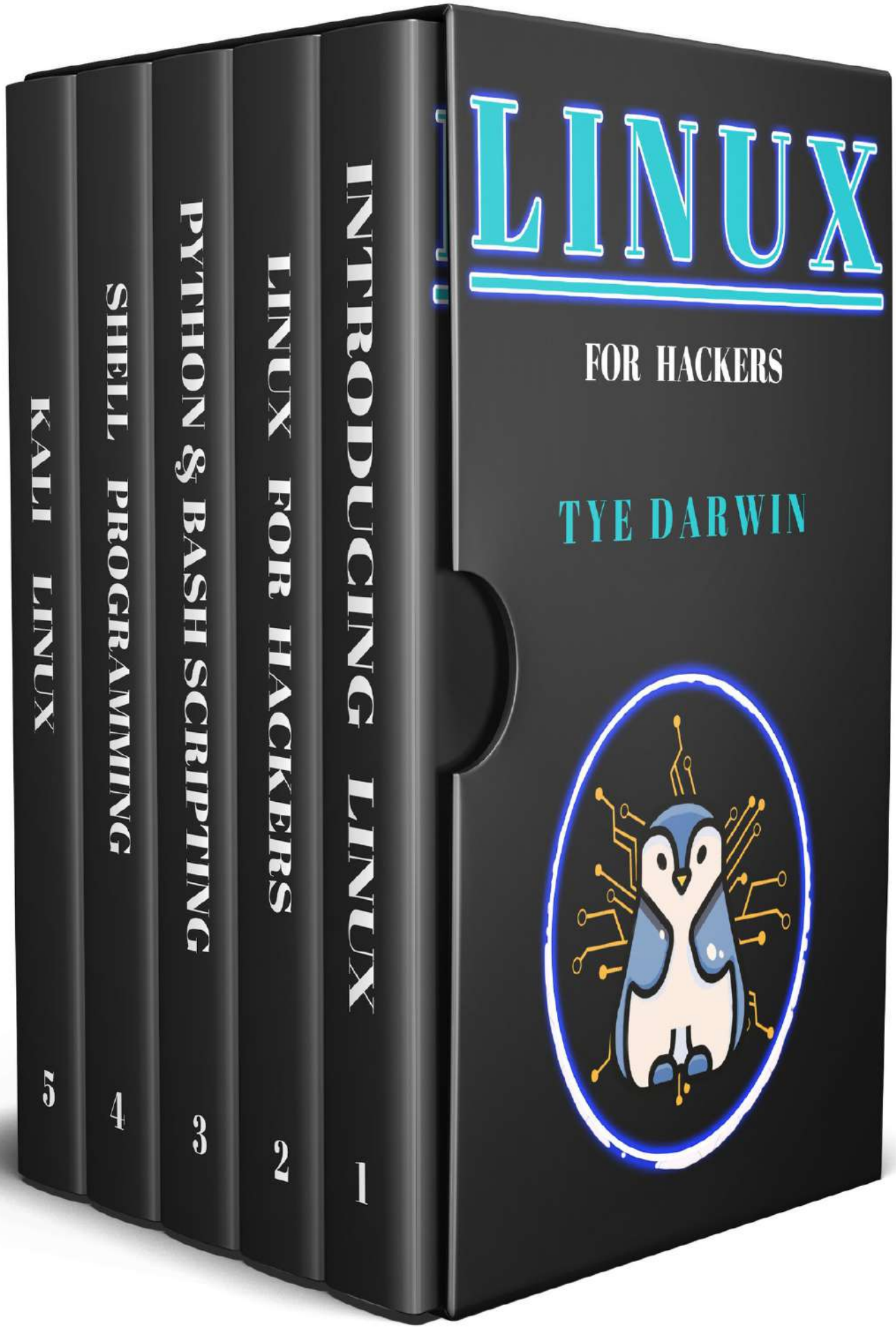

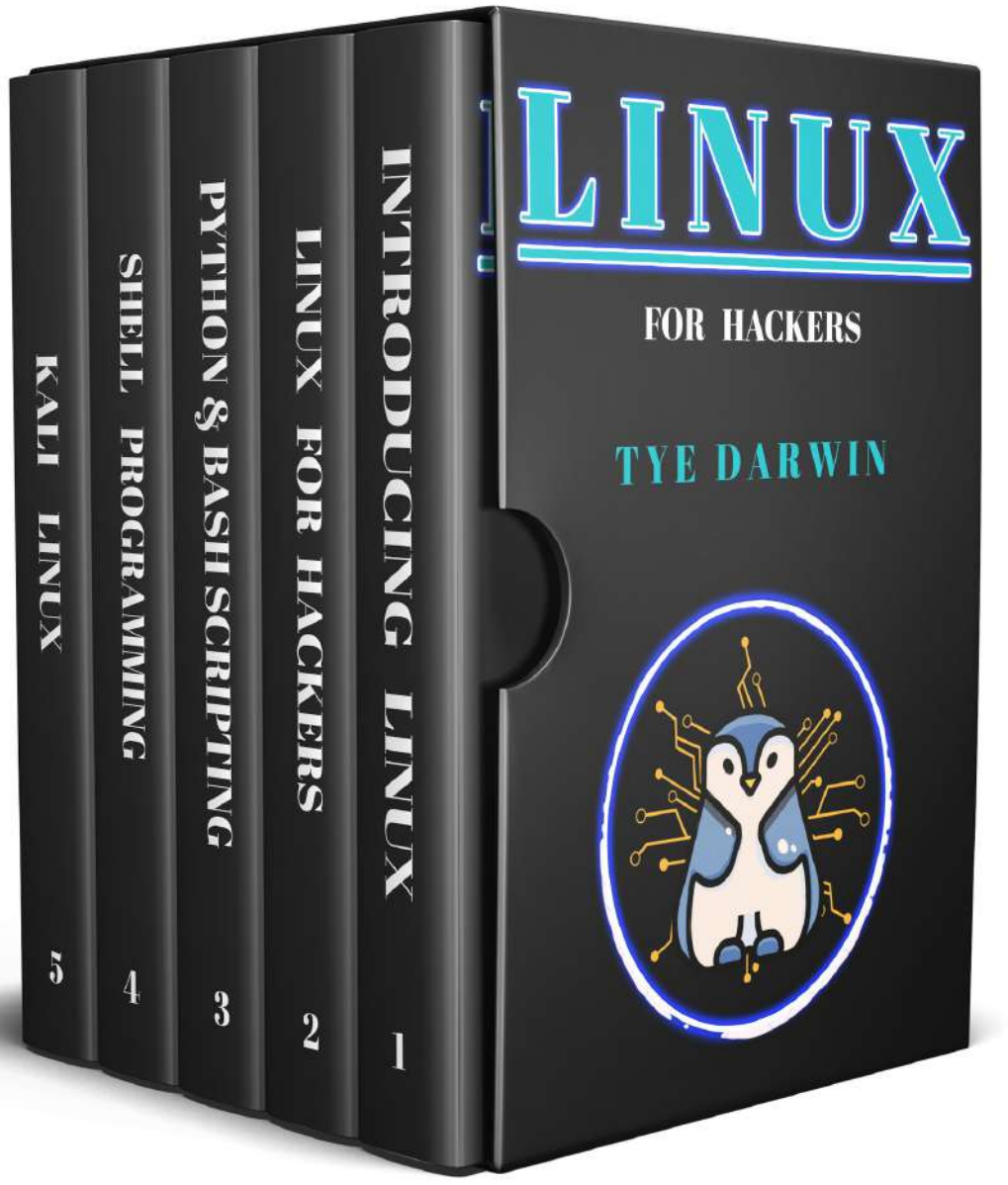

PRAISE FOR TYE DARWIN

This is one of the most important hacking books that came after a long time. A must read for all beginners looking toward cybersecurity as a career pathway

STEVE , SENIOR PENETRATION TESTER

Hacking Essentials Series has been one of the favorite sequential hacking books I have read in my lifetime

ANONYMOUS HACKER REVIEW

TYE DARWIN made me experiment with Kali Linux and helped me raise interest in Shell programming like no one ever did

AMAZON REVIEW

LINUX FOR HACKERS

LEARN LINUX BASICS AND BASH,SHELL, PYTHON SCRIPTING FOR HACKING USING KALI LINUX

TYE DARWIN

Edited by

DYE GUIND

GVS PUBLICATIONS

ALSO BY TYE DARWIN

Hacking for Beginners

For all the hackers that changed the world by destroying and To all the programmers who built all over again with much more safety

To be a hacker you first need to be a programmer. To be a programmer you need not be a hacker, but developing hacking skill can boost your confidence exponentially

ANONYMOUS QUOTE

CONTENTS

UNDERSTANDING LINUX AND INSTALLING YOUR FAVORITE LINUX DISTRO

INTRODUCTION

IF YOU ARE a beginner you might be probably wondering what Linux is? Let me help you with the basic stuff. Linux is an operating system just like windows and MacOS but with more security and stability and customizability. And you know what? It is completely free and is open-source (that means source code is not hidden or encrypted).

Seems awesome right? But you might wonder why Linux is less popular among normal users. Linux is an operating system that is specifically developed by professionals for professionals. It offers very less creative software when compared to Windows and macOS. You can't find a photoshop version for Linux because Adobe thinks that Linux is not a platform that developers use to create their products.

Then, you may wonder who uses Linux as a daily work machine?

Linux may not be a creative operating system but is definitely a preferred choice by developers , programmers, ethical and unethical hackers, database system managers and system administrators. Linux is complex and definitely has a learning curve. Linux enthusiasts should be aware of commands and particularly an in-depth understanding of shell programming to achieve what they want using the Linux kernel.

Are you overwhelmed by the possibilities of Linux? Don't worry because we are here to help you introduce Linux with simple explanations and instructions. Welcome to the first module of this book where you will get a simple introduction to the Linux ecosystem and a detailed step-by-step installation instructions. Let us go!

CHAPTER ONE
INTRODUCTION TO LINUX

THIS CHAPTER mainly describes some methods and experiences of learning Linux system, including how to choose a Linux distribution that suits you, as well as the relationship and applicable environment of each Linux distribution. Then it summarizes the general roadmap for learning Linux. It will also give sufficient information about building your own Linux learning environment, and finally will talk about some resources for learning Linux. By the study of this chapter, beginners will no longer feel confused in the process of learning Linux, and can find a learning method that suits them. Initially we will talk about some of the applications of Linux in everyday life.

CURRENT STATUS AND TRENDS OF LINUX IN VARIOUS FIELDS

Many novices have a very confusing question:

> *"I have heard of Linux, but after learning the Linux system, what can you do on it? Or what can the Linux system specifically do?"*

With this question, the book begins with an overview of Linux and its relationship with open source software. We will also talk about application fields and future development trends of Linux.

LINUX AND OPEN SOURCE SOFTWARE

Linux is a free and open source UNIX-like operating system. The kernel of this operating system was first released by Linus Torvalds in 1991. After that, with

the addition of user space applications, it became the Linux operating system.

Strictly speaking, Linux is just the operating system kernel itself. The term "Linux kernel" is usually used to express this. Linux is often used to refer to a complete operating system based on the Linux kernel, which includes GUI components and many other utilities.

The GNU General Public License (most commonly called as GNU GPL or GPL) is a widely used free software license agreement. Originally written by Richard Stallman for the GNU project, the GPL gives the definition of free software for computer programs. Any product derived from GPL software must be released under the GPL license and the source code must be disclosed.

Linux is the most famous example of the development of free software and open source software. As long as they follow the GNU GPL, any individual and organization can freely use all the underlying source code of Linux, and can also freely modify and redistribute it. With the rapid development of the Linux operating system, various integrated open source software and utility tools on Linux have also been applied and popularized. Therefore, Linux has also become synonymous with open source software. Red Hat Linux is one of the famous Linux operating systems that commercialized its tools to be sold for enterprises looking out a database administration system.

THE DEVELOPMENT OF LINUX IN THE SERVER FIELD

With the increasing influence of open source software worldwide, the Linux server operating systems has occupied more and more market shares in the server operating system market structure and has conventionally formed a situation of large-scale market applications, and has maintained a rapid Growth rates, especially in key national areas such as government, finance, agriculture, transportation, and telecommunications. In addition, considering the rapid growth of Linux and the support of relevant national policies in different countries, Linux server products are bound to impact the larger server market.

According to statistics from authoritative departments, Linux currently has a 75% market share in the server field. At the same time, the rapid rise of Linux in the server market has attracted great attention from the global IT industry and has become the backbone of the server operating system field with a strong momentum.

DEVELOPMENT OF LINUX IN THE DESKTOP FIELD

In recent years, especially in the technological market, the development trend of Linux desktop operating system is very rapid. The Linux desktop operating system launched by local system software vendors such as the winning Red Flag Linux, Deepin Linux, etc., has been widely used in the government, enterprises, OEM and other fields.

In addition, SUSE and Ubuntu have also launched Linux-based desktop systems, especially Ubuntu Linux, which has accumulated a large number of community users. However, from the perspective of the overall functions and performance of the system, there is still a certain gap between the Linux desktop system and the Windows series, mainly in terms of system ease of use, system management, software and hardware compatibility, and software richness. macOS is also popular among designers, creative professional in films and people who write books. Linux has a purpose and serve it well. So, does the other operating systems. As a hacker it is mandatory to understand the importance of Linux as it maybe hard to crack any system using windows and Mac software.

In the next section, we will talk about the development of Linux in various fields using a set of examples. Follow along!

THE DEVELOPMENT OF LINUX IN THE MOBILE EMBEDDED FIELD

Linux's low cost, powerful customization features, and good portability have made Linux widely used in embedded systems. At present, Linux has been widely being used in mobile phones, tablet computers, routers, TVs, and electronic game consoles. The Android operating system widely used on mobile devices is created on top of the Linux kernel. At present, Android has become the world's most popular smartphone operating system crossing the much hyped iOS operating system. According to the latest statistics from authoritative departments in 2015, the global market share of Android operating system has reached 84.6%.

In addition, Cisco has also used customized Linux in network firewalls and routers, and AWS Cloud has also developed a Linux-based operating system which can be used on smart phones, tablets and Internet TVs. Common digital video recorders, Stage lighting control systems are gradually adopting customized versions of Linux to achieve, and all this is due to the power of

Linux and open source. Linux is no where near to get extinct. It will spread readily across every corner in the world.

