioning step (Figure 4.20, "Guided Partitioning with Encrypted LVM" [page 87]) where you

will instead select “Guided - use entire disk and set up encrypted LVM.” The net result will be a system that cannot be booted or accessed until the encryption passphrase is provided. This will encrypt and protect the data on your disk.

Figure 4.20 *Guided Partitioning with Encrypted LVM*

The guided partitioning installer will automatically assign a physical partition for the storage of encrypted data, as shown in Figure 4.21, “Confirm Changes to the Partition Table” [page 88]. At this point, the installer will confirm the changes before they are written on the disk.

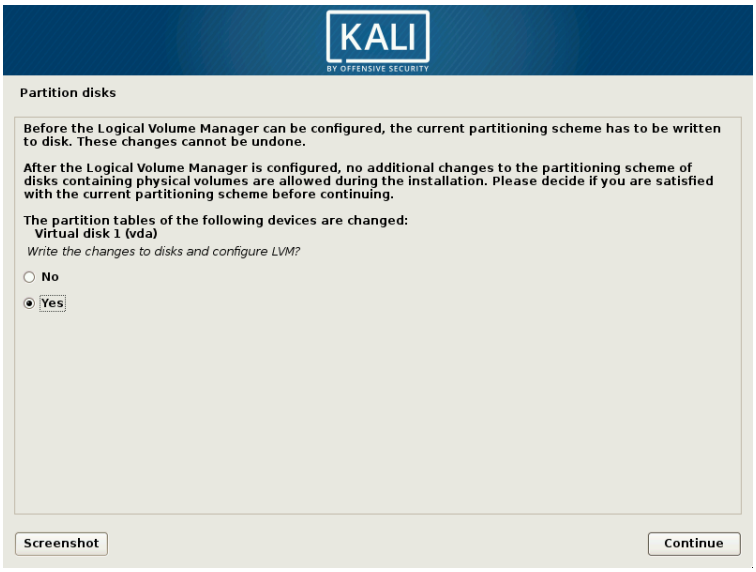


Figure 4.21 *Confirm Changes to the Partition Table*

This new partition is then initialized with random data, as shown in Figure 4.22, "Erasing Data on Encrypted Partition" [page 88]. This makes the areas that contain data indistinguishable from the unused areas, making it more difficult to detect, and subsequently attack, the encrypted data.

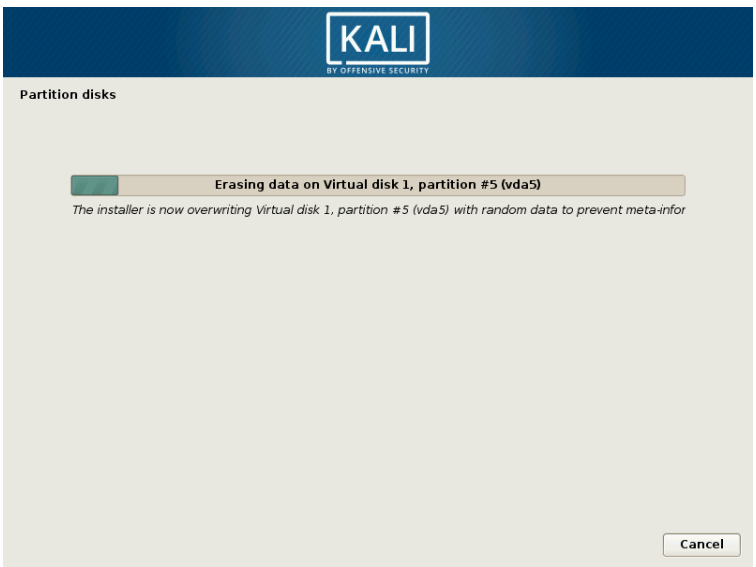


Figure 4.22 *Erasing Data on Encrypted Partition*

Next, the installer asks you to enter an encryption passphrase (Figure 4.23, "Enter Your Encryption Passphrase" [page 89]). In order to view the contents of the encrypted partition, you will need to enter this passphrase every time you reboot the system. Note the warning in the installer: your encrypted system will only be as strong as this passphrase.

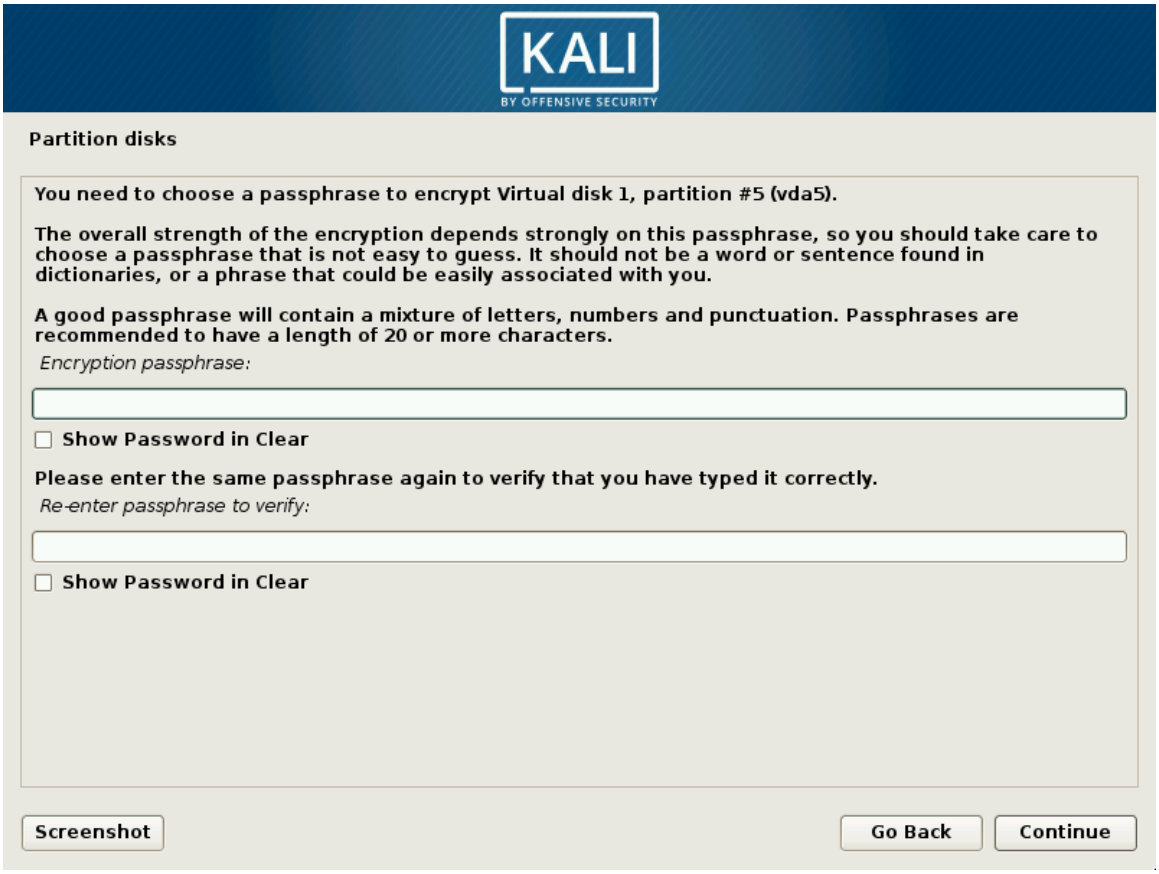


Figure 4.23 *Enter Your Encryption Passphrase*

The partitioning tool now has access to a new virtual partition whose contents are stored encrypted in the underlying physical partition. Since LVM uses this new partition as a physical volume, it can protect several partitions (or LVM logical volumes) with the same encryption key, including the swap partition (see sidebar "Encrypted Swap Partition" [page 86]). Here, LVM is not used to make it easy to extend the storage size, but just for the convenience of the indirection allowing to split a single encrypted partition into multiple logical volumes.

Next, the resulting partitioning scheme is displayed (Figure 4.24, "Validating Partitioning for Encrypted LVM Installation" [page 90]) so you can tweak settings as needed.

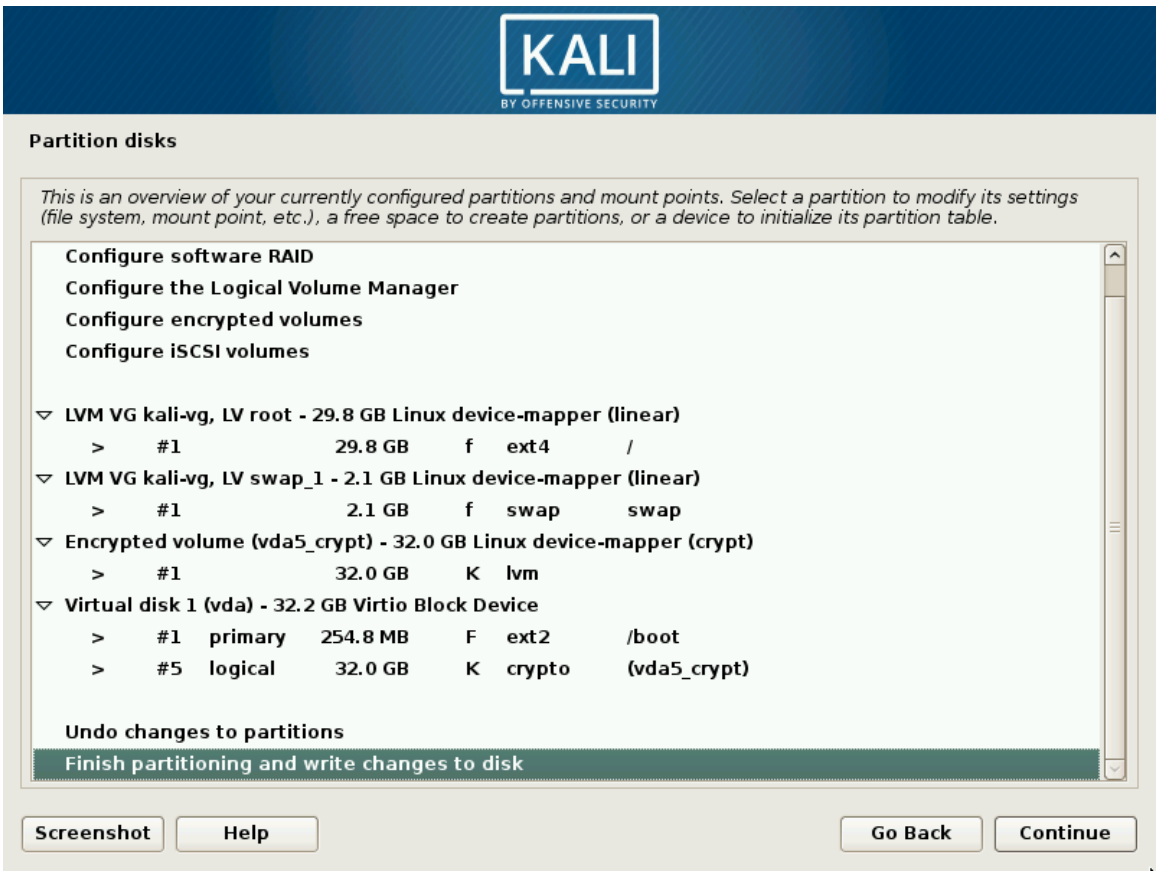


Figure 4.24 *Validating Partitioning for Encrypted LVM Installation*

Finally, after validating the partition setup, the tool asks for confirmation to write the changes on the disks, as shown in Figure 4.25, “Confirm Partitions to be Formatted” [page 91].

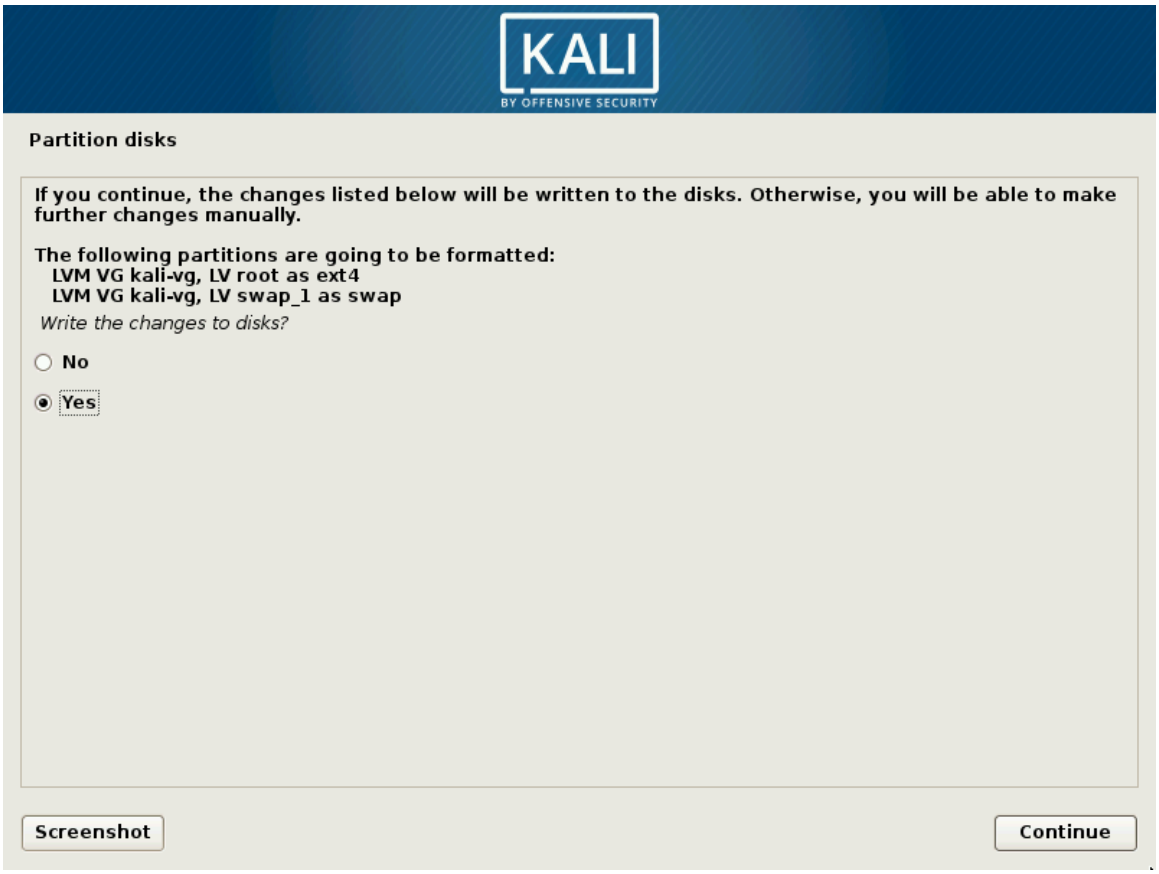


Figure 4.25 *Confirm Partitions to be Formatted*

Finally, the installation process continues as usual as documented in section 4.2.1.14, “Configuring the Package Manager (`apt`)” [page 81].

4.3. Unattended Installations

The Debian and Kali installers are very modular: at the basic level, they are just executing many scripts (packaged in tiny packages called udeb—for µdeb or micro-deb) one after another. Each script relies on `debconf` (see “The `debconf` Tool” [page 214]), which interacts with you, the user, and stores installation parameters. Because of this, the installer can also be automated through

debconf preseeding, a function that allows you to provide unattended answers to installation questions.

4.3.1. Preseeding Answers

There are multiple ways to preseed answers to the installer. Each method has its own advantages and disadvantages. Depending on when the preseeding happens, the questions that can be preseeded vary.

With Boot Parameters

You can preseed any installer question with boot parameters that end up in the kernel command-line, accessible through `/proc/cmdline`. Some bootloaders will let you edit these parameters interactively (which is practical for testing purposes), but if you want to make the changes persistent, you will have to modify the bootloader configuration.

You can directly use the full identifier of the debconf questions (such as `debian-installer/language=en`) or you can use abbreviations for the most common questions (like `language=en` or `hostname=duke`). See the full list[1] of aliases in the Debian installation manual.

There is no restriction on which questions you can preseed since boot parameters are available from the start of the installation process and they are processed very early. However, the number of boot parameters is limited to 32 and a number of those are already used by default. It is also important to realize that changing the boot loader configuration can be non-trivial at times.

In section 9.3, "Building Custom Kali Live ISO Images" [page 236] you will also learn how to modify the Isolinux configuration when you generate your own Kali ISO image.

With a Preseed File in the Initrd

You can add a file named `preseed.cfg` at the root of the installer's `initrd` (this is the initrd which is used to start the installer). Usually, this requires rebuilding the debian-installer source package to generate new versions of the initrd. However, `live-build` offers a convenient way to do this, which is detailed in section 9.3, "Building Custom Kali Live ISO Images" [page 236].

This method also does not have any restrictions on the questions that you can preseed as the preseed file is available immediately after boot. In Kali, we already make use of this feature to customize the behavior of the official Debian installer.

[1] `https://www.debian.org/releases/stable/amd64/apbs02#preseed-aliases`

With a Preseed File in the Boot Media

You can add a preseed file on the boot media (CD or USB key); preseeding then happens as soon as the media is mounted, which means right after the questions about language and keyboard layout. The preseed/file boot parameter can be used to indicate the location of the preseeding file (for instance, `/cdrom/preseed.cfg` when installing from a CD-ROM, or `/hd-media/preseed.cfg` when installing from a USB-key).

You may not preseed answers to language and country options as the preseeding file is loaded later in the process, once the hardware drivers have been loaded. On the positive side, `live-build` makes it easy to put a supplementary file in the generated ISO images (see section 9.3, "Building Custom Kali Live ISO Images" [page 236]).

With a Preseed File Loaded from the Network

You can make a preseed file available on the network through a web server and tell the installer to download that preseed file by adding the boot parameter preseed/url=http://*server*/preseed.cfg (or by using the url alias).

However, when using this method, remember that the network must first be configured. This means that network-related debconf questions (in particular hostname and domain name) and all the preceding questions (like language and country) cannot be preseeded with this method. This method is most often used in combination with boot parameters preseeding those specific questions.

This preseeding method is the most flexible one as you can change the installation configuration without changing the installation media.

Delaying the Language, Country, Keyboard Questions

To overcome the limitation of not being able to preseed the language, country, and keyboard questions, you can add the boot parameter `auto-install/enable=true` (or `auto=true`). With this option the questions will be asked later in the process, after the network has been configured and thus after download of the preseed file.

The downside is that the first steps (notably network configuration) will always happen in English and if there are errors the user will have to work through English screens (with a keyboard configured in QWERTY).

4.3.2. Creating a Preseed File

A preseed file is a plain text file in which each line contains the answer to one Debconf question. A line is split across four fields separated by white space (spaces or tabs). For instance, d-i mirror/suite string kali-rolling:

- The first field indicates the owner of the question. For example, "d-i" is used for questions relevant to the installer. You may also see a package name, for questions coming from Debian packages (as in this example: atftpd atftpd/use_inetd boolean false).
- The second field is an identifier for the question.
- The third field lists the type of question.
- The fourth and final field contains the value for the expected answer. Note that it must be separated from the third field with a single space; additional space characters are considered part of the value.

The simplest way to write a preseed file is to install a system by hand. Then the `debconf-get-selections --installer` command will provide the answers you provided to the installer. You can obtain answers directed to other packages with `debconf-get-selections`. However, a cleaner solution is to write the preseed file by hand, starting from an example and then going through the documentation. With this approach, only questions where the default answer needs to be overridden can be preseeded. Provide the priority=critical boot parameter to instruct Debconf to only ask critical questions, and to use the default answer for others.

Installation Guide Appendix

The Debian installation guide, available online, includes detailed documentation on the use of a preseed file in an appendix. It also includes a detailed and commented sample file, which can serve as a base for local customizations.

➨ `https://www.debian.org/releases/stable/amd64/apb.html`

➨ `https://www.debian.org/releases/stable/example-preseed.txt`

Note however that the above links document the stable version of Debian and that Kali uses the testing version so you may encounter slight differences. You can also consult the installation manual hosted on the Debian-installer project's website. It may be more up-to-date.

➨ `http://d-i.alioth.debian.org/manual/en.amd64/apb.html`

4.4. ARM Installations

Kali Linux runs on a wide variety of ARM-based devices (laptops, embedded computers, and developer boards, for example) but you cannot use the traditional Kali installer on these devices since they often have specific requirements in terms of kernel or boot loader configuration.

To make those devices more accessible to Kali users, Offensive Security developed scripts to build disk images[2] that are ready for use with various ARM devices. They provide those images for download on their website:

➨ `https://www.offensive-security.com/kali-linux-arm-images/`

[2] `https://github.com/offensive-security/kali-arm-build-scripts`

Since these images are available, your task of installing Kali on an ARM device is greatly simplified. Here are the basic steps:

1. Download the image for your ARM device and ensure that the checksum matches the one provided on the website (see section 2.1.3, "Verifying Integrity and Authenticity" [page 16] for explanations on how to do that). Note that the images are usually xz-compressed; make sure to uncompress them with `unxz`.
2. Depending on the storage expansion slot available on your specific ARM device, acquire an SD card, micro SD card, or eMMC module that has a capacity of at least 8 GB.
3. Copy the downloaded image to the storage device with `dd`. This is similar to the process of copying an ISO image onto a USB key (see section 2.1.4, "Copying the Image on a DVD-ROM or USB Key" [page 19]).

   ```
   # dd if=kali-image.img of=/dev/something bs=512k
   ```

4. Plug the SD-card/eMMC into your ARM device.
5. Boot your ARM device and log into it (*user "root", password "toor"*). If you don't have a screen connected, then you will have to figure out the IP address that has been assigned via DHCP and connect to that address over SSH. Some DHCP servers have tools or web interfaces to show the current leases. If you don't have anything like that, use a sniffer to look for DHCP lease traffic.
6. Change the root password and generate new SSH host keys, especially if the device will be permanently running on a public network! The steps are relatively straightforward, see "Generating New SSH Host Keys " [page 111].
7. Enjoy your new ARM device running Kali Linux!

Special Cases and More Detailed Documentation

These instructions are generic and while they work for most devices, there are always exceptions. For example, Chromebooks require *developer mode* and other devices require a special keypress in order to boot from external media.

Since ARM devices are added relatively frequently and their specifications are so dynamic, we won't cover specific installation instructions for various ARM devices here. Instead, refer to the dedicated "Kali on ARM" section of the Kali documentation website for information about each ARM device supported by Offensive Security:

➨ `http://docs.kali.org/category/kali-on-arm`

4.5. Troubleshooting Installations

The installer is quite reliable, but you may encounter bugs or face external problems such as: network problems, bad mirrors, and insufficient disk space. Because of this, it is quite useful to be able to troubleshoot problems that appear in the installation process.

When the installer fails, it will show you a rather unhelpful screen such as the one shown in Figure 4.26, “Installation Step Failed” [page 96].

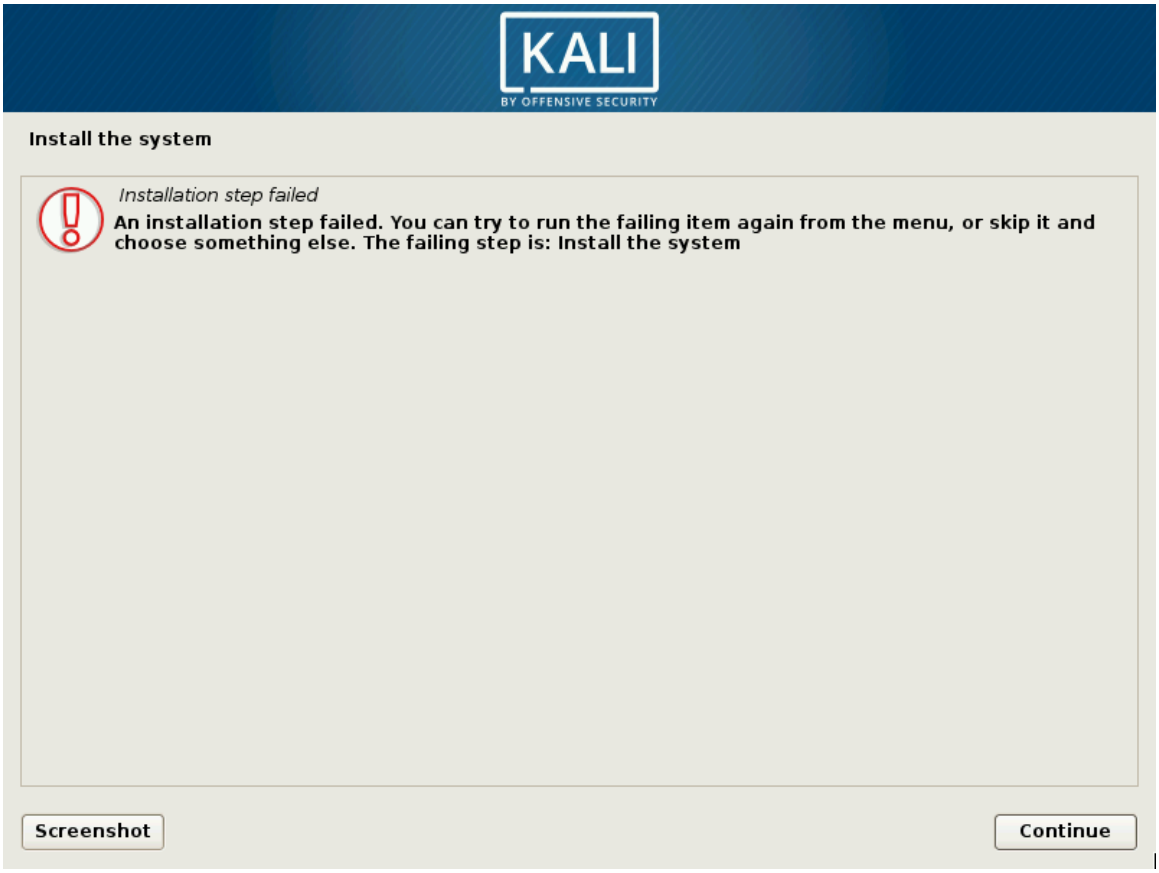


Figure 4.26 *Installation Step Failed*

At this point, it is good to know that the installer makes use of multiple virtual consoles: the main screen that you see is running either on the fifth console (for the graphical installer, CTRL+Shift+F5) or on the first console (for the textual installer, CTRL+Shift+F1). In both cases, the fourth console (CTRL+Shift+F4) displays logs of what is happening and you can usually see a more useful error message there, such as the one in Figure 4.27, “The Log Screen of the Installer” [page 97], which reveals that the installer has run out of disk space.

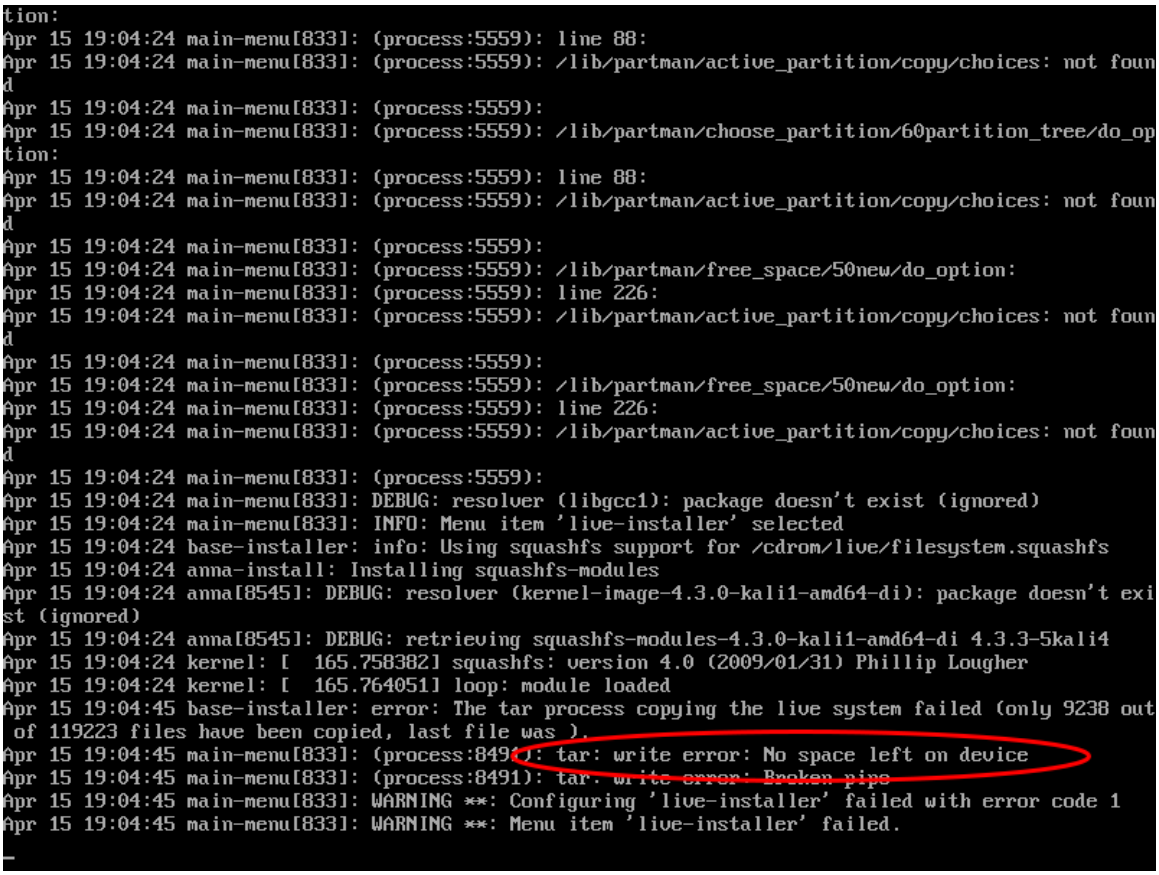


Figure 4.27 *The Log Screen of the Installer*

The second and third consoles (CTRL+Shift+F2 and CTRL+Shift+F3, respectively) host shells that you can use to investigate the current situation in more detail. Most of the command line tools are provided by BusyBox so the feature set is rather limited, but it is enough to figure out most of the problems that you are likely to encounter.

What Can be Done in the Installer Shell

You can inspect and modify the debconf database with `debconf-get` and `debconf-set`. These commands are especially convenient for testing preseeding values.

You can inspect any file (such as the full installation log available in `/var/log/syslog`) with `cat` or `more`. You can edit any file with `nano`, including all files being installed onto the system. The root file system will be mounted on `/target` once the partitioning step of the installation process has completed.

Once network access has been configured, you can use `wget` and `nc` (netcat) to retrieve and export data over the network.

Once you click Continue from the main installer failure screen (Figure 4.26, “Installation Step Failed” [page 96]), you will be returned to a screen that you will normally never see (the Main Menu shown in Figure 4.28, “Main Menu of the Installer” [page 98]), which allows you to launch one installation step after another. If you managed to fix the problem through the shell access (congratulations!) then you can retry the step that failed.