DEVELOPMENT OF LINUX IN THE FIELD OF CLOUD COMPUTING/BIG DATA

The rapid development of the Internet industry has promoted the formation and rapid development of cloud computing and big data industries. As a platform based on open source software, Linux occupies the core advantage of cloud computing and big data.

According to the Linux Foundation's research, 86% of enterprises have used the Linux operating system to build cloud computing and big data platforms. At present, Linux has begun to replace UNIX as the most popular cloud computing and big data platform operating system.

In the next chapter, we will provide details about different Linux distros to help readers chose the operating system of their choice. Choosing a Linux distro is always an overwhelming challenge. To know what is best for you we suggest to experiment with different distros to find what you really like.

There are so many Linux distributions that no one can give an accurate figure, but one thing is certain, Linux is becoming more and more popular. Faced with so many Linux distributions, beginners who plan to switch from other systems to Linux systems may be confused. Even loyal Linux users do not have the time and energy to try them one by one. Therefore, before beginners learn Linux, they need to have a clear direction. It is very important to choose a Linux system that suits them. The following next chapter is a classified introduction of Linux distributions.

CHAPTER TWO
CHOSING A LINUX DISTRO

THE PREVIOUS CHAPTER has been a good introduction to the importance and philosophy of Linux as an operating system. There are many Linux distros at present and it can be an overwhelming task to select that one perfect distro. Even professionals struggle to maintain or use one Distro consistently. As far was we say, using a Linux distro that perfectly supports your preferences is best way. There may be popular ones but they will not please you until you know what you want.

Here are some common Linux distributions:

- Red Hat Linux

Red Hat Linux is the earliest personal version of Linux released by Red Hat. Its 1.0 version was released on November 3, 1994. Although its history is not as long as other famous Linux distributions, it has a much longer history than many newer Linux distributions.

Since the release of Red Hat 9.0, Red Hat no longer develops desktop Linux distribution kits. Red Hat Linux has stopped development and concentrated all its efforts on the development of the server version, which is the Red Hat Enterprise Linux version. On April 30, 2004, Red Hat officially stopped supporting Red Hat 9.0 version, which marked the official end of desktop Red Hat Linux. The original desktop Red Hat Linux distribution was merged with Fedora from the open source community to become the Fedora Core distribution.

Red Hat is currently divided into two series: Red Hat Enterprise Linux, which is provided by Red Hat with paid technical support and updates, and the free Fedora Core developed by the community.

For who do we suggest this?

To all enterprises with a need for server administration, there is no distro that is more better than Red Hat Linux.

- Fedora Core

Fedora Core (FC) is positioned by Red Hat as a testing platform for new technologies, and many new technologies will be tested in FC. If these new technologies are stable, Red Hat will consider these software joining Red Hat Enterprise Linux.

Fedora Core 1 was released in late 2003, and FC is positioned for a desktop user. FC provides the latest software package, and its version update cycle is also very short, only 6 months. Due to frequent version updates, performance and stability cannot be guaranteed and it is generally not recommended to use Fedora Core on the server.

For users, Fedora is a free operating system with complete functions and rapid updates. Therefore, for personal applications, such as development and to experience new features, you can choose this release version.

- Kali Linux

As a hacker you need to be aware of the perfect Linux distro that exists for hackers. Previously, it is called as Blacktrack and is maintained by a number of volunteers. What Kali Linux excels is to provide tools that are essential for both hackers and forensic specialists in the same operating system.

While now a days it is true that Kali is being given competition with Parrot Linux, another famous Linux distro we still recommend beginners to use Kali Linux to begin their hacking career. Kali Linux is also free to download and fully supports all open-source policies. If you need to be serious with hacking then it is obvious that you take advantage by using Kali as your main operating system.

- CentOS

The full name of CentOS is "Community Enterprise Operating System". It is

compiled from the source code released by RHEL in accordance with the open source regulations. Since RHEL is a commercial product, all Red Hat logos must be changed to their own CentOS logo, which gave rise to the CentOS operating system.

The difference between the two is that CentOS does not include closed source software. Therefore, CentOS can not only be used freely, but also can enjoy the long-term free upgrade and update service provided by CentOS. This is a great advantage of CentOS.

CentOS is built from the source code package of RHEL. Its version number has two parts: a major version and a minor version. The major and minor version numbers correspond to the major and update packages of RHEL respectively. For example, CentOS 6.5 is built on the 5th version of the RHEL 6.0 update.

However, since the RHEL7.0 version, the CentOS version naming format has changed slightly. The major version still corresponds to the major version of RHEL, and the minor version is subject to the release date of the RHEL update package.

In 2014, CentOS announced its cooperation with Red Hat, but CentOS will continue to operate under the new committee and will not be affected by RHEL. This strategy indicates that the follow-up development of CentOS will be strongly supported by Red Hat.

- SuSE Linux

SUSE is the most famous Linux distribution in Germany and enjoys a high reputation, but its fate is quite rough now. On November 4, 2003, Novell announced the acquisition of SUSE. In January 2004, the acquisition was successfully completed, and Novell also officially named SUSE as SUSE Linux. Novell's acquisition of SUSE accelerated the development of SUSE Linux and changed the free SUSE Linux into an openSUSE community project.

But in 2010, Attachmate acquired Novell. After being acquired, the development of SUSE Linux was blocked. And just 3 years later, SUSE changed ownership again. In September 2014, Attachmate was acquired by the listed company Micro Focus. Fortunately, SUSE officially announced that open source is the foundation of SUSE development and will continue to contribute to open source. SUSE still Will fully support openSUSE.

Although SUSE has changed ownership many times, it does not affect its professionalism. According to incomplete statistics, SUSE Linux now occupies nearly 80% of the European Linux market, and most key applications are built under SUSE Linux.

- Ubuntu Linux

Ubuntu is a Linux operating system based on desktop applications. Based on Debian GNU/Linux, Ubuntu aims to provide general users with a main and also latest and stable operating system built by free software. Ubuntu has a huge community power, users can easily get help from the community.

Using this Ubuntu, Linux mint is also based upon which we will be mostly using in this book for discussing about various concepts.

The above section mainly introduced the most common Linux distributions. In fact, there are many Linux distributions. The more common ones are Debian GNU/Linux, Mandriva, Gentoo, Slackware, Knoppix, MEPIS and Xandros, as well as Red Flag Linux, Deepin Linux and Kylin Linux, etc. won't be introduced here. In fact, looking at the various distributions of Linux, Linux distributions are nothing more than the development of these two aspects, one is the server market, and the other is the desktop market.

Finally what are our recommendation according to the purpose?

Linux distributions represented by Ubuntu Linux take the desktop market route. Although they bring many surprises to users and are updated quickly, the development of the Linux desktop market is not optimistic because the desktop market has strong competitors like Windows. At present, Ubuntu Linux has also begun to exert its strength in the enterprise server market.

Linux distributions represented by the Red Hat series are now mainly for the enterprise-level Linux server market, focusing on the development of enterprise versions of Linux, and other distributions are focused on Linux Server market. The two major Linux distribution vendors have now taken the route of the Linux server market, which shows that Linux as an enterprise server has a huge development prospect. According to statistics from authoritative departments, Linux's share of the server market continues to rise every year.

In fact, many applications of Linux are aimed at Linux servers, and the description of this book is mainly aimed at various applications of Linux under

servers. As a hacker you need to understand how enterprises uses Linux to make their scenarios work. Even though cracking a target from a desktop system like Kali Linux , you need to be completely aware of the functionalities of a Linux server upon which the servers are built and maintained. We all know that all the sensitive and the information we absolutely need are stored in servers.

What is the first choice for beginners? - LinuxMint series

After learning about several major Linux distributions, we found out why we chose Linux Mint as a beginner's introductory study. Linux Mint now has a huge network of users, and 80% of network Linux resources are based on Ubuntu distributions. If you encounter any problems during the learning process, you can easily search for solutions on the Internet.

The Linux Mint series version can be easily obtained.

You can download the installation media of various versions of Mint from the official website of Linux Mint or 163 open source, SOHU open source, and Cloud open source sites. If this is the first time you are exposed to Linux, it is recommended to install Fedora Core first. Fedora Core is easy to install, has good hardware support, and has a gorgeous interface. You can also experience the latest features of Linux. If you have a certain understanding of Linux and need in-depth study, it is recommended to use the CentOS distribution system.

Linux Mint has a wide range of applications and is typical and representative.

Now basically all Internet companies' back-end servers use Linux Mint as the operating system. It can be said that after learning LinuxMint, it can not only quickly integrate into the enterprise's working environment, but can also be used by analogy, and other similar Linux distributions can also be quickly mastered. At the same time, the users who are learning Linux around are generally based on Mint, so that communication is convenient and problems in learning are easier to solve.

After getting a good grasp in Linux by using the Linux mint as a hacker you can expand your expertise by installing Kali Linux and can hack systems using tons of tools that are available in it.

What is Desktop platform first choice? -Ubuntu Linux

When it comes to the Linux desktop market, Ubuntu Linux occupies almost half of the desktop Linux. Ubuntu Linux is the most popular Linux desktop with

beautiful, simple and gorgeous interface. If you want to have entertainment and leisure under Linux, Ubuntu Linux is definitely the first choice.

Ubuntu installation is very user-friendly, just follow the prompts step by step. Ubuntu is known as one of the best and most comprehensive Linux distributions with hardware support. Many hardware that cannot be used on other distributions or in the default configuration can be easily installed and used on Ubuntu. Therefore, users can install Ubuntu as easily as Windows, and enjoy the fun of Ubuntu Linux.

What is the first choice for enterprise applications?-RHEL/CentOS series

Enterprise-level applications pursue reliability and stability, which requires high reliability and high stability on the system platform for building enterprise-level applications. Enterprise Linux distributions can solve this problem.

There is not much difference between the two Linux distributions of RHEL and Centos. The difference is that RHEL is a commercial Linux distribution. If you want to use the RHEL version, you need to purchase commercial authorization and consulting services. Red Hat provides system technical support And provide free system upgrades.

At present, the Red Hat official website no longer provides free downloadable CD media. If you need a trial, you can download an evaluation version of Linux with a trial time limit through the official website. And CentOS is a non-commercial release. You can download the installation media for each version of CentOS for free from the Internet, but CentOS does not provide commercial support. Of course, users do not have to bear any commercial responsibility.

So, should I choose CentOS or RHEL? It depends on whether your company has the corresponding technical strength. If it is a pure business enterprise, then it is recommended to purchase the RHEL distribution and purchase the corresponding services, which can save the enterprise's IT management costs and receive professional technical support services. On the contrary, if the enterprise technology is relatively strong, and has many years of Linux experience, then the CentOS distribution will be the best choice.

What is the first choice for hackers and penetration testers ? Kali Linux

Kali Linux is the first operating system that expanded the horizons for helping hackers to maintain a reliable distro that consists of all kinds of tools for various purposes. Kali Linux also provides TOR binding and will protect your

anonymity better than other software that are available. You can also try parrot Linux but it has more bloat then the Kali Linux and is hence only recommended for systems with huge memory resources.

WHAT NEXT?

With this, we completed a brief introduction to various Linux distros and provided tons of examples to explain which is the best operating system for each individual purpose. In the next chapter we will talk about some tips that can help you to become a proficient Linux expert.

CHAPTER THREE
STRATEGY TO MASTER LINUX

IN THE PREVIOUS chapter we have talked about various Linux distributions. Before heading to the installation procedure we need to provide you a few tips so that you will master Linux in the way that worked for a lot of other professionals before. However, remember that this is not a strict roadmap to follow. You can experiment and find out which strategy is working for you in the long run. This is only a strategy for your better understanding about obtaining the technical skills that needs a practical use case in enterprise industry.

HOW TO DEVELOP GOOD LINUX OPERATING HABITS?

After learning Linux, please do not think about the problem in the way of working in Windows, because there are indeed very big differences between them. For example, the memory management mechanism and process operation mechanism between them are very different. These similarities and differences Points will be described in the following chapters. Therefore, it is very important for beginners to put aside the kind of thinking of Windows and try to tap the unique potential of Linux with a new concept. Always make sure that you are thorough with all the Linux basics such as File management, process management and Logging management.

BE SURE TO GET USED TO THE COMMAND LINE MODE

Linux is an operating system composed of command lines. The essence and heart of the operating system is in the command line. No matter what level the graphical interface develops, the operation of the command line mode will never change. Linux commands have many powerful functions: from simple disk

operations, file access, to the production of complex multimedia images and streaming media files, they are inseparable from the command line. Although Linux also has a desktop system, X-Window is just an application running in command line mode.

Therefore, it can be said that commands are the basis for learning Linux systems. To a large extent, learning Linux means learning commands. Many Linux masters are actually commanders.

Perhaps it is too difficult for a beginner who has just switched from a Windows system to a Linux system to enter the boring command learning immediately, but once you learn it, you will love it, because its functions are so powerful.

COMBINATION OF THEORY AND PRACTICE

Many beginners will encounter such a problem. They are familiar with every command of the system, but when the system fails, they can't start, and they don't even know when to use which command to check the system. This is the most helpless thing for many Linux novices. In the final analysis, the theoretical knowledge learned is not well integrated with the actual operation of the system.

A lot of Linux knowledge, such as the meaning of the parameters of each command, is very clear in the book, and it seems to be easy to understand, but once combined and used, it is not so easy. Without multiple hands-on practice, the skills are impossible to Completely master.

The human brain is not like a computer's hard drive. Unless the hard drive is broken or the hard drive is formatted, the stored data will always be stored in the hard drive and can be recalled at all times. In the curve of human memory, one must keep repeating practice to remember one thing more firmly. The same is true for learning Linux. If you can’t keep learning, you will learn the latter and forget the former. There are also some Linux beginners who have learned a lot of Linux knowledge, but because they have not used it for a long time, the things they have learned are forgotten in a short period of time. Over time, they lose their confidence in learning.

It can be seen that to develop your own combat skills, you only need to be diligent and willing to practice. This is also the foundation of learning Linux well.