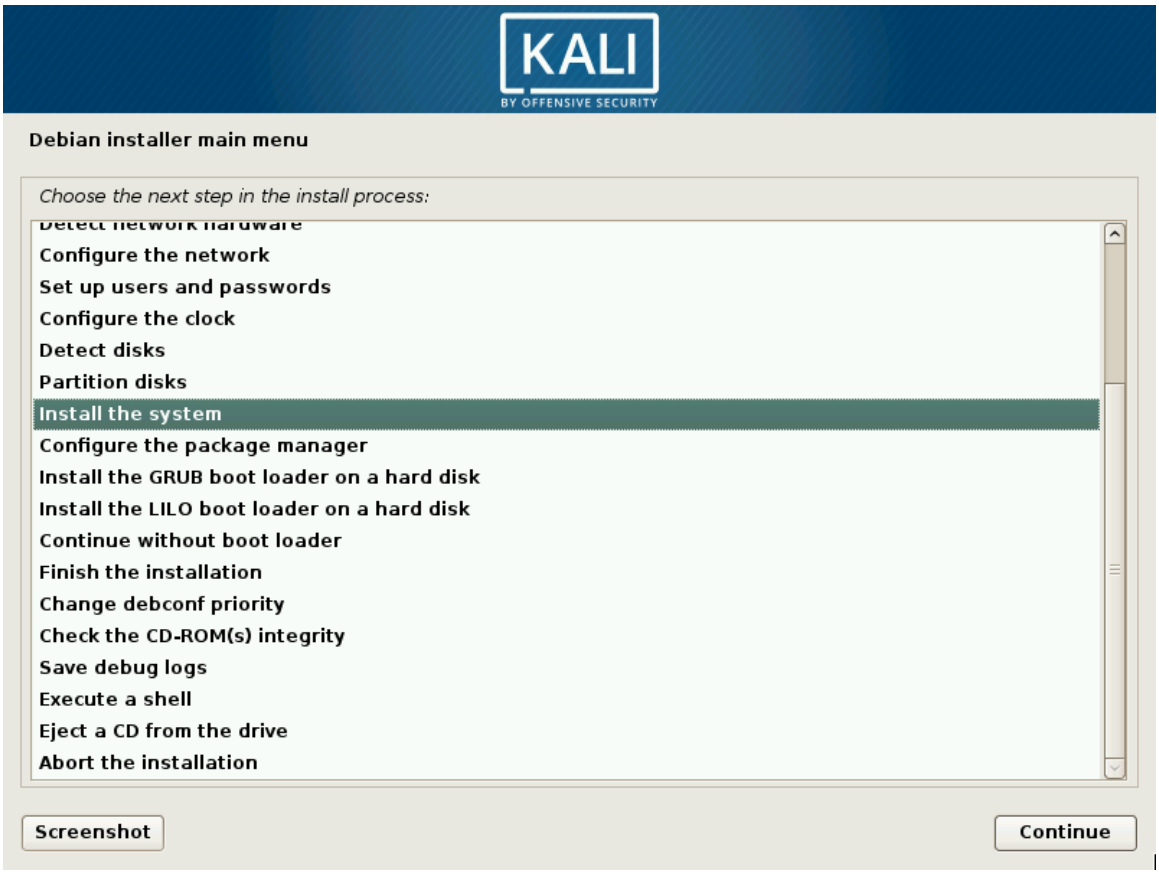


Figure 4.28 *Main Menu of the Installer*

If you are unable to resolve the problem, you might want to file a bug report. The report must then include the installer logs, which you can retrieve with the main menu's "Save debug logs" function. It offers multiple ways to export the logs, as shown in Figure 4.29, "Save Debug Logs (1/2)" [page 99].

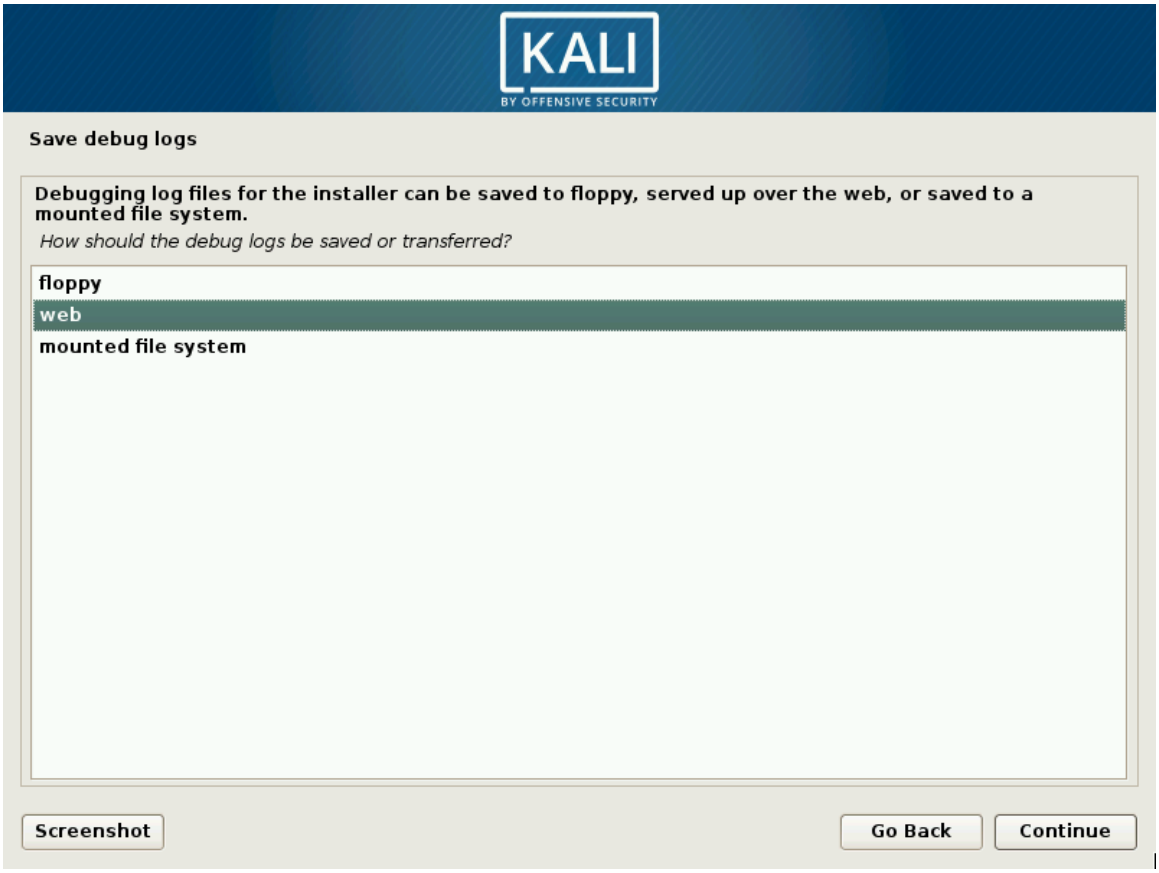


Figure 4.29 *Save Debug Logs (1/2)*

The most convenient method, and the one that we recommend, is to let the installer start a web server hosting the log files (Figure 4.30, "Save Debug Logs (2/2)" [page 100]). You can then launch a browser from another computer on the same network and download all the log files and screenshots that you have taken with the Screenshot button available on each screen.

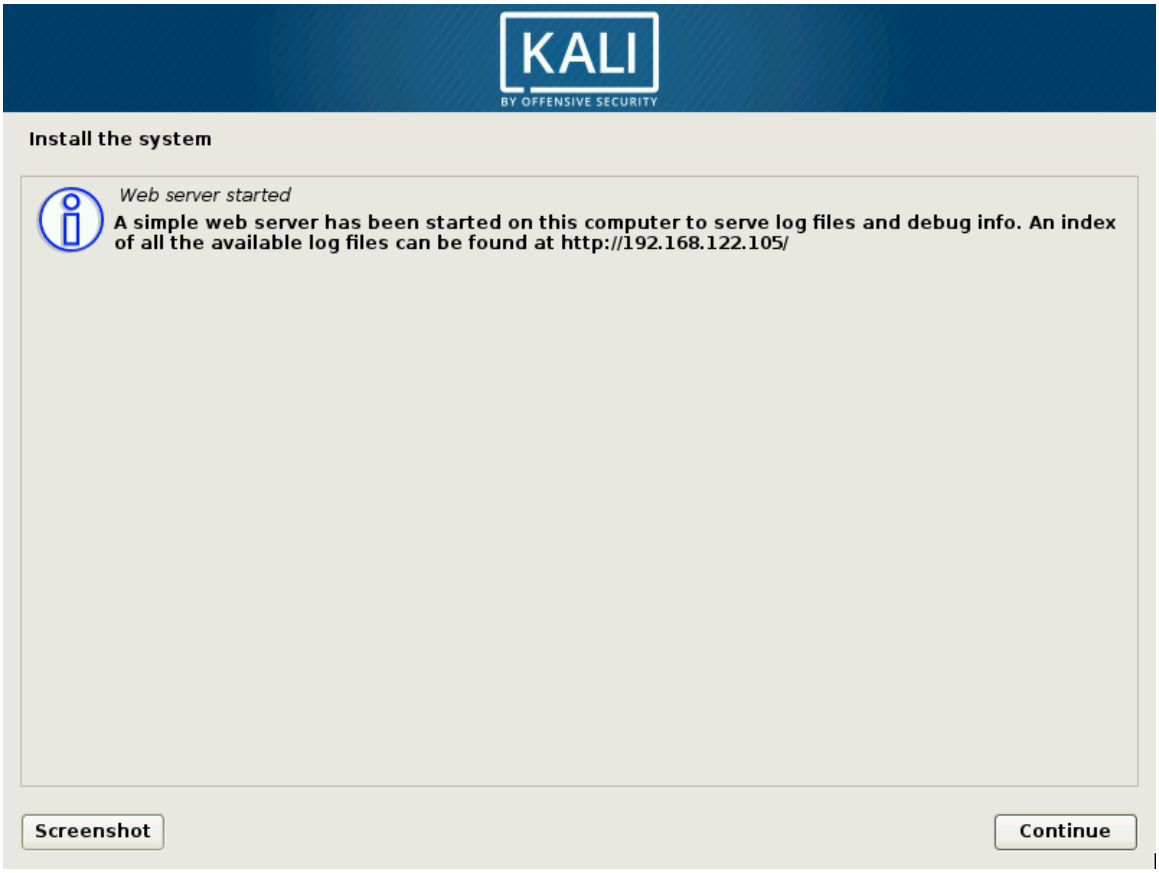


Figure 4.30 *Save Debug Logs (2/2)*

4.6. Summary

In this chapter, we focused on the Kali Linux installation process. We discussed Kali Linux's minimum installation requirements, the installation process for standard and fully encrypted file systems, preseeding, which allows unattended installations, how to install Kali Linux on various ARM devices, and what to do in the rare case of an installation failure.

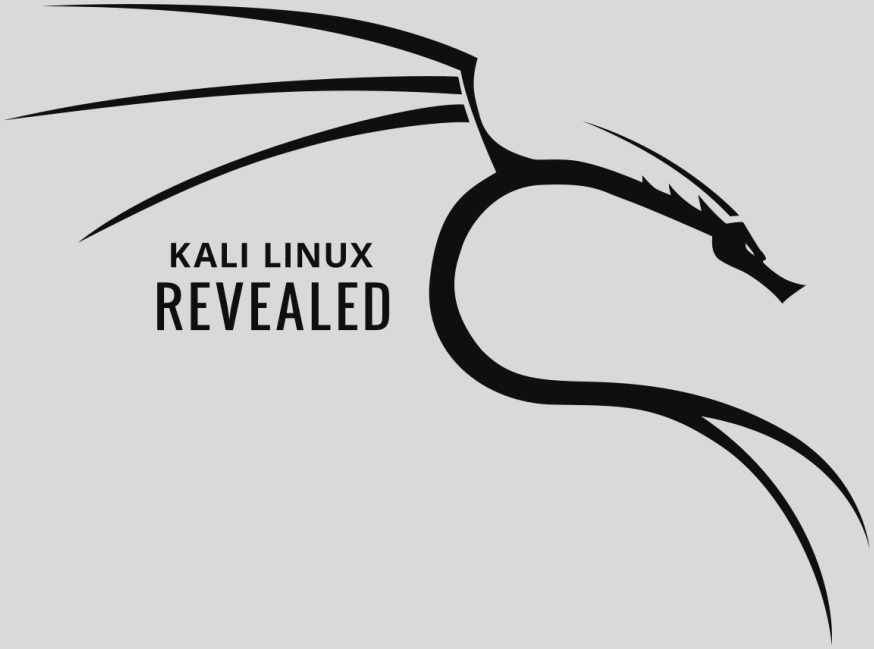
Summary Tips:

- The installation requirements for Kali Linux vary from a basic SSH server with no desktop, as little as 128 MB RAM (512 MB recommended) and 2 GB disk space, to the higher-end *kali-linux-full* meta-package, with at least 2048 MB of RAM and 20 GB of disk space. In addition, your machine must have a CPU supported by at least one of the amd64, i386, armel, armhf, or arm64 architectures.
- Kali can easily be installed as the primary operating system, alongside other operating systems through partitioning and boot loader modification, or as a virtual machine.
- To guarantee the confidentiality of your data, you can set up encrypted partitions. This will protect your data if your laptop or hard drive is lost or stolen.
- The installer can also be automated through debconf preseeding, a function that allows you to provide unattended answers to installation questions.
- A preseed file is a plain text file in which each line contains the answer to one Debconf question. A line is split across four fields separated by white space (spaces or tabs). You can preseed answers to the installer with boot parameters, with a preseed file in initrd, with a preseed file on the boot media, or with a preseed file from the network.
- Kali Linux runs on a wide variety of ARM-based devices such as laptops, embedded computers, and developer boards. ARM installation is fairly straightforward. Download the proper image, burn it to an SD card, USB drive, or embedded multi-media controller (eMMC) module, plug it in, boot the ARM device, find your device on the network, log in, and change the SSH password and SSH host keys.
- You can debug failed installations with virtual consoles (accessible with the CTRL+Shift and function keys), `debconf-get` and `debconf-set` commands, reading the `/var/log/syslog` log file, or by submitting a bug report with log files retrieved with the installer's "Save debug logs" function.

Now that we have discussed Linux fundamentals and Kali Linux installation, let's discuss configuration so you can begin to tailor Kali to suit your needs.

Keywords

Network
Users and groups
Services
Apache
PostgreSQL
SSH



Chapter

5 Configuring Kali Linux

Contents

In this chapter, we will take a look at various ways you can configure Kali Linux. First, in section 5.1, “Configuring the Network” [page 104], we will show you how to configure your network settings using a graphical environment and the command line. In section 5.2, “Managing Unix Users and Unix Groups” [page 107], we will talk about users and groups, showing you how to create and modify user accounts, set passwords, disable accounts, and manage groups. Finally, we will discuss services in section 5.3, “Configuring Services” [page 109] and explain how to set up and maintain generic services and also focus on three very important and specific services: SSH, PostgreSQL, and Apache.

5.1. Configuring the Network

5.1.1. On the Desktop with *NetworkManager*

In a typical desktop installation, you’ll have *NetworkManager* already installed and it can be controlled and configured through GNOME’s control center and through the top-right menu as shown in Figure 5.1, “Network Configuration Screen” [page 104].