LEARN TO USE LINUX ONLINE HELP

The technical support time of each Linux distribution is relatively short, which is often not enough for Linux beginners. In fact, when a complete Linux system is installed, it already contains a powerful help interface, but maybe you haven't found it yet, or haven't mastered the skills to use it. For example, if you are not very familiar with the use of the tar command, just enter "man tar" on the command line, and you will get a detailed description and usage of tar.

The mainstream Linux distributions come with very detailed help documents, including instructions and FAQs, from system installation to system maintenance, to system security, and detailed documentation for users at different levels. After reading the document carefully, 60% of the problems can be solved.

LEARN TO THINK INDEPENDENTLY AND SOLVE PROBLEMS INDEPENDENTLY

When encountering a problem, the first thing that comes to mind is how to solve the problem by yourself. There are many ways to solve it. For example, through these methods, 90% of the problems can be solved by reading books, searching information, searching online and browsing technical forums.

Thinking and solving problems independently not only exercises one's own ability to solve problems independently, but also improves rapidly in technology. If you can't solve the problem by the above methods, you can ask people, think about why you did it after you get the answer, and then take notes to record the resolution process. The most taboo way is to ask people whenever you encounter a problem. Although this may solve the problem quickly, in the long run, you will rely on others when encountering problems, and you will not improve in your technical knowledge.

LEARNING PROFESSIONAL ENGLISH

If you want to learn Linux in depth, you must try to view the English documentation. Because the best and most comprehensive documents of technical things are all in English, and the first high-tech released are all written in English. Even if people in non-English speaking countries publish technical documents, they are first translated into English and published in international

academic journals and the Internet.

When installing a new software, first read the Readme document, then the Install document, then the FAQ document, and finally start the installation, so that you will know the cause if you encounter a problem. Therefore, it is necessary to learn a little professional English if you are from other language speaking countries.

In the next section, we will provide a Linux roadmap for your better analysis of the subject. Follow along!

LINUX LEARNING ROADMAP

Linux operation and maintenance or management talents are one of the technical talents urgently needed by enterprises. Based on many years of work experience, the author has summed up a set of strategies for learning Linux. Readers can use this as a basis to grasp the key points and distinguish the priority. I believe that it will achieve a multiplier effect with half the effort. If readers can earnestly study and master the technical points involved in this roadmap, it will basically meet the basic application needs of enterprises for Linux operation and maintenance or management talents.

This strategy is divided into three stages: elementary, intermediate and advanced. The initial stage is mainly an introduction to the basic knowledge of Linux and basic applications of the system, and there are more contents to be mastered. If you are new to Linux, it is still difficult to get started.

Linux focuses on command operations, so the primary stage is to learn basic commands. Reading more books and practicing more is the foundation of learning commands well. The intermediate stage focuses on common server configurations, involving various application server configurations, network configurations, and system security configurations.

The difficulty at this stage lies in the construction of various servers, which requires high comprehensive knowledge. The focus of the advanced stage is programming language and cluster architecture. The development direction at this stage is senior operation and maintenance engineer or system architect. To become a system architect, it is necessary to be proficient in a programming language, and the common cluster architecture and distributed architecture under Linux are also required by senior operation and maintenance engineers.

WHAT NEXT?

With this, we have provided a brief introduction to start learning Linux by using different strategies that are usually followed by professionals. With this essential prerequisite we further move along by learning about the installation procedure of Linux in the next chapter. Follow along and make you understand Linux in a better way.

CHAPTER FOUR

LINUX INSTALLATION PREREQUISTES

IN THE PREVIOUS chapter of this book we have introduced Linux and provided some real life scenarios where Linux is being used by professionals and developers. If you are still not satisfied to opt out Linux as your operating system we suggest you to visit “www.distrowatch.com” to look at the famous Linux distros and their awesome features.

If you are proceeding further then we are believing that you have chosen your preferred Linux distro for experimenting with Linux. Is it Manjaro? Is it Linux Mint? Is it Debian based Linux distro? Or are you a tech savvy who had chosen Arch Linux? No matter what Linux system you are willing to use this book will help you get the basic philosophy that is important to master Linux. We don’t spoon feed you with clear instructions but will provide concrete instructions in a way that you can do work by yourself.

These next couple chapters in this module will help you to install Linux in your computer. We provided a couple of methods to install Linux. Follow along and be ready to hit a couple of google searches to achieve what you want. Amigos let’s go.

INSTALLING LINUX

Before learning the various operations of Linux, you must first install the Linux system. Compared with the installation of the Windows system, the installation of the Linux system has many points to pay attention to, such as choosing the appropriate installation method and determining the naming scheme you want to proceed with. This chapter will take the Linux Mint latest distro version as an example to explain the installation process of the Linux system in detail and help

solve the problems that may be encountered during the installation.

Warning:

Always make sure that you take a backup before proceeding with the installation procedure. If you are using this machine for work purposes we recommend to take backup of two copies at least for any problems that may arise during installation.

INSTALLATION REQUIREMENTS

Generally, each Linux distribution will give a list of minimum requirements and recommended configurations for the system, and different installation options (such as graphical interface or character interface) have different requirements for the system. While you are downloading the installation disk image from the official website of the Linux distro the maintainers of the website will usually display the minimum requirements. If not, please look out at the forums or do a quick google search for the details.

Linux has very low hardware requirements. Most of the machines that can run Windows can be used to install Linux, and the running speed will be much faster than Windows. The minimum hardware configuration for installing Linux is not discussed here, only some special applications and special installations are explained. This is due to the fact that we are not aware of the distro you are going to install. For this particular Linux Mint version 1GB is the minimum required RAM memory.

If you want to install a graphical interface (that is, X-Window in Linux), or run office software such as OpenOffice, the system's graphics card and memory requirements are higher, preferably a discrete graphics card, otherwise the display effect of the graphical interface will not be ideal . Linux is not particularly a well known operating system for gaming. If you are into gaming then I think that you should better stick with windows PC.

Most of the drivers on Linux are written by open source people based on the information provided by the hardware manufacturer, and some of them are difficult to write because the hardware manufacturer refuses to provide the information. In recent years, because Linux has become hot, many hardware manufacturers have changed their normals and actively assisted Linux developers to provide hardware information, but they are still conservative.

Although the Linux installation CD already contains most of the hardware drivers, it is inevitable that the Linux distribution cannot update the corresponding drivers in time due to the rapid hardware update.

If the hardware configuration is very new, you need to check whether the Linux installation version contains all the drivers for the hardware. The most common ones are network card drivers, sound card drivers, etc. Readers can check on the official website of the hardware manufacturer, which lists all the operating systems and versions supported by the hardware.

In the next chapter we will provide instructions to install Linux Mint using an optical drive. Follow along!

CHAPTER FIVE

INSTALLATION PROCEDURES FOR LINUX

LINUX SYSTEM INSTALLATION is diverse and flexible, and different installation methods can be selected according to different environments. Common installation methods include hard disk installation, U disk installation, network installation, and CD-ROM installation.

HARD DISK INSTALLATION METHOD

The hard disk installation method is generally based on the Windows system. For example, if you want to install a dual system where Windows and Linux coexist or if the system does not have an optical drive, you can use the hard disk method to install.

Attention:

Because the file system format of Windows is completely different from that of Linux, it is absolutely impossible to install both Windows and Linux in one partition. Even if you do this, Linux cannot recognize the hard disk partition. This type of installation is usually not recommended for normal users. If you are using a low end system but need a high memory based operating system then we suggest you to install Linux using Hard-disk installation.

U DISK INSTALLATION METHOD

Today's servers are generally not equipped with CD-ROM drives, and the installation speed of CD-ROM drives is slow, and a lot of time will be wasted when installing systems in batches. At this point, you can install the system

through a USB flash drive, which is a fast, cheap and efficient Linux installation method.

First of all, the U disk is inexpensive and can be used in many ways. Secondly, all servers or PCs are basically equipped with a USB interface, and U disk installation is universal. Finally, the installation speed of U disk is very fast, which can save a lot of time. Therefore, the U disk installation of Linux is a current development trend. The next chapters in this module will introduce in detail the method of U disk installation of Linux using Rufus and Usb writer software.

NETWORK INSTALLATION METHOD

When installing a single server or a small number of servers (no more than 10), the installation via U disk or CD-ROM can also be used, but if you want to install hundreds or even thousands of servers, this method is obviously unrealistic of. So how to quickly install it in the face of such a demand? Yes networks are the answer

Network installation of Linux is generally an automated installation process in large quantities. The most common batch network installation tool is Kickstart, which is an unattended Linux system automatic installation tool.

With Kickstart, system administrators can automatically complete the installation of Linux systems by creating an answer file. Its working principle is to generate a ks.cfg response file through various parameters that need manual intervention in the typical installation process.

The installer only needs to tell Kickstart where the installation program reads the ks.cfg file, and then Kickstart will automatically complete the installation of the system according to the settings of this file.

As the network installation needs the support of network services, it is difficult for beginners and therefore is not described too much here. The following section focuses on installing Linux through CD-ROM and U disk.

INSTALLATION USING CD-ROM

Installing Linux system by CD-ROM is the most common software installation method, which is simple and easy to understand. To be frank the focus of this

chapter is to install Linux by CD-ROM, provided that the computer has a CD-ROM and Linux system installation CD. You can easily create a Linux installation CD using during software such as Nero just like you do normal CD burning.

Readers can download the installation media in ISO format for Linux mint version from the Internet, and then burn the ISO file to CD. For example, Linux Mint version 17.1 can be downloaded from the official website.

After the download is completed, the ISO file is burned to the DVD disc through the burning software such as UltraISO, InfraRecorder and Nero Burning ROM, and then can be used to enter the BIOS to set the first startup sequence of the computer as DVD-ROM. After saving the settings, the DVD should be inserted into the optical drive and the computer will be restarted. The disc will automatically boot and start the Linux installation program.

In the next chapter, we will talk about other advanced stuff such as partitioning that are essential for installation of Linux. Let us go!

CHAPTER SIX
PARTITIONING IN LINUX

PARTITIONING IS A PRETTY basic procedure that is mandatory while performing an installation of Linux in a computer. Even if you install a windows system you need to partition the system. While windows provide a graphical interface the partition mechanism in Linux can be a bit different and some linux dusters doesn’t even offer Graphical installers such as Gparted. Follow along to know in detail about the partition scheming and installation procedure.

PARTITION NAMING SCHEME

Before starting the installation, we need to know some common sense about partition naming in Linux.

Under Linux, hard disk partitions are identified by a combination of letters and numbers, which is different from using "C disk" or "C:" to identify hard disk partitions under Windows system. This naming scheme of Linux is more flexible than that of Windows, and the meaning expressed is clearer. It is completely possible to know the hard disk partition situation in detail through partition identification.

At the same time, Linux's hard disk naming scheme is file-based, generally with the following file naming methods:

```
/dev/hda2/
dev/sdb3
```

The following details the specific meaning represented by each character in

these partition naming schemes.

/dev: This is the directory where all device files are stored.

hd and sd: They are the first two letters of the partition and represent the device type in which the partition is located, where hd represents the IDE hard disk and sd represents the SCSI hard disk.

a: It is the third letter of the partition name, indicating which device the partition is on.

For example,/dev/hda represents the first IDE hard disk,/dev/sdb represents the second SCSI hard disk,/dev/sdd represents the fourth SCSI hard disk, and so on.

2: This number represents partitions.

The first 4 partitions (main partition or extended partition) under Linux are represented by numbers 1 ~ 4, logical partitions start from 5, and so on. ~

For example,/dev/hda2 represents the second primary or extended partition of the first IDE hard disk,/dev/sdb3 represents the third primary or extended partition of the second SCSI hard disk, and/dev/sdc6 represents the second logical partition of the third SCSI hard disk.

DUAL-SYSTEM WINDOWS + LINUX HARD DISK PARTITION SCHEME

Many beginners who switch from Windows to Linux like to install dual systems on their computers. This is also called as dual booting in technical terms. This method is more dangerous for beginners of Linux and may cause the loss of hard disk data. Therefore, it is necessary to talk about how to safely and effectively install the dual system of Linux + Windows coexistence.

First of all, it should be understood that Linux and Windows are two completely different systems, so the file systems of Linux and Windows are also incompatible with each other. If you want to install Linux, you must "dedicate" a partition from the hard disk to Linux, and this dedication is not to empty the data under a certain drive letter, but to completely delete the disk from Windows.

For example, if I want to give D disk space to Linux system, I need to completely delete D disk in disk management under Windows. The size of the space depends on the size of the installation package. Generally, 10GB of space can meet the requirements.

After sorting out the space required by Linux, the next job is to restart the system, set the BIOS to start from the optical drive, put the Linux CD into the optical drive, and then enter the system installation interface.

note:

In a Dual system installation it is best to install Windows system first, and then install Linux, because every time Windows is installed, the system boot files will be re-modified. If you install dual systems in the reverse order, Linux may not boot, and you may need to repair the system boot files.

Also while installing GRUB boot loader will be replaced from windows to Linux. And in future instances if for some reason you chose to delete Linux then the Grub will be broken and you many not boot into your windows system.

This is one of the most well known problem that beginners encounter while dealing with dual-booting systems. To troubleshoot boot loader problems we suggest to check forums of your manufacturer.

In the next chapter, we will start the installation procedure of Linux Mint. Follow along!

CHAPTER SEVEN

STARTING INSTALLATION PROCEDURE

AFTER DETAIL INSTRUCTIONS provided by previous chapters, we already have sufficient preparation knowledge before installing the Linux system. Now start to install the Linux system using the specific steps as follows.

HOW TO MAKE SYSTEM BOOT?

1) In the BIOS of the motherboard, set it to boot from DVD-ROM, put the CD into the optical drive, and the interface to chose options will appear.

After the installation program successfully boots from the CD, a boot menu with several options will be displayed. If no key is pressed within 60 seconds, the default boot option will be run. To select the default option, you can wait for the counter to time out or press Enter on the keyboard.

To select options other than the default options, use the arrow keys on the keyboard and press Enter after selecting the correct option.

If you are installing from the Linux mint iso then the installation menu will appear. The Linux Mint installer has three startup options to choose from, namely Install Linux-mint , Test this media & install Linux Mint and Troubleshooting (Failure repair menu).

Almost all other Linux distros also provide a similar menu in their installation media file. Security researchers or forensic specialists who are not willing to Select Install CentOS 7 to automatically enter the installation process of the graphical interface. If you want to test whether there is a problem with the installation media, you can choose Test this media & install Linux Mint.

This option is the default option, it will first check the integrity of the installation media, and then Start the installation program and automatically will enter the graphical interface installation process. The Troubleshooting option is mainly used to help solve various installation problems and to repair system faults.

2) Select Troubleshooting and press Enter to get the interface that starts the installation procedure.

In the Troubleshooting interface, there are usually 4 options, which respectively represent the installation of Linux Mint in the basic graphics mode, the rescue of Linux Mint mode, the memory test, and the boot from the local drive option. Among these 4 options, the most used one is the rescue mode, which can be used to repair the system's failure due to system kernel problems, configuration file errors, disk errors, etc.

There are two commonly used installation methods for Linux, namely characters and graphics. Since Linux Mint latest version, the character installation method is basically not recommended. The default way is the graphical installation method. However, the system installation program still retains the character installation interface. If you need to install the system in character mode, you can press Esc in the interface and then enter "Linux text" after "boot:" to enter the character installation interface.

Attention

The process of installing Linux in graphical mode and character mode is exactly the same. If the installed computer does not have a graphics card or the graphics card does not support graphics installation, you can choose character installation. Generally choose the graphical installation method.

3) In the interface shown, select Install Linux Mint, press Enter directly, select the graphical installation method, and then enter the interface. This step is used to select the language during system installation. Here, select the English option. Of course, you can select other language options, but it is recommended to choose English installation. You can realize the benefits of choosing English when the system is installed. After selecting the language, click the Continue button to enter the next installation interface.

4) This step is an overview of system installation. As can be seen from the interface, the installation process can be divided into three parts, followed by LOCALIZATION (localized installation), SOFTWARE (software installation)

and SYSTEM (system installation).

Localized installation mainly includes time and time zone settings, keyboard settings and language support settings. Software installation includes two parts: installation source and software installation. System installation includes three parts: disk partition, kernel crash dump, and network settings.

Click each part to enter the setting interface, which will be introduced in turn.

LOCALIZATION SETTINGS

a) Click the DATA & TIME part of the interface to select the system time zone and time. If you are in America, select North America for Region and New York for City. After the selection is complete, click the Done button on the upper left to return to the interface.

b) In the next interface shown, for the KEYBOARD option, just keep the default English (US). Then click the LANGUAGE SUPPORT option, select the language pack that needs to be installed, you can choose the language pack to install according to your own needs, here choose English and French for example. Two additional language packs will be downloaded. After the selection is complete, click the Done button on the upper left to return to the interface.

SOFTWARE INSTALLATION SETTINGS

a) Enter the installation source configuration section below. Since Linux has multiple installation methods, multiple installation sources can also be selected here. In the interface shown, click the INSTALLATION SOURCE option to enter the system installation source configuration interface. Because the system introduced here is to install the system through a CD-ROM drive, you can keep the default options. After the selection is completed, click the Done button on the upper left to return to the interface.

b) Enter the system installation software configuration section under interface, click the SOFTWARE SELECTION option to enter the software package installation selection interface.

The package selection interface of the Linux Mint version is very different from the 5.x/6.x version. In the 7.x version, the software packages are classified according to the various uses of Linux, mainly divided into 10 For each

application scenario, readers can choose the corresponding classification according to their needs.

There are certain rules for the selection of software packages. The general experience is as follows.

If you are new to Linux, it is recommended to choose GNOME Desktop or KDE Plasma Workspaces. These two environments provide a very friendly and intuitive Linux desktop environment, allowing beginners to quickly integrate into Linux learning.

If you want to develop programs on Linux, it is recommended to choose Development and Creative Workstation. This environment provides the software, hardware, graphics and other tools needed for development. If you only need a Linux environment, you can choose Minimal Install, which only installs some basic software necessary for the Linux system.

If you want to run a virtualization program on Linux, you can choose Virtualization Host. This environment contains the software and applications necessary to run the virtualization program.

If you want to build a Linux server, it is recommended to choose Server with GUI. This environment includes basic network service facilities and GUI desktop.

In this installation, we use Linux as a machine, so here we choose with GUI. After selecting the installation environment on the left, the plug-ins of the selected environment will appear on the right. These plug-ins can be selected or not.

For Linux servers, security is the first priority. Install whatever software packages are needed, and do not install those that are not needed. Excess software packages not only occupy disk space, but also bring potential security risks to the server. Therefore, when Linux is used as a server, it must follow the principle of installation on demand and no third party installation until verified. For example, to build a DNS server, you only need to install the DNS software package and a basic system kernel. This can maximize system efficiency and ensure system safety.

After the selection is complete, click the Done button on the upper left to return to the interface with all your settings.

WHAT NEXT?

In this chapter, we have provided a couple of ways to start the installation procedure in Linux. In the next chapter, we will talk about the system installation settings that are mandatory for any Linux distro. Follow along!

CHAPTER EIGHT

SYSTEM INSTALLATION SETTINGS

THE PREVIOUS CHAPTER has been a good start for linux enthusiasts to understand about different processes that will help to understand what Linux is. In this chapter we will talk about different system installation settings that are normally used in Linux installation procedure. Follow along!

SYSTEM INSTALLATION SETTINGS

1) Now enter the disk partitioning stage. In the interface shown, click INSTALLATION DESTINATION to enter the disk partitioning stage. You can see that there is a hard disk sda with a size of 100GB. The size shown will be according to your disk size.

Before partitioning, some necessary instructions for Linux partitioning.

The necessary partitions under the Linux system are the root partition (the root partition is marked with "/") and the swap (marked as swap) partition.The swap partition is equivalent to the concept of virtual memory in Windows, which is the exchange of memory data with the hard disk.Regarding the size of the swap partition, many sources point out that if it is at least twice the size of the physical memory, it is not.

A basic principle is: if the memory is small (according to experience, the physical memory is less than 4GB), generally set the swap partition size to twice the memory. If the physical memory is greater than 4GB but less than 16GB, you can set the swap partition size to be equal to the physical memory. If the memory size is above 16GB, you can set the swap partition to 0. The swap partition can be set to 0, but this is not recommended, because setting a certain size of swap

partition still has a certain effect.

Although Linux only needs to divide the root partition and the swap partition to complete the system installation by default, it is not recommended because if only the root partition is divided, the system may not be able to boot after the root partition is damaged and it is stored in the root partition. The data may also be lost, which is very insecure. Therefore, it is recommended to assign independent partitions to independent applications, so that even if a partition is destroyed, it will not affect the data of other partitions, and can minimize the loss caused by system crashes.

The following partitions are recommended to be allocated independently when installing the system.

- /boot: Store system boot information and kernel information.
- /usr: Storage system application software installation information.
- /var: Store system log information.

For the meaning of each partition, please refer to the next chapter for more detailed information.

The root partition contains all the directories of the Linux system. If only the root partition is allocated when the system is installed, the above /boot, /usr, and /var will all be included in the root partition, that is, these partitions will occupy the space of the root partition. If you divide /boot, /usr, etc. separately, these partitions will no longer occupy the space of the root partition.

After understanding some basic partitioning knowledge, start disk partitioning below.

2) In the interface shown, first select the 100GB sda disk, and then there will be two partition options in the lower left corner. The first is Automatically configure partitioning, which means automatic partitioning. The second is I will configure partitioning, which means manual partitioning.

If you are not familiar with partitioning, you can directly select automatic partitioning by default. However, for learning considerations, it is recommended to choose manual partitioning, even if you are a novice choosing manual partitioning is more helpful for understanding and understanding of system partitions. Select manual partition here. After selecting, click the Done button in the upper left corner to enter the interface shown.

3) After selecting manual partitioning, you also need to select the partition scheme. You can select the partition scheme from the drop-down menu in the left pane. The available partition schemes are Standard Partition, Btrfs, LVM and LVM Thin Provisioning (thin provisioning).

Among them, standard partitions can be used for various file systems or swap space (swap), which is a commonly used partitioning scheme. Btrfs is the next-generation Linux file system, which is used by companies and communities such as Oracle, Red Hat, Intel, and Suse.

LVM is a logical volume management partition scheme. When creating an LVM partition, an LVM logical volume is automatically generated, and the physical disk can be used flexibly through LVM and can improve disk performance. Using LVM thin provisioning, you can dynamically create and allocate storage pools, and then freely manage disk space. LVM thin provisioning is an upgraded version of LVM.

In fact, there is also a RAID partition scheme. When there are two or more disks, a RAID partition scheme will appear. This is a software RAID that can provide high-end storage functions and redundant data security for the server.

Choose the Standard Partition partition scheme here. There are two steps to add a partition. First, you need to create a mount point. Click the "+" button in the lower left corner to create a mount point.

4) Create a root partition first, and set the root partition space as large as possible, because if the root partition space is full, the system may stop responding, and the applications running on the system may become abnormal. Since partitions such as /boot and /usr have been separated, 20GB of space in the root partition is sufficient.

In fact, the partition size setting is related to the hard disk size. If the hard disk is large enough, you can set a larger root partition space. Ensuring that the system is sufficient without wasting space is the criterion for partitioning disks.

The Mount point (mount point) can be selected by clicking the drop-down list, or can be filled in by yourself. Desired Capacity (partition size) such as 20GB can be directly filled in "20G", 500MB can be filled in "500M", And so on. After the setting is complete, click the Add mount point button to complete the addition of the mount point.

5) After the mount point is successfully created, it will enter the interface. Click

the mount point in the left pane, and you can do various modifications in the right pane, for example, edit the mount point, Modify the partition size, select the device type, select the file system type, label options for customization, and whether to encrypt or reformat the corresponding partition.

For File System selection, the default is xfs, but there can be seven types of swap, vfat, BIOS boot, ext2, ext3, ext4, and xfs. Among them, ext2, ext3, and ext4 are the mainstream on Linux systems File system, in versions before RHEL/CentOS 7.x, the ext series file system is adopted by default. VFAT file system corresponds to the FAT file system on Windows and swap is the swap partition on Linux. BIOS boot is mainly used for system boot devices , Is a very small partition and the xfs file system is a high-performance file system that only appeared on RHEL/CentOS 7.x.

The Label option is a label corresponding to the disk partition, through which the partition can be quickly identified. The Label option can be filled in or not, and it is blank by default.

Once the mount point is established, the corresponding disk ID is fixed. For example, the disk ID corresponding to the root partition is /dev/sda1. This "sda1" cannot be modified unless the partition is deleted and re-established.

After all settings are modified, click the Update Settings button in the interface to complete the editing of the disk partition.

6) Click the "+" button in the lower left corner of the interface shown to continue to create the /boot mount point. After that select "/boot" from the Mount Point drop-down list, and in the Desired Capacity input box select the entered size is "500M".

The /boot partition does not need to be too large, because it is only used to store some system boot information and kernel information. Click the Add mount point button so that the /boot mount point is created.

7) Then create a mount point of swap type. Select swap in the Mount Point drop-down list, and then enter the size in the Desired Capacity input box, where 4096M is specified.

Note that this is "swap", not "/swap".

8) Then create the /usr and /var partitions separately. The creation method is basically the same as that of the /boot partition, but the partition size is different.

If more application software is installed, the /usr partition can be increased appropriately. The /var partition is recommended to be set larger, because after the system runs for a long time, more logs will be generated accordingly.

9) After creating the necessary partitions for the system, if there is enough space, you can also create a partition of your own to store your own data. Here, create a /mydata partition. Note that there is a "/" before each partition, because each partition is created under the root partition.

In order to ensure that all the disk space can be used by Linux, divide all the remaining space of the disk to the /mydata partition. The specific operation is to enter "/mydata" in the input box of Mount Point, and then leave the Desired Capacity input box blank, so that all the remaining disk space is allocated to the /mydata partition.

10) So far, the partition work has been basically completed, and the completed partition will be shown.

To delete a partition, you can click the "-" button in the lower left corner of the interface to delete a partition. After all settings are completed, click the Done button in the upper left corner to enter the interface.

11) In the interface shown, all operations related to disk partitions are displayed, including creating, resizing or deleting partitions and file systems. You can view all the changes. If you need to modify the partition again, click the Cancel & Return to Custom Partitioning button to return to the partition interface. To confirm the partition change operation, click the Accept Changes button to return to the interface.

So far, all the basic operations on the partitioning part have been introduced. Now we will enter the network configuration section.

WHAT NEXT?

While this section may seem complex it is very important to make your linux distro function. If not done properly your hard disk may become prone to errors. In the next chapter, we will talk about networking settings that need to be filled before installing the system. Follow along!

CHAPTER NINE

NETWORK INSTALLATION SETTINGS

THIS CHAPTER TALKS about the networking settings that you need to enter before starting the Installation procedure of Linux mint. As we well know that network settings are utmost important for the functioning of a system it is mandatory ti learn about them. Let us go and configure the network setting to complete the installation procedure.

CONFIGURING THE NETWORK

1) In the interface that appears after the previous settings, the KDUMP option is enabled by default. Just keep it enabled so that no other additional tweaks and other settings are required. Then click the NETWORK & HOST NAME option to enter the network setting interface.

In the left pane, the network interface information recognized by the installer is displayed. Click the network interface in the list on the left, and the detailed information will be displayed on the right.

In this sample procedure, there is only one network card. We used ethernet. Ethernet represents the type of networking device and eno16777736 represents the device ID of the network card. Your device id may differ. You can find to using the following command.

```
root@server : netstat

// Look out for the “ethernet” word in the output information.
```

You are using wireless network then search for “wlan”

The right pane in the interface shows the connection status of this network card, such as MAC address, connection rate, etc.

It is possible that you can also set the host name information. Enter the corresponding host name in the Host name input box in the lower left corner.

By default, the network card is disconnected and you can click the ON/OFF (switch) button in the upper right corner to activate the network card. The activated network card cannot automatically obtain an IP address, so you need to manually set the IP address information.

Click the Configure button in the lower right corner to enter the network card information configuration interface.

2) In the configuration interface that appears the wired, wireless, VPN or DSL connection can be configured according to the type of network connected. Here, select Automatically and connect to this network when it is available so that All users may connect to this network. Select the box to automatically connect after the system starts.