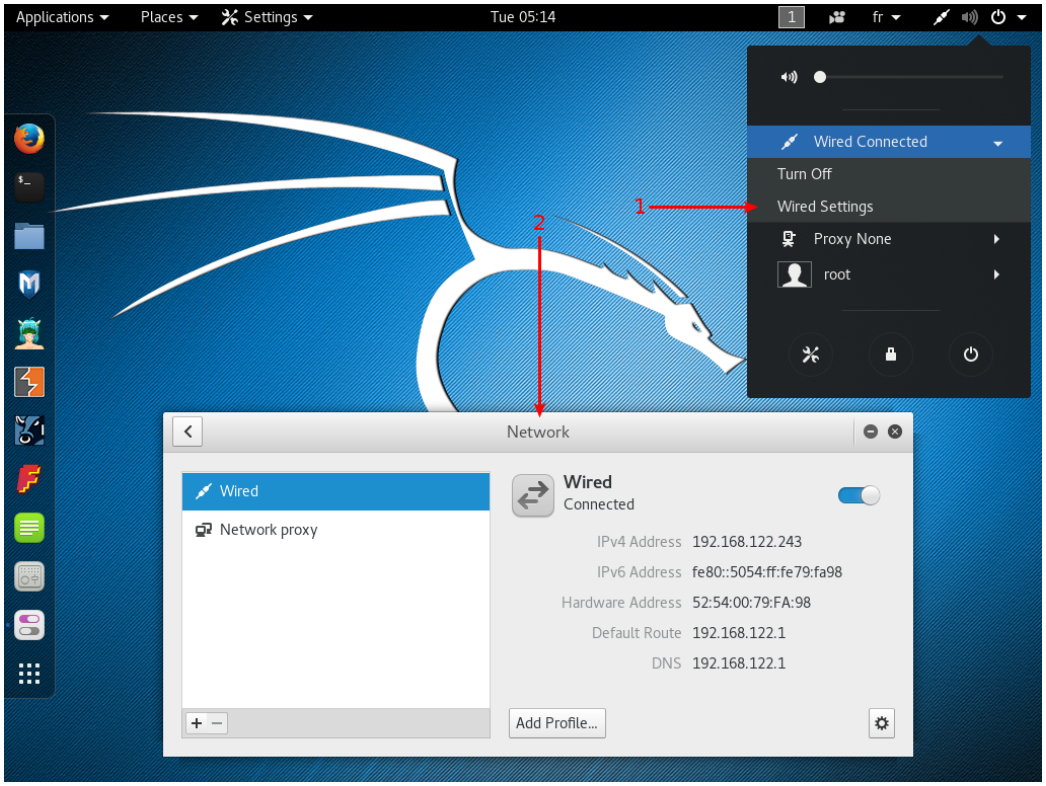


Figure 5.1 *Network Configuration Screen*

The default network configuration relies on DHCP to obtain an IP address, DNS server, and gateway, but you can use the gear icon in the lower-right corner to alter the configuration in many ways (for example: set the MAC address, switch to a static setup, enable or disable IPv6, and add additional routes). You can create profiles to save multiple wired network configurations and easily switch between them. For wireless networks, their settings are automatically tied to their public identifier (SSID).

NetworkManager also handles connections by mobile broadband (Wireless Wide Area Network WWAN) and by modems using point-to-point protocol over ethernet (PPPoE). Last but not least, it provides integration with many types of virtual private networks (VPN) through dedicated plugins: SSH, OpenVPN, Cisco's VPNC, PPTP, Strongswan. Check out the *network-manager-** packages; most of them are not installed by default. Note that you need the packages suffixed with -gnome to be able to configure them through the graphical user interface.

5.1.2. On the Command Line with Ifupdown

Alternatively, when you prefer not to use (or don't have access to) a graphical desktop, you can configure the network with the already-installed *ifupdown* package, which includes the `ifup` and `ifdown` tools. These tools read definitions from the `/etc/network/interfaces` configuration file and are at the heart of the `/etc/init.d/networking` init script that configures the network at boot time.

Each network device managed by *ifupdown* can be deconfigured at any time with `ifdown` *`network-device`*. You can then modify `/etc/network/interfaces` and bring the network back up (with the new configuration) with `ifup` *`network-device`*.

Let's take a look at what we can put in ifupdown's configuration file. There are two main directives: auto *network-device*, which tells *ifupdown* to automatically configure the network interface once it is available, and iface *network-device inet/inet6 type* to configure a given interface. For example, a plain DHCP configuration looks like this:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Note that the special configuration for the loopback device should always be present in this file. For a fixed IP address configuration, you have to provide more details such as the IP address, the network, and the IP of the gateway:

```
auto eth0
iface eth0 inet static
  address 192.168.0.3
  netmask 255.255.255.0
  broadcast 192.168.0.255
  network 192.168.0.0
  gateway 192.168.0.1
```

For wireless interfaces, you must have the *wpasupplicant* package (included in Kali by default), which provides many wpa-* options that can be used in `/etc/network/interfaces`. Have a look at `/usr/share/doc/wpasupplicant/README.Debian.gz` for examples and explanations. The most common options are wpa-ssid (which defines the name of the wireless network to join) and wpa-psk (which defines the passphrase or the key protecting the network).

```
iface wlan0 inet dhcp
  wpa-ssid MyNetWork
  wpa-psk plaintextsecret
```

5.1.3. On the Command Line with *systemd-networkd*

While *ifupdown* is the historical tool used by Debian, and while it is still the default for server or other minimal installations, there is a newer tool worth considering: *systemd-networkd*. Its integration with the *systemd* init system makes it a very attractive choice. It is not specific to Debian-based distributions (contrary to *ifupdown*) and has been designed to be very small, efficient, and relatively easy to configure if you understand the syntax of systemd unit files. This is an especially attractive choice if you consider *NetworkManager* bloated and hard to configure.

You configure `systemd-networkd` by placing `.network` files into the `/etc/systemd/network/` directory. Alternatively, you can use `/lib/systemd/network/` for packaged files or `/run/systemd/network/` for files generated at run-time. The format of those files is documented in `systemd.network(5)`. The Match section indicates the network interfaces the configuration applies to. You can specify the interface in many ways, including by media access control (MAC) address or device type. The Network section defines the network configuration.

Example 5.1 *DHCP-based Configuration in `/etc/systemd/network/80-dhcp.network`*

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Example 5.2 *Static Configuration in `/etc/systemd/network/50-static.network`*

```
[Match]
Name=enp2s0

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
DNS=8.8.8.8
```

Note that `system-networkd` is disabled by default, so if you want to use it, you should enable it. It also depends on `systemd-resolved` for proper integration of DNS resolution, which in turn requires you to replace `/etc/resolv.conf` with a symlink to `/run/system/resolve/resolv.conf`, which is managed by `systemd-resolved`.

```
# systemctl enable systemd-networkd
# systemctl enable systemd-resolved
# systemctl start systemd-networkd
# systemctl start systemd-resolved
# ln -sf /run/system/resolve/resolv.conf /etc/resolv.conf
```

Although `systemd-networkd` suffers from some limitations, like the lack of integrated support for wireless networks, you can rely on a pre-existing external `wpa_supplicant` configuration for wireless support. However, it is particularly useful in containers and virtual machines and was originally developed for environments in which a container's network configuration depended on its host's network configuration. In this scenario, `systemd-networkd` makes it easier to manage both sides in a consistent manner while still supporting all sorts of virtual network devices that you might need in this type of scenario (see `systemd.netdev(5)`).

5.2. Managing Unix Users and Unix Groups

The database of Unix users and groups consists of the textual files `/etc/passwd` (list of users), `/etc/shadow` (encrypted passwords of users), `/etc/group` (list of groups), and `/etc/gshadow` (encrypted passwords of groups). Their formats are documented in `passwd(5)`, `shadow(5)`, `group(5)`, and `gshadow(5)` respectively. While these files can be manually edited with tools like `vipw` and `vigr`, there are higher level tools to perform the most common operations.

5.2.1. Creating User Accounts

Although Kali is most often run while authenticated as the root user, you may often need to create non-privileged user accounts for various reasons, particularly if you are using Kali as a primary

operating system. The most typical way to add a user is with the `adduser` command, which takes a required argument: the username for the new user that you would like to create.

The `adduser` command asks a few questions before creating the account but its usage is fairly straightforward. Its configuration file, `/etc/adduser.conf`, includes many interesting settings. You can, for example, define the range of user identifiers (UIDs) that can be used, dictate whether or not users share a common group or not, define default shells, and more.

The creation of an account triggers the population of the user's home directory with the contents of the `/etc/skel/` template. This provides the user with a set of standard directories and configuration files.

In some cases, it will be useful to add a user to a group (other than their default main group) in order to grant additional permissions. For example, a user who is included in the *sudo* group has full administrative privileges through the `sudo` command. This can be achieved with a command such as `adduser` *`user group`*.

Using `getent` to Consult the User Database

The `getent` (get entries) command checks the system databases (including those of users and groups) using the appropriate library functions, which in turn call the name service switch (NSS) modules configured in the `/etc/nsswitch.conf` file. The command takes one or two arguments: the name of the database to check, and a possible search key. Thus, the command `getent passwd kaliuser1` will return the information from the user database regarding the user `kaliuser1`.

```
root@kali:~# getent passwd kaliuser1
kaliuser1:x:1001:1001:Kali User
    ➥ ,4444,123-867-5309,321-867-5309:/home/kaliuser1:/bin/
    ➥ bash
```

5.2.2. Modifying an Existing Account or Password

The following commands allow modification of the information stored in specific fields of the user databases:

- `passwd`—permits a regular user to change their password, which in turn, updates the `/etc/shadow` file.
- `chfn`—(CHange Full Name), reserved for the super-user (root), modifies the GECOS, or "general information" field.
- `chsh`—(CHange SHell) changes the user's login shell. However, available choices will be limited to those listed in `/etc/shells`; the administrator, on the other hand, is not bound by this restriction and can set the shell to any program chosen.
- `chage`—(CHange AGE) allows the administrator to change the password expiration settings by passing the user name as an argument or list current settings using the -l *user* option.

Alternatively, you can also force the expiration of a password using the `passwd -e user` command, which forces the user to change their password the next time they log in.

5.2.3. Disabling an Account

You may find yourself needing to disable an account (lock out a user) as a disciplinary measure, for the purposes of an investigation, or simply in the event of a prolonged or definitive absence of a user. A disabled account means the user cannot login or gain access to the machine. The account remains intact on the machine and no files or data are deleted; it is simply inaccessible. This is accomplished by using the command `passwd -l user` (lock). Re-enabling the account is done in similar fashion, with the -u option (unlock).

5.2.4. Managing Unix Groups

The `addgroup` and `delgroup` commands add or delete a group, respectively. The `groupmod` command modifies a group's information (its gid or identifier). The command `gpasswd`*group* changes the password for the group, while the `gpasswd -r group` command deletes it.

Working with Several Groups

Each user may be a member of many groups. A user's main group is, by default, created during initial user configuration. By default, each file that a user creates belongs to the user, as well as to the user's main group. This is not always desirable; for example, when the user needs to work in a directory shared by a group other than their main group. In this case, the user needs to change groups using one of the following commands: `newgrp`, which starts a new shell, or `sg`, which simply executes a command using the supplied alternate group. These commands also allow the user to join a group to which they do not currently belong. If the group is password protected, they will need to supply the appropriate password before the command is executed.

Alternatively, the user can set the `setgid` bit on the directory, which causes files created in that directory to automatically belong to the correct group. For more details, see sidebar "`setgid` directory and *sticky bit*" [page 58].

The `id` command displays the current state of a user, with their personal identifier (`uid` variable), current main group (`gid` variable), and the list of groups to which they belong (`groups` variable).

5.3. Configuring Services

In this section we will take a look at services (sometimes called daemons), or programs that run as a background process and perform various functions for the system. We will start by discussing configuration files and will proceed to explain how some important services (such as SSH, PostgreSQL, and Apache) function and how they can be configured.

5.3.1. Configuring a Specific Program

When you want to configure an unknown package, you must proceed in stages. First, you should read what the package maintainer has documented. The `/usr/share/doc/`*`package`*`/README.Debian` file is a good place to start. This file will often contain information about the package, including pointers that may refer you to other documentation. You will often save yourself a lot of time, and avoid a lot of frustration, by reading this file first since it often details the most common errors and solutions to most common problems.

Next, you should look at the software's official documentation. Refer to section 6.1, "Documentation Sources" [page 124] for tips on how to find various documentation sources. The `dpkg -L` *`package`* command gives a list of files included in the package; you can therefore quickly identify the available documentation (as well as the configuration files, located in `/etc/`). Also, `dpkg -s` *`package`* displays the package meta-data and shows any possible recommended or suggested packages; in there, you can find documentation or perhaps a utility that will ease the configuration of the software.

Finally, the configuration files are often self-documented by many explanatory comments detailing the various possible values for each configuration setting. In some cases, you can get software up and running by uncommenting a single line in the configuration file. In other cases, examples of configuration files are provided in the `/usr/share/doc/`*`package`*`/examples/` directory. They may serve as a basis for your own configuration file.

5.3.2. Configuring SSH for Remote Logins

SSH allows you to remotely log into a machine, transfer files, or execute commands. It is an industry standard tool (`ssh`) and service (`sshd`) for connecting to machines remotely.

While the *openssh-server* package is installed by default, the *SSH* service is disabled by default and thus is not started at boot time. You can manually start the SSH service with `systemctl start ssh` or configure it to start at boot time with `systemctl enable ssh`.

The SSH service has a relatively sane default configuration, but given its powerful capabilities and sensitive nature, it is good to know what you can do with its configuration file, `/etc/ssh/sshd_config`. All the options are documented in `sshd_config(5)`.

The default configuration disables password-based logins for the root user, which means you must first set up SSH keys with `ssh-keygen`. You can extend this to all users by setting PasswordAuthentication to no, or you can lift this restriction by changing PermitRootLogin to yes (instead of the default prohibit-password). The SSH service listens by default on port 22 but you can change this with the Port directive.

To apply the new settings, you should run `systemctl reload ssh`.

Generating New SSH Host Keys

Each SSH server has its own cryptographic keys; they are named "SSH host keys" and are stored in `/etc/ssh/ssh_host_*`. They must be kept private if you want confidentiality and they should not be shared by multiple machines.

When you install your system by copying a full disk image (instead of using debian-installer), the image might contain pre-generated SSH host keys that you should thus replace with newly-generated keys. The image probably also comes with a default root password that you want to reset at the same time. You can do all this with the following commands:

```
# passwd
[...]
# rm /etc/ssh/ssh_host_*
# dpkg-reconfigure openssh-server
# service ssh restart
```

5.3.3. Configuring PostgreSQL Databases

PostgreSQL is a database server. It is rarely useful on its own but is used by many other services to store data. Those services will generally access the database server over the network and usually require authentication credentials to be able to connect. Setting up those services thus requires creating PostgreSQL databases and user accounts with appropriate privileges on the database. To be able to do that, we need the service to be running, so let's start it with `systemctl start postgresql`.

Multiple PostgreSQL versions supported

The PostgreSQL packaging allows for multiple versions of the database server to be co-installed. It is also possible to handle multiple *clusters* (a cluster is a collection of databases served by the same `postmaster`). To achieve this, the configuration files are stored in `/etc/postgresql/`*`version/cluster-name/`*.

In order for clusters to run side-by-side, each new cluster gets assigned the next available port number (usually 5433 for the second cluster). The `postgresql.service` file is an empty shell, making it easy to act on all clusters together as each cluster has its own unit (`postgresql@`*`version-cluster`*`.service`).

Connection Type and Client Authentication

By default, PostgreSQL listens for incoming connections in two ways: on TCP port 5432 of the localhost interface and on file-based socket `/var/run/postgresql/.s.PGSQL.5432`. This can be configured in `postgresql.conf` with various directives: listen_addresses for the addresses to listen to, port for the TCP port, and unix_socket_directories to define the directory where the file-based sockets are created.

Depending on how they connect, clients are authenticated in different ways. The `pg_hba.conf` configuration file defines who is allowed to connect on each socket and how they are authenticated. By default, connections on the file-based socket use the Unix user account as the name of the PosgreSQL user, and it assumes that no further authentication is required. On the TCP connection, PostgreSQL requires the user to authenticate with a username and a password (though not a Unix username/password but rather one managed by PostgreSQL itself).

The postgres user is special and has full administrative privileges over all databases. We will use this identity to create new users and new databases.