3) In the network configuration part, it seems that there are many options to be set, but in fact, not many need to be configured. In the interface shown select the IPv4 Settings tab to enter the IP address configuration interface. First select Method from the options(network connection mode). Commonly used are Automatic DHCP, Manual, Link-Local Only, etc.. If you are not aware or looking for more customizations then Choose Manual.

After chosing manual then click the Add button to add an IPv4 IP address. The IP address added here is "192.234.56.101”. The subnet mask is "255.155.255.0", and the gateway is "192.987.56.1", and then add a DNS servers address “113.5.5.5”. If the network needs to set up multiple routes, you can also click the Routes button in the lower right corner to add routes.

After all settings are completed, click the Save button to save the settings and return to the interface that you are previously in to continue the procedure.

4) After the setting is completed, the network card will automatically try the network connectivity. The configured network will be shown as a summary for your reference. Click the Done button to complete the network settings and return to the interface that holds options for all other settings that are available.

5) So far, the basic introduction of the Linux installation needs to be set is completed. A summary that provides all the settings that you have selected will be appeared on the screen. Make sure that the settings are correct. Click the Begin Installation button in the lower right corner to start installing the Linux system.

Now its the end game. Install Linux in your system.

COMPLETING THE INSTALLATION

1) After clicking the Begin Installation button, the Linux installation progress interface appears. At this time, the system will be installed on the disk. During the installation process, you also need to set the password of the Linux administrator account ROOT. Click the ROOT PASSWORD option to enter the ROOT password setting interface. Make sure that you are entering a secure password as dictionary attacks are possible to be entered using a Live USB of linux system.

2) The installation program will verify the entered password. If a too simple password is set, it will be prompted to reset it. The ROOT user is the super administrator in the Linux system. Therefore, the password setting must be strict, and it is best to set it to include numbers, Password with a combination of letters and special characters.

After entering the password twice, click the Done button to return to the installation interface.

3) In the interface shown, click the USER CREATION option to create an ordinary user. Linux is a multi-user operating system. When using it, it is best not to log in to the system directly with the ROOT account, but to log in through an ordinary user, which helps the security of the system. Therefore, it is necessary to create an ordinary user to log in to the system .

Enter the user's Full name, User name and Password to create an ordinary user. After the input is complete, click the Done button to return to the installation interface.

4) The time required for the installation process depends on the number of selected software packages, generally it takes 10 to 60 minutes to complete the installation. As shown, after the system is installed, the settings under USER SETTINGS will be unavailable, so you need to set ROOT PASSWORD and

USER CREATION before the system is installed.

After the installation is complete, click the Reboot button to restart the system and finish installation.

STARTING LINUX SYSTEM

1) After the installation is complete, the system will be automatically ejected by the CD. If the CD is not ejected, you need to remove the installation media. After the server restarts, it will automatically enter the boot interface. There are two boot options, the first is the normal boot program, the second is the boot program to enter the rescue mode, the first one is selected by default. If all done correctly as mentioned press Enter to enter the boot process.

2) By default, a graphical interface showing a progress bar is hidden behind the startup process. If the graphical interface is not installed, the character interface will be entered by default. If the graphical interface is installed, the graphical interface will automatically start. Now click the ixdba user, and then enter the password to log in to the system. If you need to log in with another user, select the Not listed button below and enter the user who needs to log in using the desired Name and password. You can comfortably switch users to log in to the system.

3) After logging in to the system through the graphical interface, the language selection interface will pop up. You can select the desired language as needed, and then select the default as English. Then click the Next button to enter the next interface.

4) Enter the keyboard selection below and select the default English (US), and click the Next button to enter the next interface where all this completes.

5) At this point, the user initialization process after installation is all over, click the Start using Linux Mint button in the interface to start the Linux learning journey.

WHAT TO DO NEXT?

Congratulations. You have now installed the Linux Mint in your system. After a successful installation it is highly recommended to update the system so that there will be no lags or bugs in the installed operating system. For every couple

of months the linux kernel can be updated to improve the performance of the Linux Mint.

With this, we have completed the important part of this book. In the next module of this book we will introduce complex Linux topics that are important for making Linux a better place for programmers and developers. Let us dive into the second module of this book. But before going, have a look at the next chapter that simplifies U disk installation process for you.

BONUS : U-DISK INSTALLATION PROCESS

AS EXPLAINED in the previous section there are different ways to install Linux in a desktop computer or a laptop or sometimes even in server based systems. We have discussed in the previous section using the much used CD rom files. In this section, we will discuss about USB installation procedure which is now being used as a mainstream way to install Linux. Follow along carefully to know more about it.

U DISK INSTALLATION PROCESS

Most PCs and servers nowadays are not equipped with CD-ROM drives, and are replaced by USB interface devices. At the same time, the USB 3.0 specification has been widely popularized, and the data transmission speed of U disk can reach gigabit per second. There are tons of products in market ranging from 2gb to 2tb according to your requirements.

Due to this uprise in the installation procedure, there are now more and more applications based on the USB interface, such as USB keyboard, USB mouse, USB optical drive, etc. When installing the system, although USB optical drive can be used to install, the current cost of USB optical drive is relatively high, and the installation of the system through U disk is not only of low cost, but also fast and convenient. In large-scale systems installation, through U disk has become a mainstream procedure.

The basic steps for installing a Linux system through a U disk are as follows:

1) First, a U disk of about 8GB is needed. The larger the capacity, the better it will be. The full ISO image file size of Linux Mint has reached about 7GB now a days. Always make sure to verify the file size before deciding to make an USB

installation medium.

2) Download the ISO image file of Linux Mint from the open source site (http://mirrors.163.com). Or you can just visit the official Linux mint website to select your preferred disk image file for installation.

3) Download a U disk burning tool from the Internet. Common U disk burning tools include UltraISO, USBWriter and Rufus. USBWriter is recommended here, because USBWriter is easy to use and can be used to burn a disk image file in one step. Although UltraISO can also burn ISO images to U disk, the burned Usb disk will sometimes have problems during the installation and boot process. If you are using a windows system to make the installation medium then we highly suggest you to use Rufus for making the Usb disk as it is more convenient and recommended one.

4) USBWriter is an open source small tool, which can be downloaded from http://sourceforge.net/projects/usbwriter/ . It is a U disk burning tool that is specially used for Linux systems.

Its working principle is to divide the U disk into two partitions, a FAT32 partition (only a few megabytes in size) that is used to install the boot program, and the other Linux partition is used to store the system installation package.

There is sometimes a known problem:

The capacity of the U disk burned by USBWriter will automatically become smaller. This is because only one FAT32 partition is seen, and the other Linux partition is invisible under Windows. But don’t worry, it’s easy to restore the USB flash drive to its original size. Download DiskGenius utility tool from the Internet to restore the USB flash drive to its original size.

5) Use USBWriter to burn the downloaded ISO image file to the U disk. First select the location of the ISO image file, and then specify the U disk that needs to be burned to. For example, a 16GB U disk is selected. After selecting, click the Write button to start the burning process.

After burning the ISO file to the U disk, all the previous data on the U disk will be lost, so before burning, back up the data on the U disk.

6) The U disk burning process is relatively fast, usually 10 to 30 minutes. If there is no error during the burning process, the burning process can basically be successful. After the U disk burning is completed, it will prompt that the burning

is successful.

7) After the U disk is successfully burned, you need to re-plug the U disk. At this time, in the Windows Disk Manager, you can see that the 16GB U disk has become about 6MB in size.

8) After the U disk is successfully burned, insert the U disk into the USB interface of the server, then enter the BIOS, set to boot from the USB device, and then restart the system to enter the interface that provides more options.

9) The following installation process is exactly the same as the installation method of the optical drive that we introduced before, so the introduction will not be repeated. But one thing to note is that during the disk partitioning phase, you will see two disks.

After the installation procedure the USB disk will be ready and you can use it to install Linux in any computer easily.

WHAT NEXT?

With this, we almost completed the first module of this book successfully. Even though being a short introduction to Linux and its installation procedure this module is opening to the much important roadmap that you are going to take now to become a Linux enthusiast and a cyber security specialist.

In the next module of this book you will learn about important Linux topics such as Linux file system, Process management and Log analysis. There are a lot of other topics that we are going to discuss. Follow along!

LINUX BASICS FOR HACKERS

INTRODUCTION

IN THE PREVIOUS module of this book you learned about basics of Linux that are absolutely necessary to wander around the Linux world. Learning Linux is not an easy task as it is developed by thousands of developers from twenty years.

A small titbit:

Do you know that the number of lines that are written to make Linux kernel work are stunningly 1 million lines? Yes, it's right . Linux kernel is approximately 1 million lines large and is still running. So, if you want to learn all the components of Linux and all the drivers that are supposed to be run with the Linux kernel then I think that your approach is wrong.

HOW TO LEARN LINUX?

Learn smartly not effectively should be your motto while starting to learn Linux. You are trying to be a hacker so learn Linux like a hacker should learn. Don't worry about virtualization utilities that come with Linux kernel and focus about network monitoring and spoofing tools that can help you become a hacker.

In the previous module we explained basic stuff that is the foundation for any Linux distro. We now think that you are ready to learn Linux as a hacker now. In the coming five chapter you will be learning about important linux topics that can seamless increase your performance as a hacker. Follow along!

CHAPTER ONE

INSTALLING SOFTWARES AND PACKAGES IN LINUX

AS WITH EVERY OPERATING SYSTEM, installing software is a mandatory use case for Linux also. Linux uses organized package management systems and installs software in a more perfect way when done correctly. It may feel like a bit of complexity when first dealing with installing and updating software in linux but as time goes on you will feel how organized and easy Linux installation philosophy is. As a hacker it is important to understand and find easy ways to install software to not break the system. Even if you are dealing with Live USB to hack systems it is very essential to understand how installing packages work in Linux.

To help you understand these concepts we are providing various use cases and examples in this chapter. Follow along!

HOW IS IT DIFFERENT FROM WINDOWS?

When compared to Linux windows installation procedure may seem straightforward but it’s not. Windows installation files are especially large because they create a sandbox to include all the API’s that the software calls at different instances. This makes installation file big and may cause lag issues down the road.

Linux however uses different approach to install files. It uses packages which are tightly enclosed and independent and are updated by their own instance. This may cause problems sometimes but most of the time it just works perfectly.

HOW TO INSTALL SOFTWARE IN LINUX MINT?

As we are aware that Mint is a light weight Debian based system that can work even under low end based systems. Different Linux distros use different package managers. For example, Arch Linux which is a developer favorite Linux distro uses Pacman to install software.

All the Debian based systems fundamentally use “apt” to install software. Apt is a simple package manager that tightly integrates all the components and installs them in the Linux machine using the instructions provided in the script file.

How does it work?

When a software is called first the script file is downloaded and then the package manager will analyze all the components that are necessary for the system to function properly. If any component is already installed in the system then it skips to the next one to download it. After all the components are installed the software will be packaged and installed within seconds into the Linux system.

Here is a command:

```
apt-get install python

// This will download Python to the Linux system
```

HOW TO SEARCH PACKAGES?

Apt almost provides all the famous packages that Linux enthusiasts and developers root for. However, there are a lot of third party packages that may not be available because of compatibility reasons.

If you are trying to download a package and if not you are sure whether that package exists or not in the package manager then you can use the search command to find it. If you do not get any accurate results for your package name then it is evident that the apt is not supporting your package.

Here is the command:

```
root@server : apt-cache search Python4

// This will search the apt repository and will display “no results” in the command line
```

To get more accurate results please use the web version of apt repository where all the available packages are sorted even according to a category.

If you are using Arch Linux, Pacman also provides a search option to check packages before downloading and installing them.

HOW TO REMOVE SOFTWARE?

Removing software from an operating system is as important as installing software into a system.

You may wonder why it is important to remove software that is significantly not doing any damage to your Linux system? But listen to experts because idle software that are outdated can conflicts with the newly installed software.

If there are potential conflicts in a Linux system it may lead to breaking the whole operating system. Removing unwanted software also improves system performance and can help users allocate those wasted memory resources to something worthwhile.

Ethical hackers should also understand how to remove software because when they detect any trojans or spiders that are inserted by the attackers to hijack system they should be able to remove their traces from the system as soon as possible. Some high risk trojans will not get removed until you end their process by manually finding them. It is tough to remove software if a hacker users potentially dangerous ways to implement them.

For now, we will learn about removing software by using a package manager like apt.

Here are some commands:

```
root@server : apt-get remove python

// This removes python packages along with all the dependencies that are not being used by any other package in the operating system.
```

HOW TO UPDATE PACKAGES/SOFTWARE?

Updates are a way to improve the performance of your software. Almost every

software release updates to provide new features to their users. In windows system, when you click update it first downloads an executable file and asks you to install it manually.

In linux however all you need to do is to use the package managers to install them and all of its dependencies. You can also manually provide instructions in the installation file to not update the repositories that you are not willing to update. In linux, updating software gives a lot of freedom unlike traditional operating systems like Windows and Mac.

Linux also provides an easy way to update all the software that are available with a one click. During this update process Linux kernels and header files will also be updated to support the newer versions of the software.

Here are some commands:

```
apt-get update

// Use this command to update all your software by one click

Apt-get upgrade

// use this command to upgrade all the software packages that are installed on the system

Apt-get upgrade python

// This command will check whether python has any updates and if there are any updates available then it will install them cumulatively.
```

GUI BASED INSTALLATION PROCEDURE

Linux is a terminal centric operating system and will always support the philosophy of using commands to change the system files such as installing/removing software. However, it should be understood that GUI is also supported extensively by some Linux enthusiasts.

There has always been a wild debate about GUI installers that are developed by third party developers. Unlike in the beginnings of Linux deployment now a days Linux is also being used as a daily operating system by a lot of users. And, it is a known fact that a lot of them are not tech savvy and prefer GUI based installers because they are more easy to operate.

This is the reason why some of the Linux distros that have huge downloads are

including GUI based installers in the operating system. For example, a popular arch based Linux distro Manjaro comes with GUI version of Pacman and solves a lot of installation problems that users may face.

For Debian based system synaptic installer is highly recommended and is binded with a lot of Debian based distros.

If you do not find Synaptic installer we suggest you to use the below command

```
root@server : apt-get install synaptic

// This will install the synaptic GUI installer in the Linux system
```

Now, all you need to do is search your preferred package in the search column of the GUI installer so that it can be installed seamlessly along with all of it's dependencies that are required to run the software.

Removing software is also easy with a GUI based installer. All you need to do is find the software name and select it to remove from your system along with all of its dependencies.

Now, in the next section of this chapter we will talk about git which is a famous developer tool that needs to be mastered if you are into open-source and working in teams. You might have already heard "Github" that uses the git structure to share code with others.

GIT INSTALLATION

If you never heard about Github it is a git based code sharing platform that offers both free and professional plans. In the free plan your code will be public and is therefore automatically comes under open-source license.

What does git do exactly?

Git uses a complex technique that involves commits to automatically update the source code in a project in a way that it supports all the present code.

It becomes handy especially if you are working in teams to check the compatibility of the newly written code. It also makes testing process easy for enterprise applications.

For hackers, a lot of scripts that automate certain things are usually uploaded by hackers for the community. A lot of key logger tools and monitor ng tools are also uploaded by the open-source community.

If you are a beginner and are trying to use that code for your exploitation techniques then it makes sense that you need to clone that repository into your Linux system. To make that possible use the below mentioned commands.

```
root@server : apt-get install git
// This installs the git supported system in your Linux system
root@server : git clone https:github.com/enterurlhere
// This clones your desired repository into the Linux system
```

With this, we have completed a brief introduction to installing, updating and removing software using package managers in Linux. You can also install software in Linux using the executable .deb files after giving necessary permissions. In the next chapter, we will learn about process management that is said to be a necessary skill for hackers who are using kali linux for exploitation purposes. Follow along!

CHAPTER TWO

PROCESS MANAGEMENT IN LINUX

AFTER HAVING a thorough understanding about the installation philosophy and Linux uses you are Now all set to enter into a new Linux arena which helps you to ignite much more significant thoughts as a hacker. Linux uses processes to maintain and interact all its components with Linux kernel. If there are no process running then we can confidently say that the Linux system has been shut down. As a hacker understanding processes, manipulating them and ending them whenever necessary is a blessing skill. In this chapter we will not only provide basics about processes in Linux but will also provide examples which can help you understand the foundations of Linux more concretely. Follow along!

WHAT ARE PROCESSES?

Processes are what makes Linux programs run. They are a representation that the software/ command line utility is interacting with system resources such as Linux kernel to do what it is ought to do. Every software that is running in Linux has a process and if we end the process manually then the software will stop working.

As a hacker, when you get access to a system you will be easily monitored by the anti virus processes that are running in the Linux system. To make them stop sending your logs to the system administrator you may need to end their processes manually.

While it may seem simplistic to just end processes like in Task manager using windows this can have some serious consequences if done wrong. You need to have a practical experience and a good theoretical understanding of process management to achieve your wishes.

In the coming sections you will be learning some tips to find and manage processes along with a way to automate them according to your convenience.

HOW TO VIEW PROCESSES?

Whenever a Linux system boots up some processes will start opening automatically. Every time when you open a terminal or a web browser their respective processes will also be opened. When you close them only some of the processes will be automatically shut down while other will be running in the background consuming your system resources.

Here is the command for viewing processes:

```
root@server : ps
// This displays all the processes that are running in the Linux system
```

Here is a sample output:

```
PID
45637
// The numbers are the identification for a process
```

If you want to make any changes to a particular process then you should note down the process number. The output information will also show other information such as the process name, time and the type of command it is using. If you have opened the process from a terminal then the bash cmd will appear in the output information.

Note :

Finding processes is sometimes an overwhelming task because you need to search from tens of processes that are running. Always make sure that you are confident before changing the process status. If you by any mean end the system processes that are running then your Linux system may stop responding until the next reboot.

HOW TO DISPLAY ADDITIONAL INFORMATION?

Usually by using the ps command you can only get basic information. You can display additional information about processes using the aux command. Let us check it with an example.

```
root@server : ps aux

// This prints an output that describes a lot of information about a process
```

Here is a sample output:

```
PID - 21314 %CPU -23% %MEM - 32% Start Time - 22:34

// It displays additional information such as the memory and CPU power it is consuming. It also displays start time to easily estimate the process name
```

The aux command also describes about the instance that started the process. This is essential for forensic investigators because they can easily track the origin point of the attack using this procedure.

HOW TO FILTER PROCESSES?

Imagine that you are starting to find a process that you badly need to stop but you can’t easily find it because there are a lot of other processes that are being displayed on the output screen. If you are smart you would find a way to filter the process you need instead of manually going through all the processes to find the one you want.

If you are looking to be smart then Linux is there for you. Linux specifically provides functionalities to filter processes using the grep command. Here is an example for your understanding.

```
root@server : burp suite

// First start a process or any software that you like using the default command

root@server : ps aux | grep burpsuite
```

```
// Now use this command to filter the process that starts by using the “burp suite” command
```

This will display an output like we have shown before with all the information that one needs to understand about the process.

HOW TO FIND PROBLEMATIC PROCESSES?

Before describing about greedy processes we should talk about a scenario that have become popular these days especially in the windows systems.

A lot of spammers and hackers are now a days including mining programs in applications to install them your system. When you willingly or by any other reason install them your computer will become very slow and will not function normally.

Why does this happen?

Because, the mining program forcefully uses all the system resources and will run in the background. A normal user can’t observe it because it will be hidden in an invisible tray. Only professional who can access the terminal or can analyze services in the operating system can only understand that there is a malware in the system.

In the similar process imagine greedy processes in Linux as mining programs in window systems. They may not be harmful but are definitely a bad use case if you using Linux as your daily driver. As a system administrator or a hacker you should be aware of killing the greedy process that maybe installed by other users or maybe sometimes even hackers.

HOW TO FIND TOP PROCESSES?

To find top processes all you need to do is to enter the following command in your terminal as a root user.

```
root@server : top

// This will display an output information that is allocated with maximum system resources
```

If you feel that any of these top processes are not necessary for your system then you can end them with the kill command which we will be discussing in the coming section.

HOW TO PRIORITISE PROCESSES IN LINUX?

In a linux system with a ton of processes running as an ethical hacker or as a system administrator it is always recommended to prioritize your preferred processes using the nice command.

Why should you prioritize?

If you are an individual user then prioritizing may not seem intelligent because you have all the resources you need. But imagine if you are in an enterprise and are using the server resources that linux is maintained along with other users then it is impossible to use all the system resources for yourself. If you do it then your system administrator may find out and can sometimes even temporarily block your access.

A lot of newbie hackers when they crack a system attack all the resources all at once notifying sysadmins and shutting down all the access to the system. If you are a smart hacker you will carefully prioritize the processes without gathering much attention and when you get all the sensitive information you need you will then kill the process and remove all the logs so that there won't be any footprints left for the forensic investigators.

WHAT DOES NICE COMMAND EXACTLY DO?

nice is a linux utility command that is developed to give score to a process according to its importance. The priority number can be customized but they usually range from -19 to +19. In here, 0 is the neutral range whereas -19 is used to determine the weakest preferred process whereas +19 is used to determine the most preferred process.

Here are commands:

root@server: nice -n 5 /bin/terminal

// This is a process with priority 5

root@server: nice -n -10 /bin/game

```
// This is a process with priority -10
```

So, when there is a descent in system resources first the process with -10 nice priority will end automatically. Lower priority processes should not be system processes at any case. System administrators usually spend a lot of time to prioritize the perfect processed for the system. If you are a hacker you should also learn to prioritize to effectively manage the resources while trying to steal information.

At any time you can change the priority order of the process using the renice command.

```
root@server: renice 13 7732
// Here the priority order of the process id 7732 is changed to 13
```

In the next section, we will talk about killing processes. Learning to kill processes is a much hyped skill for hackers as they often need to switch off the intrusion detection systems that the administrators use.

HOW TO KILL A PROCESS?

Some processes are often called as zombie processes because they consume a lot of resources even when they are freezes or stuck in the system. If you did find any processes that are not worthy of the resources they are using then it is a wise choice to kill them at any cost.

Here is a command:

```
root@server : kill -1 3638
// This -1 says that the process should be restarted after killing it first
root@server : kill -9 3638
// This -9 says that the process needs to be terminated immediately
```

In the last section of this chapter we will discuss about scheduling processes.

HOW TO SCHEDULE PROCESSES?

In any operating system or computer operation management system scheduling is an important skill to master. Scheduling not only decreases workload but also helps you understand how to use resources smartly.

For example, you can schedule to start an antivirus process only when you have downloaded a file from the web browser.

In Linux, at utility is used to schedule a process.

Here is a command:

```
root@server : at 9:00pm

// This will start the at command utility that can schedule the process

root@server : at > /root/users/startscript

// At the above mentioned time the script mentioned will start as a process
```

By using scheduling functionality hackers can run remote programs even when they lose access. You can bind programs with metasploit vulnerable scripts to make them send information to your mobile or server within seconds.

WHAT NEXT?

With this, we have completed a brief and complete introduction to process management. In the next chapter, we will talk about file management in Linux with tons of examples. Let us go!

CHAPTER THREE
FILE MANAGEMENT IN LINUX

IN THE PREVIOUS chapter we have coherently discussed about the importance of process management for hackers. If you are serious into hacking with Linux you should also know a sufficient information about file management and directory permissions in Linux.

A lot of new hackers even after having obtained the access will not be able to grab the information they need because of the lack of file and directory permissions. Kali Linux uses complex file structures and directory management philosophy to counter the loopholes that usual Linux distros may face. In this chapter we will discuss a lot of these scenarios with concrete examples. Follow along!

WHAT ARE USERS AND GROUPS IN LINUX?

Every operating system uses the concept of users to determine who uses what. For example, windows uses something called administrator account to access the system files. If you are not an administrator then there is no chance that you can access the C program and system files that are important for hackers.

In the same manner, Linux uses much more complex user management system for easier access and advanced security functions.

Users - Users are individuals who can access a particular area of the system

Groups - Groups are bundle of users who can access particular resources combinely.

We will now discuss about different types of users in Linux in much more detail.

1) First you need to know about the head of the Linux system. That is, the most powerful one of all. Usually it is called as a root user. Root user is the most powerful and can do anything in a system. He can add users , groups and delete them whenever he want to. Usually root access is given to the system administrators in an enterprise. As a hacker, your main motto should be to attain root access because with it you can do anything you want to.

2) The next is simple. Normal users are the users who doesn't have permissions to have root access. What they can access is completely decided by the root user. If there is a service company, developers will only get access to coding resources where as management employees will only get access to management files. Every enterprise uses their own ways to decide what the users should access.

HOW TO GRANT PERMISSIONS?

Linux administrators usually use permissions to control how users can interact with the system files and directories. There are commonly three type of permissions namely : read, write and execute.

A) read

This is permission to just to read the files. You cannot any way modify or delete them. These permissions are usually given to the entry level users of an enterprise. It is represented by "r" in Linux command terminology.

B) write

This is permission that can make the users to write the file. To write means to modify the content the file. However, even after writing the file it is not guaranteed that you can execute the newly written file. For example, git branches contributors are given written permissions to a file in the repository but are executed only after it is accepted by the administrator of the git branch. It is represented by "w" in Linux command terminology.

C) Execute

This is the high level permission that root users have. If a user gets execute permissions then he can not only read and write files but can also execute them to change the system files and directories. When a hacker tries to attack a system he often looks forward to gain executing permissions. However, remember that sometimes root administrators only provide execute permissions without giving

read and write permissions. It is represented by “x” according to Linux command terminology.

With this, we have completed a brief introduction to permissions. In the next section, we will talk about how to give permissions for an individual user or a group from the command line.

HOW TO GRANT PERMISSIONS TO AN USER?

When you grant permission to an individual user he can interact with the file or directory without any errors.

Chown command is used for granting permissions for an individual user in Linux.

Here is the command :

```
root@server : chown sample /home/pictures
```

In the above command “chown” is the default command that tells the Linux kernel to provide permissions. “Sample” stands for the name of the user and the “/home/pictures” is the directory that is being given access to.

HOW TO GRANT PERMISSIONS TO A GROUP?

Just like how a root user allocates a permission to an individual user he can do it for a group.

chgroup is the command that is used in Linux to allocate permission to a particular group. In enterprises, creating and managing groups is an absolute necessity. While doing projects that are short term allocating permissions solves a lot of problems.

Here is a command:

```
root@server : chgroup hackers /home
```

In the above command chgroup is the default identifier that allows groups to allocate permissions. Hackers is the group name whereas /home is the directory that the permission is being given to.

In the next section we will learn how to view the current permissions for a file or directory.

HOW TO CHECK PERMISSIONS IN LINUX?

If you are unaware of what permissions you are currently holding then you can check using the following command in Linux

```
root@server : ls -l /var/games
```

This command will display an output information saying about your current user status and whether or not you have access to this directory. You will also information about the owner of the file. Another significant option is details about the time when it was modified. If you are a system administrator you should be constantly checking the file modification status to be safe from attackers who try to gain the system permissions at any cost to steal sensitive information.

If you are a hacker trying to gain control over the system you should know a way to modify permissions in linux. This is where the “chmod” command comes into use.

HOW TO MODIFY PERMISSIONS?

Permissions are generally changed in linux using the chmod command. All you have to do is add the permission that you are willing to change with chmod as a root user to change it.

For example:

```
root@server : chmod +x sum.img

// This will give the current user the permission to execute the image file
```

Advanced hackers use SUID to grant temporary permissions for the file they are going to execute. Learning about it is out of the scope of the book. If you are interested we suggest you to learn about SUID and Umask before jumping into it.

With this, we have completed a brief introduction to file management system in Linux. In the next chapter we will talk about important stuff that a hacker needs mostly that is maintain anonymity. We will also talk about TOR, dark web and VPN to help you understand how to protect your privacy even when attacking systems. Let us go!

CHAPTER FOUR
BEING SECURE IN LINUX

IN EARLY 90'S it is almost difficult to track hackers who are using techniques like phreaking to make free calls from the telephone services. However, after the rapid growth of Internet and its potential to earn money has made ISP's restrict users and track them. Even governments now a days restricts their users from accessing websites that are illegal according to their rules. For example, a lot of countries like India, China restrict torrent websites. China even censors websites such as Google, Facebook because they feel that they are missing the user privacy.