Creating Users and Databases

The `createuser` command adds a new user and `dropuser` removes one. Likewise, the `createdb` command adds a new database and `dropdb` removes one. Each of these commands have their own manual pages but we will discuss some of the options here. Each command acts on the default cluster (running on port 5432) but you can pass --port=*port* to modify users and databases of an alternate cluster.

These commands must connect to the PostgreSQL server to do their job and they must be authenticated as a user with sufficient privileges to be able to execute the specified operation. The easiest way to achieve this is to use the postgres Unix account and connect over the file-based socket:

```
# su - postgres
$ createuser -P king_phisher
Enter password for new role:
Enter it again:
$ createdb -T template0 -E UTF-8 -O king_phisher king_phisher
$ exit
```

In the example above, the -P option asks `createuser` to query for a password once it creates the new king_phisher user. Looking at the `createdb` command, the -O defines the user owning the new database (which will thus have full rights to create tables and grant permissions and so on). We also want to be able to use Unicode strings, so we add the -E UTF-8 option to set the encoding, which in turn requires us to use the -T option to pick another database template.

We can now test that we can connect to the database over the socket listening on localhost (-h localhost) as the king_phisher user (-U king_phisher):

```
# psql -h localhost -U king_phisher king_phisher
Password for user king_phisher:
psql (9.5.2)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
    ➥ compression: off)
Type "help" for help.

king_phisher=>
```

As you can see, the connection was successful.

Managing PostgreSQL Clusters

First, it is worth noting that the concept of "PostgreSQL cluster" is a Debian-specific addition and that you will not find any reference to this term in the official PostgreSQL documentation. From the point of view of the PostgreSQL tools, such a cluster is just an instance of a database server running on a specific port.

That said, Debian's *postgresql-common* package provides multiple tools to manage such clusters: `pg_createcluster`, `pg_dropcluster`, `pg_ctlcluster`, `pg_upgradecluster`, `pg_renamecluster`, and `pg_lsclusters`. We won't cover all those tools here, but you can refer to their respective manual pages for more information.

What you must know is that when a new major version of PostgreSQL gets installed on your system, it will create a new cluster that will run on the next port (usually 5433) and you will keep using the old version until you migrate your databases from the old cluster to the new one.

You can retrieve a list of all the clusters and their status with `pg_lsclusters`. More importantly, you can automate the migration of your cluster to the latest PostgreSQL version with `pg_upgradecluster` *`old-version cluster-name`*. For this to succeed, you might have to first remove the (empty) cluster created for the new version (with `pg_dropcluster` *`new-version cluster-name`*). The old cluster is not dropped in the process, but it also won't be started automatically. You can drop it once you have checked that the upgraded cluster works fine.

5.3.4. Configuring Apache

A typical Kali Linux installation includes the Apache web server, provided by the *apache2* package. Being a network service, it is disabled by default. You can manually start it with `systemctl start apache2`.

With more and more applications being distributed as web applications, it is important to have some knowledge of Apache in order to host those applications, whether for local usage or for making them available over the network.

Apache is a modular server and many features are implemented by external modules that the main program loads during its initialization. The default configuration only enables the most common modules, but enabling new modules is easily done by running `a2enmod` *`module`*. Use `a2dismod` *`module`* to disable a module. These programs actually only create (or delete) symbolic links in `/etc/apache2/mods-enabled/`, pointing at the actual files (stored in `/etc/apache2/mods-available/`).

There are many modules available, but two are worth initial consideration: PHP and SSL. Web applications written with PHP are executed by the Apache web server with the help of the dedicated

module provided by the *libapache-mod-php* package, and its installation automatically enables the module.

Apache 2.4 includes the SSL module required for secure HTTP (HTTPS) out of the box. It first needs to be enabled with `a2enmod ssl`, then the required directives must be added to the configuration files. A configuration example is provided in `/etc/apache2/sites-available/default-ssl.conf`. See `http://httpd.apache.org/docs/2.4/mod/mod_ssl.html` for more information.

The full list of standard Apache modules can be found online at `http://httpd.apache.org/docs/2.4/mod/index.html`.

With its default configuration, the web server listens on port 80 (as configured in `/etc/apache2/ports.conf`), and serves pages from the `/var/www/html/` directory by default (as configured in `/etc/apache2/sites-enabled/000-default.conf`).

Configuring Virtual Hosts

A virtual host is an extra identity for the web server. The same Apache process can serve multiple websites (say www.kali.org and www.offensive-security.com) because the HTTP requests embed both the name of the website requested and the URL localpart (this feature is known as *name-based virtual hosts*).

The default configuration for Apache 2 enables name-based virtual hosts. In addition, a default virtual host is defined in the /etc/apache2/sites-enabled/000-default.conf file; this virtual host will be used if no host matching the request sent by the client is found.

> Important
>
> 
>
> Requests concerning unknown virtual hosts will always be served by the first defined virtual host, which is why the package ships a `000-default.conf` configuration file, which sorts first among all other files that you might create.

Each extra virtual host is then described by a file stored in `/etc/apache2/sites-available/`. The file is usually named after the hostname of the website followed by a .conf suffix (for example: `www.example.com.conf`). You can then enable the new virtual host with `a2ensite www.example.com`. Here is a minimal virtualhost configuration for a website whose files are stored in `/srv/www.example.com/www/` (defined with the DocumentRoot option):

```
<VirtualHost *:80>
ServerName www.example.com
ServerAlias example.com
```

```
DocumentRoot /srv/www.example.com/www
</VirtualHost>
```

You might also consider adding CustomLog and ErrorLog directives to configure Apache to output logs in files dedicated to the virtual host.

Common Directives

This section briefly reviews some of the commonly-used Apache configuration directives.

The main configuration file usually includes several Directory blocks; they allow specifying different behaviors for the server depending on the location of the file being served. Such a block commonly includes Options and AllowOverride directives:

```
<Directory /var/www>
Options Includes FollowSymLinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

The DirectoryIndex directive contains a list of files to try when the client request matches a directory. The first existing file in the list is used and sent as a response.

The Options directive is followed by a list of options to enable. The None value disables all options; correspondingly, All enables them all except MultiViews. Available options include:

- ExecCGI—indicates that CGI scripts can be executed.
- FollowSymLinks—tells the server that symbolic links can be followed, and that the response should contain the contents of the target of such links.
- SymLinksIfOwnerMatch—also tells the server to follow symbolic links, but only when the link and its target have the same owner.
- Includes—enables *Server Side Includes* (SSI). These are directives embedded in HTML pages and executed on the fly for each request.
- Indexes—tells the server to list the contents of a directory if the HTTP request sent by the client points to a directory without an index file (that is, when no files mentioned by the DirectoryIndex directive exist in this directory).
- MultiViews—enables content negotiation; this can be used by the server to return a web page matching the preferred language as configured in the browser.

Requiring Authentication In some circumstances, access to part of a website needs to be restricted, so only legitimate users who provide a username and a password are granted access to the contents.

The `.htaccess` file contains Apache configuration directives enforced each time a request concerns an element from the directory where the `.htaccess` file is stored. These directives are recursive, expanding the scope to all subdirectories.

Most of the directives that can occur in a Directory block are also legal in an `.htaccess` file. The AllowOverride directive lists all the options that can be enabled or disabled by way of `.htaccess`. A common use of this option is to restrict ExecCGI, so that the administrator chooses which users are allowed to run programs under the web server's identity (the www-data user).

Example 5.3 *.htaccess File Requiring Authentication*

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

Basic Authentication Offers No Security

The authentication system used in the above example (`Basic`) has minimal security as the password is sent in clear text (it is only encoded as *base64*, which is a simple encoding rather than an encryption method). It should also be noted that the documents protected by this mechanism also go over the network in the clear. If security is important, the entire HTTP session should be encrypted with Transport Layer Security (TLS).

The `/etc/apache2/authfiles/htpasswd-private` file contains a list of users and passwords; it is commonly manipulated with the `htpasswd` command. For example, the following command is used to add a user or change their password:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private user
New password:
Re-type new password:
Adding password for user user
```

Restricting Access The Require directive controls access restrictions for a directory (and its subdirectories, recursively).

It can be used to restrict access based on many criteria; we will stop at describing access restriction based on the IP address of the client but it can be made much more powerful than that, especially when several Require directives are combined within a RequireAll block.

For instance, you could restrict access to the local network with the following directive:

```
Require ip 192.168.0.0/16
```

5.4. Managing Services

Kali uses `systemd` as its init system, which is not only responsible for the boot sequence, but also permanently acts as a full featured service manager, starting and monitoring services.

`systemd` can be queried and controlled with `systemctl`. Without any argument, it runs the `systemctl list-units` command, which outputs a list of the active *units*. If you run `systemctl status`, the output shows a hierarchical overview of the running services. Comparing both outputs, you immediately see that there are multiple kinds of units and that services are only one among them.

Each service is represented by a *service unit*, which is described by a service file usually shipped in `/lib/systemd/system/` (or `/run/systemd/system/`, or `/etc/systemd/system/`; they are listed by increasing order of importance, and the last one wins). Each is possibly modified by other *`service-name`*`.service.d/*.conf` files in the same set of directories. Those unit files are plain text files whose format is inspired by the well-known "*.ini" files of Microsoft Windows, with *key* = *value* pairs grouped between [*section*] headers. Here we see a sample service file for `/lib/systemd/system/ssh.service`:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Target units are another part of systemd's design. They represent a desired state that you want to attain in terms of activated units (which means a running service in the case of service units). They exist mainly as a way to group dependencies on other units. When the system starts, it enables the units required to reach the `default.target` (which is a symlink to `graphical.target`, and which in turn depends on `multi-user.target`). So all the dependencies of those targets get activated during boot.

Such dependencies are expressed with the Wants directive on the target unit. But you don't have to edit the target unit to add new dependencies, you can also create a symlink pointing to the

dependent unit in the `/etc/systemd/system/`*`target-name`*`.target.wants/` directory. And this is exactly what `systemctl enable` *`foo.service`* does. When you enable a service, you tell systemd to add a dependency on the targets listed in the WantedBy entry of the [Install] section of the service unit file. Conversely, `systemctl disable` *`foo.service`* drops the same symlink and thus the dependency.

The `enable` and `disable` commands do not change anything regarding the current status of the services. They only influence what will happen at next boot. If you want to run the service immediately, you should execute `systemctl start` *`foo.service`*. Conversely, you can stop it with `systemctl stop` *`foo.service`*. You can also inspect the current status of a service with `systemctl status` *`foo.service`*, which usefully includes the latest lines of the associated log. After having changed the configuration of a service, you may wish to reload it or restart it: those operations are done with `systemctl reload` *`foo.service`* and `systemctl restart` *`foo.service`* respectively.

```
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset:
       ➥ disabled)
   Active: inactive (dead)
# ls -al /etc/systemd/system/multi-user.target.wants/postgresql.service
ls: cannot access '/etc/systemd/system/multi-user.target.wants/postgresql.service': No
    ➥ such file or directory
# systemctl enable postgresql
[...]
# ls -al /etc/systemd/system/multi-user.target.wants/postgresql.service
lrwxrwxrwx 1 root root 38 Apr 21 16:21 /etc/systemd/system/multi-user.target.wants/
    ➥ postgresql.service -> /lib/systemd/system/postgresql.service
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset:
       ➥ disabled)
   Active: inactive (dead)
# systemctl start postgresql
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset:
       ➥ disabled)
   Active: active (exited) since Thu 2016-04-21 16:22:29 EDT; 2s ago
  Process: 6355 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 6355 (code=exited, status=0/SUCCESS)

Apr 21 16:22:29 kali-rolling systemd[1]: Starting PostgreSQL RDBMS...
Apr 21 16:22:29 kali-rolling systemd[1]: Started PostgreSQL RDBMS.
```

5.5. Summary

In this chapter, we learned how to configure Kali Linux. We configured network settings, talked about users and groups, and discussed how to create and modify user accounts, set passwords, disable accounts, and manage groups. Finally, we discussed services and explained how to set up and maintain generic services, specifically SSH, PostgreSQL, and Apache.

Summary Tips:

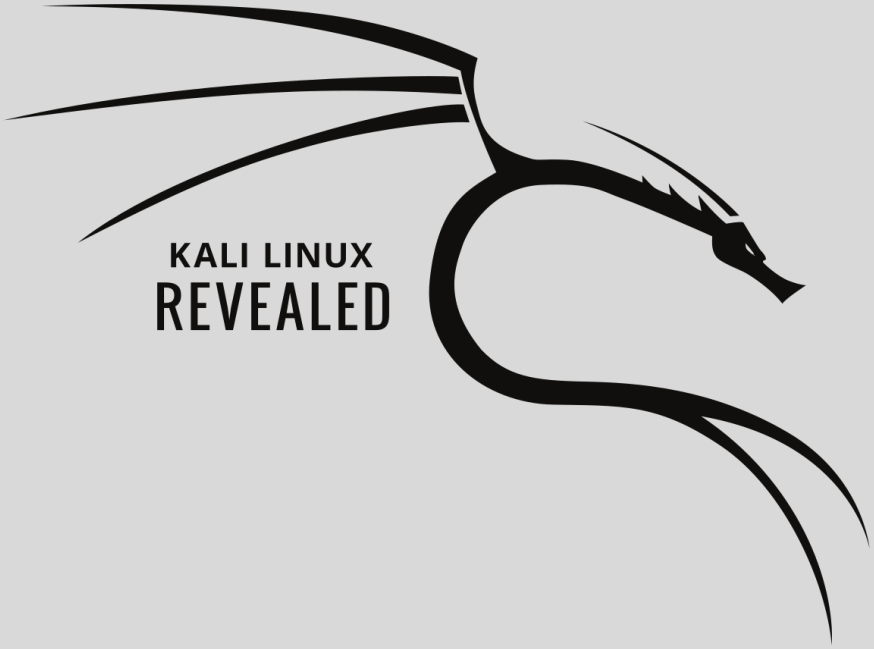
- In a typical desktop installation, you will have *NetworkManager* already installed and it can be controlled and configured through GNOME's control center and through the top-right menu.
- You can configure the network from the command line with the `ifup` and `ifdown` tools, which read their instructions from the `/etc/network/interfaces` configuration file. An even newer tool, *systemd-networkd* works with the *systemd* init system.
- By default, the database of Unix users and groups consists of the textual files `/etc/passwd` (list of users), `/etc/shadow` (encrypted passwords of users), `/etc/group` (list of groups), and `/etc/gshadow` (encrypted passwords of groups).
- You can use the `getent` command to consult the user database and other system databases.
- The `adduser` command asks a few questions before creating the account, but is a straightforward way to create a new user account.
- Several commands can be used to modify specific fields in the user database including: `passwd` (change password), `chfn` (change full name and the GECOS, or general information field), `chsh` (change login shell), `chage` (change password age), and `passwd -e` *`user`* (forces the user to change their password the next time they log in).
- Each user can be a member of one or multiple groups. Several commands can be used to modify group identity: `newgrp` changes the current group ID, `sg` executes a command using the supplied alternate group, the setgid bit can be placed on a directory, causing files created in that directory to automatically belong to the correct group. In addition, the `id` command displays the current state of a user including a list of their group membership.
- You can manually start SSH with `systemctl start ssh` or permanently enable it with `systemctl enable ssh`. The default configuration disables password-based logins for the root user, which means you must first setup SSH keys with `ssh-keygen`.
- PostgreSQL is a database server. It is rarely useful on its own but is used by many other services to store data.
- A typical Kali Linux installation includes the Apache web server, provided by the *apache2* package. Being a network service, it is disabled by default. You can manually start it with `systemctl start apache2`.

- With its default configuration, Apache listens on port 80 (as configured in `/etc/apache2/ports.conf`), and serves pages from the `/var/www/html/` directory by default (as configured in `/etc/apache2/sites-enabled/000-default.conf`).