While banning websites is often a way to destroy economic growth of their rival countries now a days user privacy is also a huge issue that is being taken seriously. Much recently US government has decided to restrict its citizens from downloading Tiktok, a famous video sharing social networking platform because of the claims that they are misusing the user data. While some of these claims may be debatable hackers who are trying to attack systems of enterprises that have quality intrusion detection systems should be aware that they are not being tracked. This chapter will try to ignite some valuable information to you. Let us go!

WHAT CAN ISP'S TRACK?

It may seem scary but the fact is that your ISP will have every website you have ever visited in their system logs. Some unethical ISP's even try to sell this data to multinational companies who need data to create machine learning algorithms that will further target your interests and can even change your political views.

So, what should we do to stop being tracked by government and ISP's? A lot of network enthusiasts and privacy concerned engineers battled with this question

for years. After spending hundreds of hours for research a group of network specialists created something called as a TOR network.

Before understanding about TOR in the first place we need to talk about how IP addresses are used to track us with an example.

HOW DO IP ADDRESS TRACK US?

To be clear, an IP address is like an address box for your device such as a mobile/router or sometimes even a wireless connection.

For example, let us assume that you did send a WhatsApp message to your friend. When you click the send button the message content will be sent in the form of packets to a nearest router. The packet has the information about the source address and will also have a destination address. The network packet travels just like a post that is being sent to a particular address. When the destination receives the packet the status of the message will be changed in the messaging platform.

Usually a normal network packet hops between 20 packets before reaching the destination. At all happens within seconds so for normal users it doesn't even matter. But for hackers and security enthusiasts these packets can provide a lot of unencrypted information. ISP's and governments tracking can also get a lot of sensitive information with the help of these network packets.

HOW TO FIND WHERE THE PACKET IS TRAVELING?

Linux provides a special command to help you understand where the packets are traveling. It is called as trace route command.

```
root@server : traceroute yahoo.com
```

This will display a trace route for the url address that you have provided. It will show the number of hops that will take to reach the destination and will also provide the size of the network packets.

With this sufficient information you are now all set to understand the philosophy of Onion network. Let us find more about it.

HOW DOES TOR NETWORK FUNCTION?

In usual network communication all the packets that are travelled through the routers are tracked and are controlled by specific ISP's. Whenever a legal authority or a government requests for the access of these packets that travelled through these routers they provide them afraid of measures that may hurt their enterprise. It may work well for the company but for an individual when privacy matters this may feel like a huge leap.

While normal network request travel through these routers all the network requests that are used in a TOR network are sent through a highly secured network of computers that are being volunteered by network specialists all around the world. All these routers are not only can't be tracked but are also encrypted and cannot be accessed by anyone to reveal their sensitive information.

TOR not only provides complete anonymity but also helps you to access the dark web that is a walking heaven for hackers. In the dark web we can find .onion websites that are filled with several illegal activities. We suggest you to only access the websites that may serve a purpose but not that provide hacking services.

If you feel like experimenting with the TOR network you should then install the TOR browser that is developed using the Mozilla source code. It is fast and automatically connects you to a TOR relay.

IS TOR COMPLETELY SAFE?

While it is absolutely true that TOR project is now the only trusted way to gain maximum privacy but it still has its loopholes. Years back there was a leak saying that the US government has a list of some computers that are being used as TOR servers. However, the network engineers that are maintaining the TOR project said it is not true.

Another issue is that popular websites such as Google and Facebook discourage users using the TOR network by constantly irritating with security checks. If you are looking forward to use TOR as your way to interact with internet then as a hacker you should be aware of these drawbacks.

In the next section, we will talk about proxy servers that are an another popular

way to hide your identity to stop being tracked by government/ISP's.

WHAT ARE PROXY SERVERS?

To say in layman terms proxies are middle men that acts like a mediator between the source and destination. They are intermediate and allows all the traffic coming from both source and destination to pass through them.

If you are curious to know how a proxy server works then follow this simple explanation.

Imagine that you want to visit a website using your web browser. In this scenario,Your IP address is known as host address and the IP address of the website you are visiting is known as destination address. Whenever you send a request to that website using your web browser then your ISP will log down both of these requests.

However, now assume that whenever you send a request to the website instead of going directly to the website in a situation where proxy is there all the traffic will be first sent to the proxy server. When the traffic reaches the proxy then all the traffic that was going to the destination will change its IP address. In this scenario it is difficult to track from where the traffic is originally because of an intermediate that exists. In the same way the destination website also sends response data to the source (i.e proxy) and now proxy will send all the data to the source address. So, the data will be displayed as-usual in your web browser.

WHAT ARE THE DRAWBACKS?

Like every solution to be anonymous proxies too come with some drawbacks. The most important one is because there involves a middle man who receives and sends the data for you the speed will be reduced significantly.

Also proxies are definitely not completely safe. While they may usually secure from accessing blocked websites or moderate stuff that involves hacking procedures you should not completely rely on them. Especially with the free proxies. Experienced hackers use proxies that are Sock5 and secured. If you do some petty stuff and if there is a search warrant to search for the proxy logs then the proxy providers may give away your details because of the pressure from the law enforcement. So, make sure that you are using proxies from highly secured

sources.

HOW TO PERFECTLY UTILIZE PROXIES IN LINUX?

Linux users love proxies. Proxies not only help them stay anonymous but also significantly increase their performance with the internet. Kali Linux, a famous hacking distro comes with a proxy utility tool known as proxy chains that bundle two or more proxy addresses to send and receive data to the operating system. This not only makes you completely anonymous but is also secure and tracking you can be a nightmare.

Here is the command :

```
root@server : proxychains
```

No matter what task you do , if you are willing to be anonymous then you can enter ‘proxy chains’ command in the beginning so that it will be routed through the proxies that are entered in it’s config file. In the config file of proxy chains you can enter a single proxy address or can enter multiple of them. Advanced hackers use random proxies and multiple proxies to improve their security.

In the next section, we will talk about Virtual Private Networks (VPN’s) and encrypted mail in detail. Let us go!

WHAT ARE VIRTUAL PRIVATE NETWORKS IN LINUX?

You might be often seeing advertisements in Internet about VPN providers such as Nord VPN, Express VPN etc,. If you still don’t know what the hell they are then don’t be confused because they are just like proxy servers but with more efficiency and completely encrypted data so that even the VPN providers cannot read or access what you are doing with your internet.

Vpn works basically by using the same principle as proxies. While using a VPN all your data/traffic will be sent through a intermediate network device such as a router. All the received data to the router will be sent to the destination and vice-versa.

The major fuss about VPN’s is they are easy to use and consists a lot of proxies

according to your preferred location. So, they can be mainly used to evade restrictions that your governments or enterprises imply on you. Also, they are cheap now a days and can cloak your information so that you will not be a victim to spoofing or snooping attacks by the hackers.

The famous VPN providers are Nord VPN , Express VPN and Hide My ass VPN. We suggest you to do a complete research about their plans and prices before making a purchase.

WHAT IS AN ENCRYPTED MAIL?

We all know that E-mail is a professional way to communicate as a worker or a freelancer. Even personal communications can be done using an E-mail. However, are you sure that all your data that goes through an E-mail is safe?

You don’t. Because, all the free email providers save all your encrypted data in their servers to use them for developing algorithms that are interlinked with machine learning and artificial intelligence. Yes, we are now living in a dystopian world where we don’t even give a damn about our personal data that can be used to change our thoughts and political views.

To not be a prey to multi national madness we suggest you to use encrypted e-mail services such as Proton mail. They work as good as famous Email providers like Gmail. Also, you will be completely secured and will not become prey to any phishing or spam mails because of their advanced filtering features.

WHAT NEXT?

With this, we have given a complete introduction to the security concerns that a beginner Linux user can have. As a hacker you need to constantly delete your logs and cookies to not being tracked by the Network providers or internet gaints. With this sufficient knowledge you are now all set to learn about Logging system in Linux. This is the final chapter of this module and can give us a good clarity about an important expertise that all hacker should master. Let us jump into into right now.

CHAPTER FIVE
LOGGING SYSTEM IN LINUX

IN THE PREVIOUS chapter we talked about ways to make you be secure and anonymous in the internet. As a hacker staying in the dark and stealing sensitive information is usually your main motto. As a fellow cyber security enthusiast I would suggest you to maintain a good grasp about different logging systems in Linux distros to help you understand what is actually going in the system that you are trying to gain access. Also, deleting sensitive log files that can reveal your identity is also an important task for hackers to master. To help you understand different logging systems and daemons that make up the log files we have introduced this chapter with tons of examples. Follow along!

WHAT ARE LOG FILES?

Log files are information about all the important events that occur in an operating system. Traditional operating systems like Windows and macOS doesn't give access to log files unless you have special administrative privileges. Linux however provides details about all the errors and warnings that exist in the system. Some log files are accessed even when there are no root privileges. However, all the log files that are created can be modified or deleted only if you have root privileges.

HOW LOG FILES CAN HELP HACKERS?

Imagine log files as a path to your destination (I.e your IP address / other sensitive information). Each time a new user gains access to the system the Linux system automatically registers all of its information and will connect only after this procedure. This security feature can be a nightmare to hackers because

forensic enthusiasts can extract all the information / proxy addresses and can investigate further.

To make sure that you are not to be tracked whenever you gain access to a system you should head over to the logs location and should delete them permanently. Administrators can find that there are missing log files but can never know who did it. There will be suspicion but not conformation by deleting log files.

HOW LOG FILES CAN HELP SYSTEM ADMINISTRATORS?

System administrators and database specialists often need to troubleshoot different Linux systems in their professional life. Some administrators solve problems using remote ways. To make administrators understand what the problem is he needs a definite access to log files. Log files will help them understand what the problem is. Without log files, it would be a tremendous task to manually check where the problem exists.

How logs are created?

Log files in linux are generally created using a specific daemon known as syslogd.

But, what is a daemon ?

A daemon is a system service that automatically starts whenever a system is rebooted. They are the first processes that run no matter what software are installed in the system. They are also known as default system processes. syslogd is a specific system process that takes care about creating log files for every event that runs in the operating system.

Also, it is important to remember that every Linux distribution uses its own way to create log files. For example, Debian systems use ‘syslog’ to create log files in the system. Other operating systems like Arch and Redhat uses different logging systems according to their use cases. The choice of logging system depends on the majority of users the distro is being used. Debian systems are the windows of Linux world and is therefore light-weight and can be used as a daily driver. This is the reason why Debian uses simple logging system such as ‘rsyslog’.

Complex operating systems such as Red hat Linux that are used in server administration needs a lot of additional factors while creating log files as they

are often need to be used for troubleshooting purposes.

HOW TO UNDERSTAND AND ANALYZE LOG FILES?

In this section, we will give a simple procedure to analyze log files that are created using syslog daemon that is present in Debian systems. All you need to do as a prerequisite is to install your favorite text editor to read the log files.

After installing a text editor, head over to the terminal and type the below command.

```
root@server : locate rsyslog
```

This default command searches all the files in the Linux system. Make sure that you have given root permissions using the ‘su’ command so that the command will search all the system files. After a couple of seconds the output information will show all the locations where the log files are present.

For example :

```
/etc/ home/example.rsyslog
```

This is the format for the files that we will find with the above mentioned command. After selecting a log file to analyze head over to the location and open it using your favorite text editor.

Now, you will see a bunch of default code that exists whenever a log file is created using the ‘rsyslog’ daemon. Don’t edit anything and head over to the end of the log file where you will find a section with the title ‘RULES’ as shown below.

Here is the format :

```
### RULES
## Enter rules here
## End of the log file
```

Now, you need to enter your preferred rules here. But first, analyze already present rules to understand how logging system works in linux.

You will see a list of rules in the following format :

```
facility.priority action
```

What does these individual identifiers mean?

A) facility

This is the identifier where we select the service that we are using to create log files. Also, it specifically says which part of the Linux system the logging system is interacting with.

Here are some examples :

Cron , daemon , kern and user. You ca also use any of them just by using the comma between them.

b) priority

This is an identifier that says which type of log files need to be created using the daemon that is running. There are different type of priority operations such as creating debug files or error files or alert files according to your preference.

Note :

If you use * then all the priority options that are available will be logged into the log file. Remember that the lowest priority is debug whereas the highest priority is panic.

C) action

This is an identifier that provides the location where the log files need to be saved in the directory.

Here is an example rule :

```
Kern.error \etc\home\sample.syslog
```

Here, kern stands for facility and error stands for priority. \etc\home\sample.syslog is the location where the log file will be created automatically.

You can also use log files to automatically clean up files. There are a lot of use cases with the logging system and as a hacker you need to master them to execute your ideas perfectly while tracking or cracking a system.

WHAT NEXT?

With this, we have completed our second module of this book where we have discussed a lot of basic information about the Linux operating system that can help us to improve skills as hackers. As a hacker to exploit systems you need to have a good information about programming languages such as Python. In the next module of this book we will discuss in detail about different component of Python along with tens of examples. Head over to the next module to further gain knowledge to be a better hacker. Let us go!

PYTHON SCRIPTING AND BASH SCRIPTING FOR HACKERS

INTRODUCTION

IN THE PREVIOUS two modules of this book we have introduced Linux along with an installation procedure. In the second module of this book we introduced various Linux concepts such as process management and logging systems to help you understand the kernel level implementation of all the resources while trying to crack a system. This book, is a next step to your excellent understanding of the topics that makes you a good hacker.

What does this module deal with?

In this module, we will talk about scripting by introducing two scripting languages namely python and bash scripting. Learning this can help you understand much advanced shell programming techniques in the fourth module of this book.

How to read this module?

Unlike the first two modules this is a practical experiment module. That is, without doing these scripts on your own you can never completely understand the essence of this module. Without any thoughts, install a Linux distro and other python interpreters to start utilizing the content of this book.

What are the prerequisites?

Learn a bit about Python and its history before starting this module. Also, make sure that you have pycharm and python interpreter installed in your Linux system. If not, you will face errors while compiling the programs.

Bash scripting needs just a text editor and Shell interpreter to complete its tasks.

With this introduction now we are all set to start our adventure to Scripting

world that can make us better and efficient hackers. Follow along!