Now that we have tackled Linux fundamentals and Kali Linux installation and configuration, let's discuss how to troubleshoot Kali and teach you some tools and tricks to get you back up and running when you run into problems.

Keywords

Documentation
Forums
IRC channel
Bug report



Chapter

6

Helping Yourself and Getting Help

Contents

No matter how many years of experience you have, there is no doubt that—sooner or later—you will encounter a problem. Solving that problem is then often a matter of understanding it and then taking advantage of various resources to find a solution or work-around.

In this chapter, we will discuss the various information sources available and discuss the best strategies for finding the help you need or the solution to a problem you might be facing. We will also take you on a tour of some of the Kali Linux community resources available, including the web forums and Internet Relay Chat (IRC) channel. Lastly, we will introduce bug reporting and show you how to take advantage of bug filing systems to troubleshoot problems and lay out strategies to help you file your own bug report so that undocumented issues can be handled quickly and effectively.

6.1. Documentation Sources

Before you can understand what is really going on when there is a problem, you need to know the theoretical role played by each program involved in the problem. One of the best ways to do this is to review the program's documentation. Let's begin by discussing where, exactly, you can find documentation since it is often scattered.

How to Avoid RTFM Answers

This acronym stands for "read the f***ing manual," but can also be expanded in a friendlier variant, "read the fine manual." This phrase is sometimes used in (terse) responses to questions from newbies. It is rather abrupt, and betrays a certain annoyance at a question asked by someone who has not even bothered to read the documentation. Some say that this classic response is better than no response at all since this at least hints that the answer lies within the documentation.

When you are posting questions, don't necessarily be offended by the occasional RTFM response, but do what you can to at least show that you have taken the time to do some research before posting the question; mention the sources that you have consulted and describe the various steps that you have personally taken to find information. This will go a long way to show that you are not lazy and are truly seeking knowledge. Following Eric Raymond's guidelines is a good way to avoid the most common mistakes and get useful answers.

➨ `http://catb.org/~esr/faqs/smart-questions.html`

6.1.1. Manual Pages

Manual pages, while relatively terse in style, contain a great deal of essential information. To view a manual page, simply type `man` *`manual-page`*. The manual page usually coincides with the command name. For example, to learn about the possible options for the `cp` command, you would type `man cp` at the command prompt.

Man pages not only document programs accessible from the command line, but also configuration files, system calls, C library functions, and so forth. Sometimes names can collide. For example,

the shell's `read` command has the same name as the `read` system call. This is why manual pages are organized in the following numbered sections:

1. Commands that can be executed from the command line
2. System calls (functions provided by the kernel)
3. Library functions (provided by system libraries)
4. Devices (on Unix-like systems, these are special files, usually placed in the `/dev/` directory)
5. Configuration files (formats and conventions)
6. Games
7. Sets of macros and standards
8. System administration commands
9. Kernel routines

You can specify the section of the manual page that you are looking for: to view the documentation for the `read` system call, you would type `man 2 read`. When no section is explicitly specified, the first section that has a manual page with the requested name will be shown. Thus, `man shadow` returns `shadow(5)` because there are no manual pages for *shadow* in sections 1–4.

Of course, if you do not know the names of the commands, the manual is not going to be of much use to you. Enter the `apropos` command, which searches manual pages (or more specifically their short descriptions) for any keywords that you provide. The `apropos` command then returns a list of manual pages whose summary mentions the requested keywords along with the one-line summary from the manual page. If you choose your keywords well, you will find the name of the command that you need.

Example 6.1 *Finding cp with apropos*

```
$ apropos "copy file"
cp (1)               - copy files and directories
cpio (1)             - copy files to and from archives
gvfs-copy (1)        - Copy files
gvfs-move (1)        - Copy files
hcopy (1)            - copy files from or to an HFS volume
install (1)          - copy files and set attributes
ntfscp (8)           - copy file to an NTFS volume.
```

Browsing Documentation by Following Links

Many manual pages have a "See Also" section, usually near the end of the document, which refers to other manual pages relevant to similar commands, or to external documentation. You can use this section to find relevant documentation even when the first choice is not optimal.

In addition to `man`, you can use `konqueror` (in KDE) and `yelp` (in GNOME) to search man pages as well.

6.1.2. Info Documents

The GNU project has written manuals for most of its programs in the *info* format; this is why many manual pages refer to the corresponding *info* documentation. This format offers some advantages but the default program to view these documents (also called `info`) is slightly more complex. You would be well advised to use `pinfo` instead (from the *pinfo* package). To install it, simply run `apt update` followed by `apt install pinfo` (see section 8.2.2.2, "Installing Packages with APT" [page 177]).

The *info* documentation has a hierarchical structure and if you invoke `pinfo` without parameters, it will display a list of the nodes available at the first level. Usually, nodes bear the name of the corresponding commands.

You can use the arrow keys to navigate between nodes. Alternatively, you could also use a graphical browser (which is a lot more user-friendly) such as `konqueror` or `yelp`.

As far as language translations are concerned, the *info* system is always in English and is not suitable for translation, unlike the `man` page system. However, when you ask the `pinfo` program to display a non-existing *info* page, it will fall back on the *man* page by the same name (if it exists), which might be translated.

6.1.3. Package-Specific Documentation

Each package includes its own documentation and even the least documented programs generally have a `README` file containing some interesting and/or important information. This documentation is installed in the `/usr/share/doc/`*`package`*`/` directory (where *package* represents the name of the package). If the documentation is particularly large, it may not be included in the program's main package, but might be offloaded to a dedicated package which is usually named *package*-doc. The main package generally recommends the documentation package so that you can easily find it.

The `/usr/share/doc/`*`package`*`/` directory also contains some files provided by Debian, which complete the documentation by specifying the package's particularities or improvements compared to a traditional installation of the software. The `README.Debian` file also indicates all of the adaptations that were made to comply with the Debian Policy. The `changelog.Debian.gz` file allows the user to follow the modifications made to the package over time; it is very useful to try to understand what has changed between two installed versions that do not have the same behavior. Finally, there is sometimes a `NEWS.Debian.gz` file that documents the major changes in the program that may directly concern the administrator.

6.1.4. Websites

In many cases, you can find websites that are used to distribute free software programs and to bring together the community of its developers and users. These sites are loaded with relevant information in various forms such as official documentation, frequently asked questions (FAQ), and mailing list archives. In most cases, the FAQ or mailing list archives address problems that you have encountered. As you search for information online, it is immensely valuable to master search syntax. One quick tip: try restricting a search to a specific domain, like the one dedicated to the program that is giving you trouble. If the search returns too many pages or if the results do not match what you seek, you can add the keyword **`kali`** or **`debian`** to limit results and target relevant information.

From the Error to a Solution

If the software returns a very specific error message, enter it into a search engine (between double quotes, ", in order to search for the complete phrase, rather than the individual keywords). In most cases, the first links returned will contain the answer that you need.

In other cases, you will get very general errors, such as "Permission denied". In this case, it is best to check the permissions of the elements involved (files, user ID, groups, etc.). In short, don't get in the habit of always using a search engine to find a solution to your problem. You will find it is much too easy to forget to use common sense.

If you do not know the address of the software's website, there are various means of locating it. First, look for a Homepage field in the package's meta-information (`apt show` *`package`*). Alternatively, the package description may contain a link to the program's official website. If no URL is indicated, the package maintainer may have included a URL in the `/usr/share/doc/`*`package`*`/copyright` file. Finally, you may be able to use a search engine (such as Google, DuckDuckGo, Yahoo, etc.) to find the software's website.

6.1.5. Kali Documentation at docs.kali.org

The Kali project maintains a collection of useful documentation at `http://docs.kali.org`. While this book covers a large part of what you should know about Kali Linux, the documentation there might still be useful as it contains step-by-step instructions (much like how-tos) on many topics.

➨ `http://docs.kali.org/`

Let's review the various topics covered there:

- Getting started: a series of instructions, including download instructions, for those new to Kali
- Kali Linux Live: documentation describing how to use Kali Linux as a live system
- Installing Kali Linux: various documents describing Kali Linux installation, including how to install it side-by-side with other operating systems

- Kali Linux on ARM: many recipes about running Kali Linux on various ARM-based devices
- Using Kali Linux: multiple how-tos covering many common requests
- Customizing Kali Linux: instructions for the tinkerers who like to rebuild Kali based on their own requirements
- Kali Community Support: pointers to the various communities where you can get support and explanations on how to submit bug reports
- Kali Linux Policies: explanations about what makes Kali Linux special when compared to other Linux distributions
- The Kali Linux Dojo: videos of Black Hat and DEF CON workshops

6.2. Kali Linux Communities

There are many Kali Linux communities around the world using many different tools to communicate (forums and social networks, for example). In this section, we will only present two official Kali Linux communities.

6.2.1. Web Forums on forums.kali.org

The official community forums for the Kali Linux project are located at forums.kali.org[1]. Like every web-based forum, you must create an account to be able to post and the system remembers what posts you have already seen, making it easy to follow conversations on a regular basis.

Before posting, you should read the forum rules:

➨ `http://docs.kali.org/community/kali-linux-community-forums`

We won't copy them here but it is worth noting that you are not allowed to speak about illegal activities such as breaking into other people's networks. You must be respectful of other community members so as to create a welcoming community. Advertising is banned and off-topic discussions are to be avoided. There are enough categories to cover everything that you would like to discuss about Kali Linux.

6.2.2. #kali-linux IRC Channel on Freenode

IRC is a real-time chat system. Discussions happen in chat rooms that are called *channels* and are usually centered around a particular topic or community. The Kali Linux project uses the #kali-linux channel on the Freenode[2] network (you can use chat.freenode.net as IRC server, on port 6667 for a TLS-encrypted connection or port 6666 for a clear-text connection).

[1] `http://forums.kali.org`
[2] `http://www.freenode.net`

To join the discussions on IRC, you have to use an IRC client such as `hexchat` (in graphical mode) or `irssi` (in console mode). There is also a web-based client available on webchat.freenode.net[3].

While it is really easy to join the conversation, you should be aware that IRC channels have their own rules and that there are channel operators (their nickname is prefixed with @) who can enforce the rules: they can kick you out of the channel (or even ban you if you continue to disobey the rules). The #kali-linux channel is no exception. The rules have been documented here:

➠ `http://docs.kali.org/community/kali-linux-irc-channel`

To summarize the rules: you have to be friendly, tolerant, and reasonable. You should avoid off-topic discussions. In particular, discussions about illegal activities, warez/cracks/pirated software, politics, and religions are forbidden. Keep in mind that your IP address will be available to others.

If you want to ask for help, follow the recommendations listed in "How to Avoid RTFM Answers" [page 124]: do your research first and share the results. When you are asked for supplementary information, please provide it accurately (if you must provide some verbose output, don't paste it in the channel directly, instead use a service like Pastebin[4] and post only the Pastebin URL).

Do not expect an immediate answer. Even though IRC is a real-time communication platform, participants log in from all over the world, so time zones and work schedules vary. It may take a few minutes or hours for someone to respond to your question. However, when others include your nickname in a reply, your nick will be highlighted and most IRC clients will notify you, so leave your client connected and be patient.

6.3. Filing a Good Bug Report

If all of your efforts to resolve a problem fail, it is possible that the problem is due to a bug in the program. In this case, the problem may have resulted in a bug report. You can search for bug reports to find a solution to your problem but let's take a look at the procedure of reporting a bug to Kali, Debian, or directly to the upstream developers so you understand the process should you need to submit your own report.

The goal of a bug report is to provide enough information so that the developers or maintainers of the (supposedly) faulty program can reproduce the problem, debug its behavior, and develop a fix. This means that your bug report must contain appropriate information and must be directed to the correct person or project team. The report must also be well-written and thorough, ensuring a faster response.

The exact procedure for the bug report will vary depending on where you will submit the report (Kali, Debian, upstream developers) but there are some generic recommendations that apply to all cases. In this chapter we will discuss those recommendations.

[3] `http://webchat.freenode.net`
[4] `http://pastebin.com`

6.3.1. Generic Recommendations

Let's discuss some general recommendations and guidelines that will help you submit a bug report that is clear, comprehensive, and improves the chances that the bug will be addressed by the developers in a timely fashion.

How to Communicate

Write Your Report in English The Free Software community is international and unless you know your interlocutor, you should be using plain English. If you are a native speaker of English, use simple sentences and avoid constructions that might be hard to understand for people with limited English skills. Even though most developers are highly intelligent, not all of them have strong English language skills. It is best never to assume.

Be Respectful of the Developers' Work Remember that most Free Software developers (including those behind Kali Linux) are benevolent and are spending their limited free time to work on the software that you are freely using. Many are doing this out of altruism. Thus, when you file a bug report, be respectful (even if the bug looks like an obvious mistake by the developer) and don't assume that they owe you a fix. Thank them for their contribution instead.

If you know how to modify and recompile the software, offer to assist the developers in testing any patches that they submit to you. This will show them that you are willing to invest your own time as well.

Be Reactive and Ready to Provide More Information In some cases, the developer will come back to you with requests for more information or requests for you to try to re-create the problem perhaps by using different options or using an updated package. You should try to respond to those queries as quickly as possible. The quicker you submit your response, the higher the chance that they will be able to solve it quickly while the initial analysis is still fresh in their mind.

While you should aim to respond quickly, you should also not go too fast: the data submitted must be correct and it must contain everything that the developers requested. They will be annoyed if they have to request something a second time.

What to Put in the Bug Report

Instructions to Reproduce the Problem To be able to reproduce the issue, the developers need to know what you are using, where you got it from, and how you installed it.

You should provide precise, step-by-step instructions describing how to reproduce the problem. If you need to use some data to reproduce the problem, attach the corresponding file to the bug report. Try to come up with the minimal set of instructions needed to reproduce the bug.

Give Some Context and Set Your Expectations Explain what you were trying to do and how you expected the program to behave.

In some cases, the bug is only triggered because you were using the program in a way that it was not designed to operate by the developers. By explaining what you were trying to achieve, you will allow the developers to clearly see when this is the case.

In some other cases, the behavior that you describe as a bug might actually be the normal behavior. Be explicit about what you expected the program to do. This will clarify the situation for the developers. They may either improve the behavior or improve the documentation, but at least they know that the behavior of their program is confusing some users!

Be Specific Include the versions numbers of the software that you use, possibly with the version numbers of their dependencies. When you refer to something that you downloaded, include its complete URL.

When you get an error message, quote it exactly as you saw it. If possible, include a copy of your screen output or a screenshot. Include a copy of any relevant log file, ensuring that you remove any sensitive data first.

Mention Possible Fixes or Workarounds Before filing the bug report, you probably tried to resolve the problem. Explain what you tried and what results you received. Be very clear about what is a fact and what was just a hypothesis on your part.

If you did an Internet search and found some explanations about a similar problem, you can mention them, in particular when you found other similar bug reports in the Debian bug tracker or in the upstream bug tracker.

If you found a way of achieving the desired result without triggering the bug, please document that as well. This will help other users who are hit by the same issue.

Long Bug Reports Are Fine A two-line bug report is insufficient; providing all the information needed usually requires several paragraphs (or sometimes pages) of text.

Supply all the information you can. Try to stick to what is relevant, but if you are uncertain, too much is better than too little.

If your bug report is really long, take some time to structure the content and provide a short summary at the start.

Miscellaneous Tips

Avoid Filing Duplicate Bug Reports In the Free Software world, all bug trackers are public. Open issues can be browsed and they even have a search feature. Thus, before filing a new bug report, try to determine if your problem has already been reported by someone else.

If you find an existing bug report, subscribe to it and possibly add supplementary information. Do not post comments such as "Me too" or "+1"; they serve no purpose. But you can indicate that you are available for further tests if the original submitter did not offer this.

If you have not found any report of your problem, go ahead and file it. If you have found related tickets, be sure to mention them.

Ensure You Use the Latest Version It is very frustrating for developers to receive bug reports for problems that they have already solved or problems that they can't reproduce with the version that they are using (developers almost always use the latest version of their product). Even when older versions are maintained by the developers, the support is often limited to security fixes and major problems. Are you sure that your bug is one of those?

That is why, before filing a bug report, you should make sure that you are using the latest version of the problematic system and application and that you can reproduce the problem in that situation.

If Kali Linux does not offer the latest version of the application (neither in kali-rolling nor in kali-bleeding-edge, see section 8.1.3.3, "The Kali-Bleeding-Edge Repository" [page 174]), you have alternative solutions: you can try a manual installation of the latest version in a throw-away virtual machine, or you can review the upstream ChangeLog (or Git commit history) to see that there hasn't been any change that could fix the problem that you are seeing (and then file the bug even though you did not try the latest version).

Do Not Mix Multiple Issues in a Single Bug Report File one bug report per issue. That way, the subsequent discussions do not get too messy and each bug can be fixed according to its own schedule. If you don't do that, either the single bug needs to be repurposed multiple times and can only be closed when all issues have been fixed, or the developers must file the supplementary reports that you should have created in the first place.

6.3.2. Where to File a Bug Report

To be able to decide where to file the bug report, you must have a good understanding of the problem and you must have identified in which piece of software the problem lies.

Ideally, you track the problem down to a file on your system and then you can use `dpkg` to find out which package owns that file and where that package comes from. Let's assume that you found a bug in a graphical application. After looking at the list of running processes (the output of `ps auxf`), you discovered that the application was started with the `/usr/bin/sparta` executable:

```
$ dpkg -S /usr/bin/sparta
sparta: /usr/bin/sparta
$ dpkg -s sparta | grep ^Version:
Version: 1.0.1+git20150729-0kali1
```

You learn that `/usr/bin/sparta` is provided by the sparta package, which is in version 1.0.1+git 20150729-0kali1. The fact that the version string contains kali indicates to you that the package

comes from Kali Linux (or is modified by Kali Linux). Any package that does not have kali in its version string (or in its package name) comes straight from Debian (Debian Testing in general).

Double Check Before Filing Bugs against Debian

If you find a bug in a package imported straight from Debian, it should ideally be reported and fixed on the Debian side. However, before doing this, ensure that the problem is reproducible on a plain Debian system since Kali may have caused the problem by modifying other packages or dependencies.

The easiest way to accomplish this is to setup a virtual machine running Debian Testing. You can find an installation ISO for Debian Testing on the Debian Installer website:

➨ `https://www.debian.org/devel/debian-installer/`

If you can confirm the problem in the virtual machine, then you can submit the bug to Debian by running `reportbug` within the virtual machine and following the instructions provided.

Most bug reports about the behavior of applications should be directed to their upstream projects except when facing an integration problem: in that case, the bug is a mistake in the way the software gets packaged and integrated into Debian or Kali. For example, if an application offers compile-time options that the package does not enable or the application does not work because of a missing library (thus putting into light a missing dependency in the package meta-information), you may be facing an integration problem. When you don't know what kind of problem you face, it is usually best to file the issue on both sides and to cross-reference them.

Identifying the upstream project and finding where to file the bug report is usually easy. You just have to browse the upstream website, which is referenced in the Homepage field of the packaging meta-data:

```
$ dpkg -s sparta | grep ^Homepage:
Homepage: https://github.com/SECFORCE/sparta
```

6.3.3. How to File a Bug Report

Filing a Bug Report in Kali

Kali uses a web-based bug tracker at `http://bugs.kali.org` where you can consult all the bug reports anonymously, but if you would like to comment or file a new bug report, you will need to register an account.

Signing Up for a Bug Tracker Account To begin, simply click *Signup for new account* on the bug tracker website, as shown in Figure 6.1, "Kali Bug Tracker Start Page" [page 134].

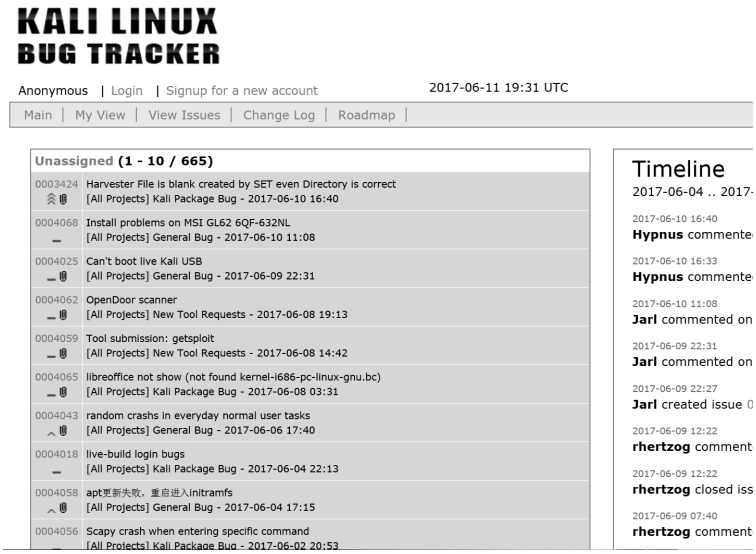


Figure 6.1 *Kali Bug Tracker Start Page*

Next, provide a username, e-mail address, and response to the CAPTCHA challenge. Then click the Signup button to proceed (Figure 6.2, "Signup Page" [page 134]).

Figure 6.2 *Signup Page*

If successful, the next page (Figure 6.3, "Signup Confirmation Page" [page 135]) will notify you that the account registration has been processed, and the bug tracker system will send a confirmation email to the address you provided. You will need to click the link in the email in order to activate your account.

Once your account has been activated, click Proceed to continue to the bug tracker login page.

Figure 6.3 *Signup Confirmation Page*

Creating the Report To begin your report, log into your account and click the Report Issue link on the landing page. You will be presented a form with many fields to fill, as shown in Figure 6.4, "Form to report a bug" [page 136].

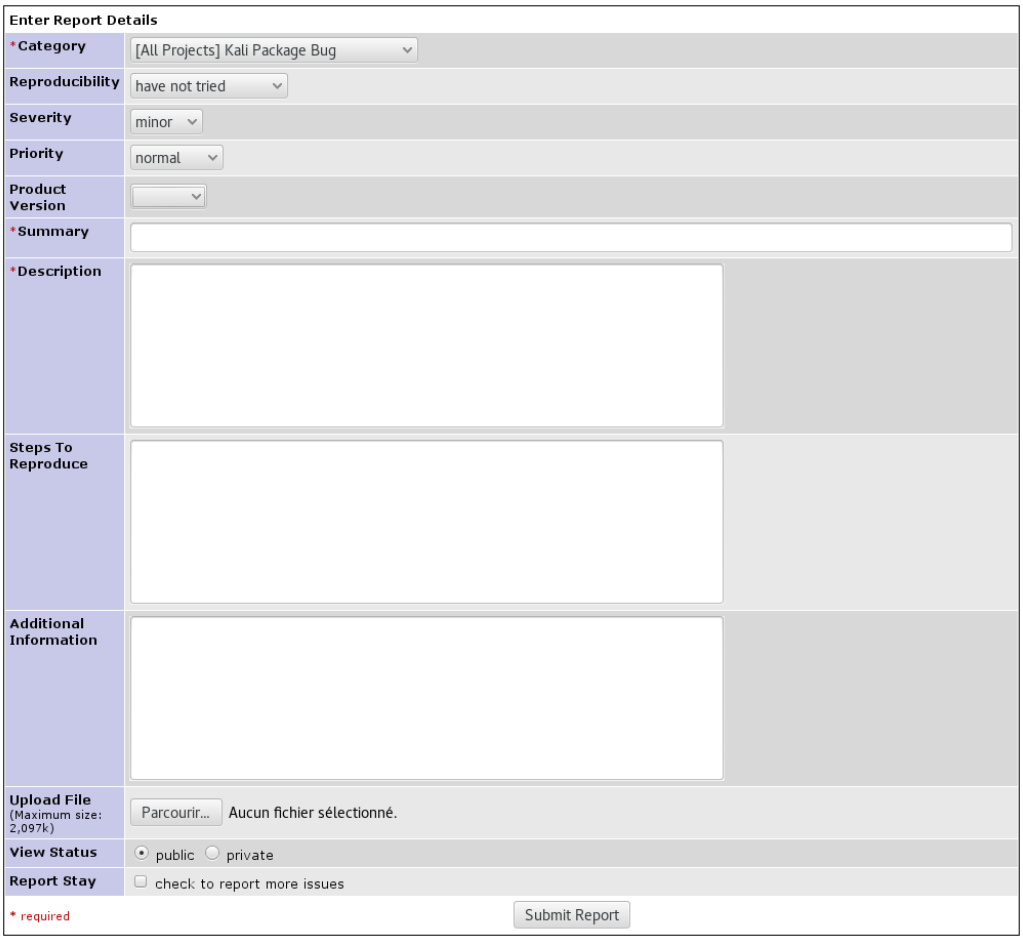


Figure 6.4 *Form to report a bug*

Here is a rundown of all the fields on the form:

Category (mandatory) This field describes the category of the bug you are submitting. Reports that can be attributed to a specific package should be filed in the Kali Package Bug or Kali Package Improvement categories. Other reports should use the General Bug or Feature Requests categories. The remaining categories are for specific use cases: Tool Upgrade can be used to notify the Kali developers of the availability of a new version of a software packaged

in Kali. New Tool Requests can be used to suggest new tools to package and integrate in the Kali distribution.

Reproducibility This field documents whether the problem is reproducible in a predictable way or if it happens only somewhat randomly.

Severity and Priority Those fields are best left unmodified as they are mainly for the developers. They can use them to sort the list of issues according to the severity of the problem and to the priority at which it must be handled.

Product Version This field should indicate what version of Kali Linux you are running (or the one which is the closest to what you are running). Think twice before reporting an issue on an old release that is no longer supported.

Summary (mandatory) This is essentially the title of your bug report and it is the first thing that people will see. Make sure that it conveys the reason why you are filing the report. Avoid generic descriptions like "X doesn't work" and opt instead for "X fails with error Y under condition Z."

Description (mandatory) This is the body of your report. Here you should enter all of the information you collected about the problem that you are experiencing. Don't forget all the recommendations given in the former section.

Steps to Reproduce In this field, list all the detailed instructions explaining how to trigger the problem.

Additional Information In this section, you can provide any additional information you believe is relevant to the issue. If you have a fix or workaround for the issue, please provide it in this section.

Upload File Not everything can be explained with plain text. This field lets you attach arbitrary files to your reports: screenshots to show the error, sample documents triggering the problem, log files, etc.

View Status Leave that field set to "public" so that everybody can see your bug report. Use "private" only for security-related reports containing information about undisclosed security vulnerabilities.

Filing a Bug Report in Debian

Debian uses a (mostly) email-based bug tracking system known as Debbugs. To open a new bug report, you will send an email (with a special syntax) to submit@bugs.debian.org. This will allocate a bug number *XXXXXX* and inform you that you can send additional information by mailing *XXX XXX*@bugs.debian.org. Each bug is associated to a Debian package. You can browse all the bugs of

a given package (including the bug that you are thinking of reporting) at https://bugs.debian.org/*package*. You can check the history of a given bug at https://bugs.debian.org/*XXXXXX*.

Setting Up Reportbug While you can open a new bug with a simple e-mail, we recommend using `reportbug` because it will help you draft a solid bug report with all the required information. Ideally, you should run it from a Debian system (for example, in the virtual machine where you reproduced the problem).

The first run of `reportbug` starts a configuration script. First, select a skill level. You should choose Novice or Standard; we use the latter because it offers more fine-grained control. Next, select an interface and enter your personal details. Finally, select a user interface. The configuration script will allow you to use a local mail transport agent, an SMTP server, or as a last resort, a Debian SMTP server.

```
Welcome to reportbug! Since it looks like this is the first time you have
used reportbug, we are configuring its behavior. These settings will be
saved to the file "/root/.reportbugrc", which you will be free to edit
further.
Please choose the default operating mode for reportbug.

1 novice    Offer simple prompts, bypassing technical questions.

2 standard  Offer more extensive prompts, including asking about things
            that a moderately sophisticated user would be expected to
            know about Debian.

3 advanced  Like standard, but assumes you know a bit more about Debian,
    ➥ including "incoming".

4 expert    Bypass most handholding measures and preliminary triage
            routines. This mode should not be used by people unfamiliar
            with Debian's policies and operating procedures.

Select mode: [novice] standard
Please choose the default interface for reportbug.

1 text   A text-oriented console user interface

2 gtk2   A graphical (GTK+) user interface.

3 urwid  A menu-based console user interface

Select interface: text
Will reportbug often have direct Internet access? (You should answer
yes to this question unless you know what you are doing and plan to
check whether duplicate reports have been filed via some other channel.)
```

```
[Y|n|q|?]? Y
What real name should be used for sending bug reports?
[root]> Raphaël Hertzog
Which of your email addresses should be used when sending bug reports?
(Note that this address will be visible in the bug tracking system, so you
may want to use a webmail address or another address with good spam
filtering capabilities.)
[root@localhost.localdomain]> buxy@kali.org
Do you have a "mail transport agent" (MTA) like Exim, Postfix or SSMTP
configured on this computer to send mail to the Internet? [y|N|q|?]? N
Please enter the name of your SMTP host. Usually it's called something
like "mail.example.org" or "smtp.example.org". If you need to use a
different port than default, use the <host>:<port> alternative
format. Just press ENTER if you don't have one or don't know, and so a
Debian SMTP host will be used.
>
Please enter the name of your proxy server. It should only use this
parameter if you are behind a firewall. The PROXY argument should be
formatted as a valid HTTP URL, including (if necessary) a port number; for
example, http://192.168.1.1:3128/. Just press ENTER if you don't have one
or don't know.
>
Default preferences file written. To reconfigure, re-run reportbug with
the "--configure" option.
```

Using Reportbug With the setup phase completed, the actual bug report can begin. You will be prompted for a package name, although you can also provide the package name directly on the command line with `reportbug` *`package`*).

```
Running 'reportbug' as root is probably insecure! Continue [y|N|q|?]? y
Please enter the name of the package in which you have found a problem, or
type 'other' to report a more general problem. If you don't know what
package the bug is in, please contact debian-user@lists.debian.org for
assistance.
> wireshark
```

Contrary to the advice given above, if you don't know against which package to file the bug, you should get in touch with a Kali support forum (described in section 6.2, "Kali Linux Communities" [page 128]). In the next step, `reportbug` downloads the list of bugs filed against the given package and lets you browse them to see if you can find yours.

```
*** Welcome to reportbug.  Use ? for help at prompts. ***
Note: bug reports are publicly archived (including the email address of
the submitter).
Detected character set: UTF-8
```

```
Please change your locale if this is incorrect.

Using '"Raphaël Hertzog" <buxy@kali.org>' as your from address.
Getting status for wireshark...
Verifying package integrity...
Checking for newer versions at madison...
Will send report to Debian (per lsb_release).
Querying Debian BTS for reports on wireshark (source)...
35 bug reports found:

Bugs with severity important
   1) #478200  tshark: seems to ignore read filters when writing to…
   2) #776206  mergecap: Fails to create output file > 2GB
   3) #780089  wireshark: "On gnome wireshark has not title bar. Does…
Bugs with severity normal
   4) #151017  ethereal: "Protocol Hierarchy Statistics" give misleading…
   5) #275839  doesn't correctly dissect ESMTP pipelining
[...]
  35) #815122  wireshark: add OID 1.3.6.1.4.1.11129.2.4.2
(24-35/35) Is the bug you found listed above [y|N|b|m|r|q|s|f|e|?]? ?
y - Problem already reported; optionally add extra information.
N - (default) Problem not listed above; possibly check more.
b - Open the complete bugs list in a web browser.
m - Get more information about a bug (you can also enter a number
    without selecting "m" first).
r - Redisplay the last bugs shown.
q - I'm bored; quit please.
s - Skip remaining problems; file a new report immediately.
f - Filter bug list using a pattern.
e - Open the report using an e-mail client.
? - Display this help.
(24-35/35) Is the bug you found listed above [y|N|b|m|r|q|s|f|e|?]? n
Maintainer for wireshark is 'Balint Reczey <balint@balintreczey.hu>'.
Looking up dependencies of wireshark...
```

If you find your bug already filed, you can choose to send supplementary information, otherwise, you are invited to file a new bug report:

```
Briefly describe the problem (max. 100 characters allowed). This will be
the bug email subject, so keep the summary as concise as possible, for
example: "fails to send email" or "does not start with -q option
specified" (enter Ctrl+c to exit reportbug without reporting a bug).
> does not dissect protocol foobar
Rewriting subject to 'wireshark: does not dissect protocol foobar'
```

After providing a one-line summary of your problem, you must rate its severity along an extended scale:

```
How would you rate the severity of this problem or report?

1 critical        makes unrelated software on the system (or the whole
                  system) break, or causes serious data loss, or
                  introduces a security hole on systems where you install
                  the package.
2 grave           makes the package in question unusable by most or all
                  users, or causes data loss, or introduces a security
                  hole allowing access to the accounts of users who use
                  the package.
3 serious         is a severe violation of Debian policy (that is, the
                  problem is a violation of a 'must' or 'required'
                  directive); may or may not affect the usability of the
                  package. Note that non-severe policy violations may be
                  'normal,' 'minor,' or 'wishlist' bugs. (Package
                  maintainers may also designate other bugs as 'serious' and
                  thus release-critical; however, end users should not do
                  so.). For the canonical list of issues worthing a serious
                  severity you can refer to this webpage:
                  http://release.debian.org/testing/rc_policy.txt
4 important       a bug which has a major effect on the usability of a
                  package, without rendering it completely unusable to
                  everyone.
5 does-not-build  a bug that stops the package from being built from source.
                  (This is a 'virtual severity'.)
6 normal          a bug that does not undermine the usability of the whole
                  package; for example, a problem with a particular option
                  or menu item.
7 minor           things like spelling mistakes and other minor cosmetic
                  errors that do not affect the core functionality of the
                  package.
8 wishlist        suggestions and requests for new features.

Please select a severity level: [normal]
```

If you are unsure, just keep the default severity of normal.

You can also tag your report with a few keywords:

```
Do any of the following apply to this report?

1 d-i       This bug is relevant to the development of debian-installer.
2 ipv6      This bug affects support for Internet Protocol version 6.
3 l10n      This bug reports a localization/internationalization issue.
4 lfs       This bug affects support for large files (over 2 gigabytes).
5 newcomer  This bug has a known solution but the maintainer requests someone
            else implement it.
```

```
6 patch     You are including a patch to fix this problem.
7 upstream  This bug applies to the upstream part of the package.
8 none

Please select tags: (one at a time) [none]
```

Most tags are rather esoteric, but if your report includes a fix, you should select the patch tag.

Once this is completed, `reportbug` opens a text editor with a template that you should edit (Example 6.2, "Template generated by `reportbug`" [page 142]). It contains a few questions that you should delete and answer, as well as some information about your system that has been automatically collected. Notice how the first few lines are structured. They should not be modified as they will be parsed by the bug tracker to assign the report to the correct package.

Example 6.2 *Template generated by `reportbug`*

```
Subject: wireshark: does not dissect protocol foobar

Package: wireshark
Version: 2.0.2+ga16e22e-1
Severity: normal

Dear Maintainer,

*** Reporter, please consider answering these questions, where appropriate ***

   * What led up to the situation?
   * What exactly did you do (or not do) that was effective (or
     ineffective)?
   * What was the outcome of this action?
   * What outcome did you expect instead?

*** End of the template - remove these template lines ***


-- System Information:
Debian Release: stretch/sid
  APT prefers testing
  APT policy: (500, 'testing')
Architecture: amd64 (x86_64)
Foreign Architectures: i386

Kernel: Linux 4.4.0-1-amd64 (SMP w/4 CPU cores)
Locale: LANG=fr_FR.utf8, LC_CTYPE=fr_FR.utf8 (charmap=UTF-8)
Shell: /bin/sh linked to /bin/dash
Init: systemd (via /run/systemd/system)
```

```
Versions of packages wireshark depends on:
ii  wireshark-qt  2.0.2+ga16e22e-1

wireshark recommends no packages.

wireshark suggests no packages.

-- no debconf information
```

Once you save the report and close the text editor, you return to `reportbug`, which provides many other options and offers to send the resulting report.

```
Spawning sensible-editor...
Report will be sent to "Debian Bug Tracking System" <submit@bugs.debian.org>
Submit this report on wireshark (e to edit) [Y|n|a|c|e|i|l|m|p|q|d|t|s|?]? ?
Y - (default) Submit the bug report via email.
n - Don't submit the bug report; instead, save it in a temporary file (exits reportbug).
a - Attach a file.
c - Change editor and re-edit.
e - Re-edit the bug report.
i - Include a text file.
l - Pipe the message through the pager.
m - Choose a mailer to edit the report.
p - print message to stdout.
q - Save it in a temporary file and quit.
d - Detach an attachment file.
t - Add tags.
s - Add a X-Debbugs-CC recipient (a CC but after BTS processing).
? - Display this help.
Submit this report on wireshark (e to edit) [Y|n|a|c|e|i|l|m|p|q|d|t|s|?]? Y
Saving a backup of the report at /tmp/reportbug-wireshark-backup-20160328-19073-87oJWJ
Connecting to reportbug.debian.org via SMTP...

Bug report submitted to: "Debian Bug Tracking System" <submit@bugs.debian.org>
Copies will be sent after processing to:
  buxy@kali.org

If you want to provide additional information, please wait to receive the
bug tracking number via email; you may then send any extra information to
n@bugs.debian.org (e.g. 999999@bugs.debian.org), where n is the bug
number. Normally you will receive an acknowledgement via email including
the bug report number within an hour; if you haven't received a
confirmation, then the bug reporting process failed at some point
(reportbug or MTA failure, BTS maintenance, etc.).
```

There is a large diversity of free software projects, using different workflows and tools. This diversity also applies to the bug trackers in use. While many projects are hosted on GitHub and use GitHub Issues to track their bugs, there are also many others hosting their own trackers, based on Bugzilla, Trac, Redmine, Flyspray, and others. Most of them are web-based and require you to register an account to submit a new ticket.

We will not cover all the trackers here. It is up to you to learn the specifics of various trackers for other free software projects, but since GitHub is relatively popular, we will take a brief look at it here. As with other trackers, you must first create an account and sign in. Next, click the Issues tab, as shown in Figure 6.5, "Main page of a GitHub project" [page 144].

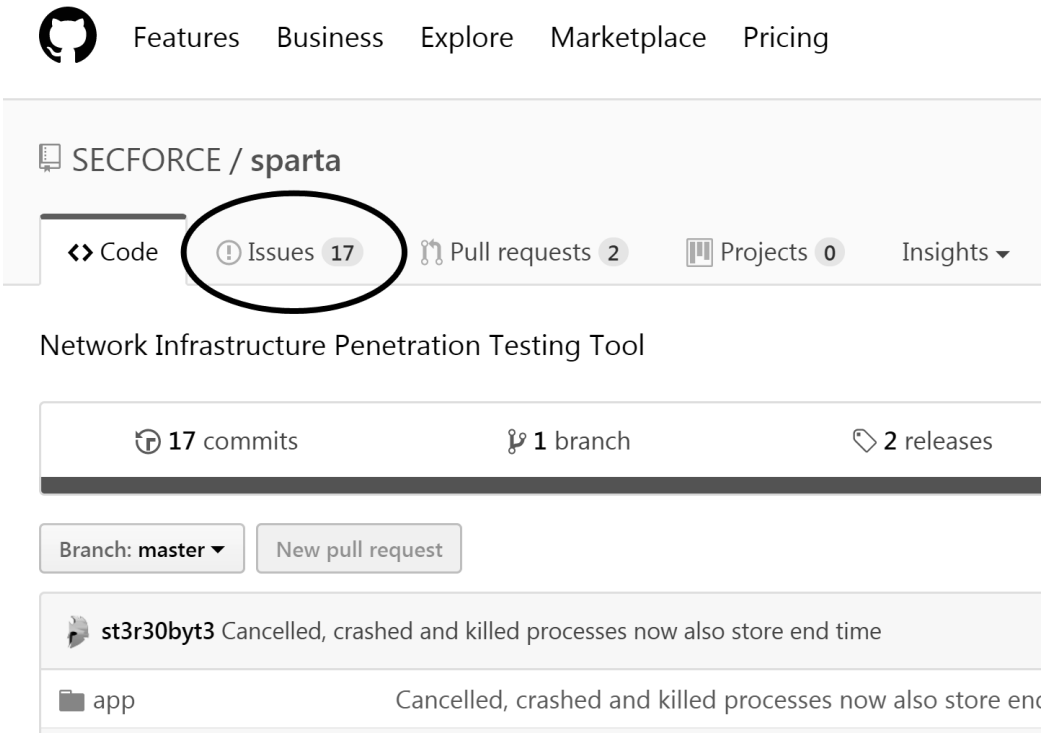


Figure 6.5 *Main page of a GitHub project*

You can then browse (and search) the list of open issues. Once you are confident that your bug is not yet filed, you can click on the New issue button (Figure 6.6, "Issues page of a GitHub project" [page 145]).

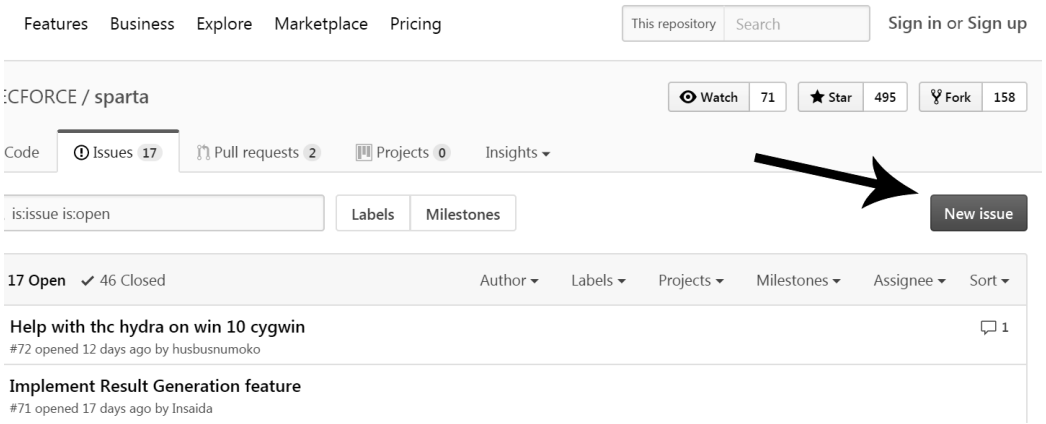


Figure 6.6 *Issues page of a GitHub project*

You are now on a page where you must describe your problem (Figure 6.7, "GitHub form to file a new issue" [page 145]). Although there is no template like the one found in `reportbug`, the bug reporting mechanism is fairly straight-forward, allowing you to attach files, apply formatting to text, and much more. Of course, for best results, be sure to follow our guidelines for creating a detailed and well-described report.

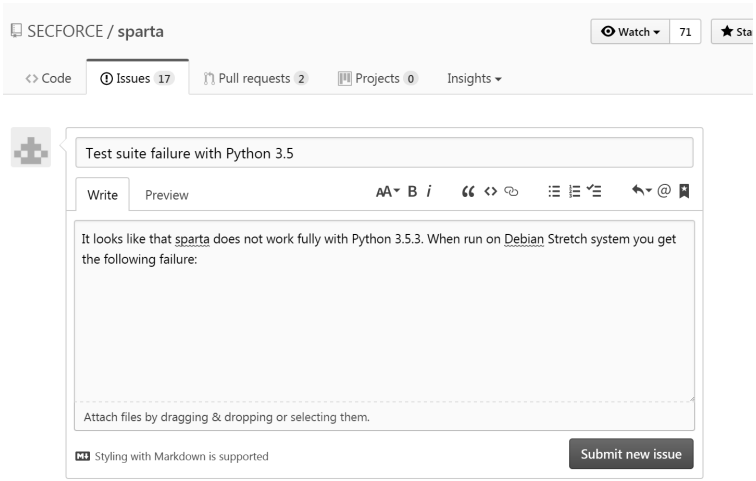


Figure 6.7 *GitHub form to file a new issue*

6.4. Summary

In this section, we discussed various methods to help you find documentation and information about programs and how to find help with problems you may encounter. We took a look at man and info pages and the `apropos` and `info` commands. We discussed bug trackers, provided some tips on how to search for and submit good bug reports, and provided some tips to help you figure out who owns the program or project in question.

Summary Tips:

- Before you can understand what is really going on when there is a problem, you need to know the theoretical role played by each program involved in the problem. One of the best ways to do this is to review the program's documentation.
- To view a manual page, simply type `man` *`manual-page`*, filling in the name of the command after an optional section number.
- The `apropos` command returns a list of manual pages whose summary mentions the requested keywords, along with the one-line summary from the manual page.
- The GNU project has written manuals for most of its programs in the *info* format. This is why many manual pages refer to corresponding *info* documentation.
- Each package includes its own documentation and even the least documented programs generally have a `README` file containing some interesting and/or important information. This documentation is installed in the `/usr/share/doc/`*`package`*`/` directory.
- In most cases, the FAQ or mailing list archives of a program's official website may address problems that you have encountered.
- The Kali project maintains a collection of useful documentation at `http://docs.kali.org`.
- The Kali Linux project uses the #kali-linux channel on the Freenode[5] IRC network. You can use chat.freenode.net as IRC server, on port 6667 for a TLS-encrypted connection or port 6666 for a clear-text connection. To join the discussions on IRC, you have to use an IRC client such as `hexchat` (in graphical mode) or `irssi` (in console mode). There is also a web-based client available on webchat.freenode.net[6].
- The official community forums for the Kali Linux project are located at forums.kali.org[7].
- If you uncover a bug in a program, you can search bug reports or file your own. Be sure to follow the guidelines that we have outlined to ensure your report is clear, comprehensive, and improves the chances that the bug will be addressed by the developers in a timely fashion.

[5] `https://www.freenode.net`
[6] `https://webchat.freenode.net`
[7] `https://forums.kali.org`

- Some bug reports should be filed to Kali, while others may be filed on the Debian side. A command like **`dpkg -s package-name | grep ^Version:`** will reveal the version number and will be tagged as "kali" if it is a Kali-modified package.
- Identifying an upstream project and finding where to file the bug report is usually easy. Simply browse the upstream website that is referenced in the Homepage field of the packaging meta-data.
- Kali uses a web-based bug tracker at `https://bugs.kali.org` where you can consult all the bug reports anonymously, but if you would like to comment or file a new bug report, you will need to register an account.
- Debian uses a (mostly) email-based bug tracking system known as Debbugs. To open a new bug report, you can send an email (with a special syntax) to submit@bugs.debian.org or you can use the `reportbug` command, which will guide you through the process.
- While many projects are hosted on GitHub and use GitHub Issues to track their bugs, there are also many others hosting their own trackers. You may have to research the basics of third-party bug trackers if you need to post to them.

Now that you have the basic tools for navigating Linux, installing and configuring Kali, and troubleshooting your system and getting help, it is time to look at locking down Kali so that you can protect your installation as well as your client's data.

Keywords

Security policy
Firewall
iptables
Monitoring
Logging