CHAPTER ONE
INTRODUCTION TO PYTHON

PYTHON FOLLOWS SIMPLE SYNTAX RULES. In fact, the reason for Python's popularity among beginners is due to the fact that it is very easy to learn by someone who never heard of programming before. All it needs as some patience and a good guide to start writing small python projects. While it is easy to become a moderate programmer in Python increasing the expertise is as tough as other languages. Using modules and interlinking them in your newly written code is a tough task and can need a lot of debugging before it starts to work as you expected.

Whatever your expertise maybe in Python having a good solid knowledge about Python basics such as variables and data types can not only help you to write efficient code but will also make you save a lot of time.

First, let us start with data types in this chapter.

WHAT ARE DATA TYPES IN PYTHON?

Data types are an easy way to say to the interpreter about the data you are dealing with. Unlike traditional programming languages such as C and Java python automatically detects what data type you are using.

Confused? Let us take an example in C program.

Example :

```
Int a ;
Int b;
```

```
A = 23;
B = 46;
```

The above code is an example of C program. Here, we need to first declare variables with a datatype (int) before assigning them a value.

However, in Python all we need to do is enter the below code:

```
A = 23
B = 46
```

You can further simplify as below

```
A,b = 23,46
```

Writing in very few lines of code can improve overall performance of the program.

In the next section, we will talk about variables with examples. Follow along!

WHAT ARE VARIABLES?

Computer has memory in it which all the values are temporarily stored. All the data that we see in our computers and smartphones use the same basic philosophy.

Data changes constantly and we need to use a certain temporary entity that computer can understand to make things easier. Before the invention of variables in programming computer scientists used to waste a lot of data and that is the reason why computers used to occupy rooms in those days. Invention of variables certainly changed the way we looked at computers.

Variables are data entities that can be used to store numbers, strings or characters that can be further manipulated or modified using Python.

HOW TO REPRESENT A VARIABLE?

Variables are case sensitive and can cause problems if you do not follow these rules. Also, remember that you cannot start a variable name with a number. Python also consists of more than forty reserved keywords which cannot be used as any identifier. Some of the examples of reserved keywords are if, for, while etc,.

Here is an example :

```
first = 3
// Here ‘first’ is the variable with a value 3.
```

Note : Python automatically recognizes 3 as a value with ‘int’ data type unlike traditional programming languages which need them to declared first before using.

HOW DOES VARIABLES WORK?

It just acts like a pointer to a memory location. Whenever a variable is created a particular memory reserve will be created for that variable with the value you are giving it. And whenever the variable is called it points it to that particular memory location. Variables can be replaced and are so called as temporary values.

For example :

```
First = 3
First = 5
Print ( first)
```

Output :

```
5
```

In the above code, the variable value of 3 that is created first is replaced with 5. The replaced values will be sent to garbage and can be utilized using garbage mechanism or cache memory for complex programs. For now, it is out of scope of this book.

All the famous data types that can be used to create variables in Python are int,float,double,string, boolean and characters.

In the next section of this chapter we will talk about comments that are important to display some information about the code for reference or for others if involved in a group.

WHAT ARE COMMENTS IN PYTHON?

Comments are an easy way to give a clarity about the code that was written. At the beginning of the code multi-line comments can be used to introduce all the modules that are used along with a brief description about what is being done in the program.

Comments become very useful for beginners to understand the process flow of the code that has been written. Even for experienced programmers who are working in teams on a project these comments can help to understand the code more easily.

The Python interpreter whenever it sees a single-line comment or multi-line comment ignores them completely. They are only for information purpose.

Here is a comment:

```
“””
This is multi-line comment
This is an excellent comment
“””
I = 5
Print (i)
// This is a single-line comment
```

From the above example, you can understand that “”” can be used to create multi-line comments where as // can be used to create a single line comment.

With this, we have provided necessary basics to the Python language. In the next chapter we will discuss about other important stuff in Python that are important for you as a hacker. Follow along!

CHAPTER TWO

CONTROL STATEMENTS AND LOOPS IN PYTHON

IN THE PREVIOUS chapter we have introduced data types and variables that are the foundational blocks of the programming. In this chapter we will introduce concepts of conditionals, loops and advanced data structures with examples. Let us start!

WHAT ARE CONTROL STATEMENTS?

In programming, taking decisions is a vital role for the program to create complex programs. By using control statements we can create dynamic programs instead of the usual static ones that do not consider the user views. By using these control statements we can make the user decide what he wants to do with a program.

For example, while playing a computer game the user has the option to select whether to start the game or quit the game. When the user clicks of the “start” button the computer game code executes the game and in the other case it just quits. This is a practical example for the implementation of control statements in a computer program.

If and else are the two important blocks of the control statements. While the first executes when the condition is true the latter executes when the condition is file. Let us look at an example!

Python program code:

```
I = 3
J = 5
```

```
If i == j
Print (“ These are equal numbers”)
Else
Print (“ These are not equal”)
```

Explanation :

This example, provides a simple scenario to explain the importance of control statements in Python.

I) First, two variables are created with values, and then they are both compared using the if control statement.

ii) Here ‘==‘ stands for the equal to operator. It says that if both the values in the variables are equal then the below string should be printed on the screen.

iii) In the next block of code that is ‘else’ the print statement executes if the statement is false.

In the real world programs we can increase the complexity using nested control statements and blocks of control statements with a single execution. Control statements in Python are handy and can help us create efficient programs which can interact with user inputs.

Now, with this done we will now talk about loops that execute a statement or result again and again. Let us go!

WHAT ARE LOOPS IN PYTHON?

Loops are programming statements that are used to execute a statement again and again. There are different types of loops such as for loop, while loop and do-while loop available for python and almost every programming language. In this section, we will discuss in detail about the philosophy of loops along with a brief explanation about for and while loops with examples. Follow along!

A) For loop

For loop is the most popular way to use loops in Python. While using a for loop we check a logic and print the suggested results until the condition stops. When the condition doesn’t satisfy anymore then the program just quits without

heading over to the next step.

Here is the format:

```
for ( i =0; i > 30 ; i++)
// Here i is the variable
```

Here is the explanation :

Whenever a for loop is used ‘I’ is given by the user using the input devices. Now, when the program starts it operates by checking the input that is present. Here I>30 condition is used to satisfy the loop procedure. By using loops, we can print infinite numbers without using just one print statement.

B) While loop

While loop uses the same procedure but will print results only after the successful execution. The main difference between for and while is that for can be effectively used to print nested loops without any delay while loops can be used to take decisions while solving problems.

WHAT NEXT?

With this, we have given a complete introduction to conditional statements and loops in Python. While these topics may seem complex they are very useful for hackers who are trying to write programs using various scripting languages. Remember that understanding the goal that you want to achieve is essential for hackers. In the next section of this module we will discuss about other advanced topics. Follow along!

CHAPTER THREE
MODULES IN PYTHON

IN THE PREVIOUS chapter we have discussed about conditional statements and loops with examples. While Python syntactical structures are easy to master we need to use them in our programs using different link concepts such as modules. Modules are the primary reason why Python has become a popular language. Let us now discuss about in depth about Python modules in this chapter.

WHAT ARE MODULES?

In general terms, modules are like boxes that can be used to fit anywhere needed. When it comes to programming terminology modules are code that are independent but can be used by other projects whenever they seem necessary. They are like spare parts in a vehicle but giving a lot more capabilities than the default features.

There are usually two types of modules, namely system or standard modules and third-party modules.

A) Standard modules

Standard modules are the ones which come by default when Python is installed in the Linux system. For example, print statement that is used to display information on the screen is from the input/output module in Python. Even certain mathematical functions such as max, min work because of the Math module that comes with the Python installation by default.

However, standard modules are definitely not sufficient for creating new software because of its limiting capabilities. If you want to achieve more with

Python then you need to know about third-party modules.

B) Third-party modules

Third-party modules are modules that are usually not installed by default. You need to search and download them to your system to install them. The most common way to install Third-party modules is using a package manager such as pip.

WHAT IS PIP?

Pip is a package manager that manages all the important packages that Python offers in its repository for easy download and upgrading. To install pip in your system use the below command.

```
root@server : apt-get install python3-pip
```

This installs pip, a package manager that is light-weight and which can efficiently link all the files while installing.

HOW TO INSTALL PACKAGES FROM PIP?

Before trying to install a package do a google search about what you are expecting to achieve the module. For example, if you are a machine learning enthusiast and are looking forward to create some machine learning algorithms then you need to install Numpy , a machine learning Python module in your system.

So, to confirm that the package is correct use the following command.

```
root@server : pip3 show numpy
```

The above command will display all the information about the package that you are trying to download. Mostly the information will be about the package name, its owner name , No. Of times it has been downloaded and its creation date. Some packages will also provide the license agreement details along with the url

address of the package.

Other ways to install python modules :

Not all packages are available from package managers such as pip. Some modules needed to be downloaded from official websites or from websites such as Github.

To install Python modules from other sources use the following procedure :

1) First of all, do a complete research about the module you are trying to download. Some modules may slow down the system intentionally. All the important Python modules are available from pip. If you are looking for this option then it may be some script that was written by a programmer to do a specific task. A lot of scripts that are written in Python are usually used to automate tasks.

2) Now, note down the address the source package is available from and use wget to download it to the Python library. Wget is a Python command that is used to download any url from the web.

Here is a command :

```
root@server : wget https://github.com/sample
```

This command will download the package to the current directory.

3) Usually the downloaded python module will be in the tar package format. You need to decompress it using the following command. However, if you are not comfortable with the command line decompression you can use GUI compression tools such as gZip.

Here is the command :

```
root@server : tar -xf filename.tar.gz
```

This command will decompress the package into a separate folder in the same directory.

4) Now, after extracting the files into the directory search for a file that says ‘install.py’. All the third party modules will have this in the directory. If you confirm the existence of file then enter the following command that is given below.

Here is the command :

```
root@server : python3 install.py
```

This will install the package/module in the Linux system. After completing the process you can run the package commands in the terminal to check whether the package is installed or not.

How to use modules in your own code?

While modules are usually run as separate entities they are also famously used while creating programs. To use a particular module in your own program code you can use the ‘import’ command.

Here is the format :

```
import packagename
```

WHAT NEXT?

With this, we have completed a simple introduction to modules in Python. In the next chapter we will discussing about advanced topics such as functions and OOP with detailed examples. Let us go!

CHAPTER FOUR

FUNCTIONS AND OOP IN PYTHON

IN THE PREVIOUS chapter we have talked about modules and provided different ways to make programmers use different third-party modules in their own code. While modules help us to borrow code written by other programmers or team members , concepts such as functions and object oriented programming help us to create complex programs that can complete even complex tasks very efficiently. This chapter not only provides explanation about these concepts but will also provide real-life and coding examples to help you master the subject thoroughly. Let us go!

WHAT ARE FUNCTIONS?

If you remember correctly you should have first heard the function in elementary mathematic classes. They come in algebra lessons.

For example :

---

f(x) = 2x+3

Where x is your preferred number.

In mathematical terms, function is a mathematical component that can be used to create different values using the same definition. A lot of these functions in mathematics are used to create complex mathematical equations.

WHAT ARE FUNCTIONS IN PYTHON?

Functions are a bunch of code that can be used again and again whenever needed. All you have to do is call the function using the parameters in your program to obtain results. While the principle may seem easy, function implementation can be overwhelming especially for beginners learning Python.

There are usually two type of function . One being in-built functions and the other being custom functions.

i) In-built functions

These are the functions that are present when you install Python in your system. For example, print is a function that displays whatever is given for it in between the quotation marks. It is a prebuilt function and thus cannot be modified or changed by the end user.

ii) Custom functions

These are the functions that developers usually create using the Python modules. All the custom functions can be imported to use in other programs. However, a lot of programmers encrypt their functions while deploying software to not make them understood for protecting their logic in the code.

However, all the open-source programs without any fear publish their code because they think that sharing can help everyone if done correctly. Being an open-source developer or not is your choice. But, if you really want to the community as a responsible hackers we obviously recommended to share whatever you have create for everyone who are waiting to utilize your work.

Tip:

Almost all downloadable libraries consist functions that start with the identifier “def”. If you want to learn how functions are written then header to an open source project and analyze all the functions written and try to understand what logic the programmer have used.

Here is the format:

```
def functionname
```

Some Example functions in Python :

1) exit ()

This is a built-in function that helps to exit the program or daemon that is created using Python whenever needed. You can use it in a command line interface or can integrate in GUI based applications

2) help()

This is a python function that is used to provide information about the modules and usage of the script or program. All the information that is provided with the help() function can only be accessed by the man command in linux.

3) len()

This is a common function in Python that is used to find the length of the string. It will print the length of the string as output.

Here is an example:

```
len(“thisisgreat”)
```

Output :

```
11
```

4) max , min

These are functions that are usually used to determine the highest and lowest of the elements in a Python list.

Here is an example:

```
nice = [ 3, 5, 8]
// This is a list in Python
```

We will discuss in detail about lists and dictionaries in the next chapter. For now, think a list as something that holds items.

```
max(nice)
// This will give the output as 8
min(nice)
// This will give the output as 3
```

In the next section, we will talk about object oriented programming topics such as Classes, objects and inheritance in detail. Follow along!

WHAT IS OBJECT ORIENTED PROGRAMMING ?

Object oriented programming is a programming paradigm that is developed much later after the invention of functional and procedural programming. In functional and procedural programming, methods are created and are used whenever needed to complete tasks. In object oriented programming we used classes and objects to complete these tasks.

We are not arguing that Object oriented programming is the best because every programming paradigm has its ups and downs. The success of Object oriented programming is mainly due to its simplicity and a logical way to create classes that can be used in much larger projects. This is the reason why OOP languages such as Java and C\++ are used extensively in enterprises and businesses.

WHAT ARE OBJECTS AND CLASSES?

In simple words, Objects are programming entities that are created to replicate real world things. In real world, things have properties and perform methods with these properties. In the same way, in OOP model objects replicate these properties and methods.

Classes are the collection of objects in a sense that can help us understand its importance. Classes may not be sequential but consist of a group of objects that are similar and are thus easy to implement or use in any project.

A real world example:

Imagine boats in the real world. They are things and thus satisfy as objects in the programming world. They have engine , deck and color which are called as

attributes. These are also known as properties and are usually represented by variables and constant modifiers in the programming implementation.

On the other hand, we can use these attributes to create methods. For example, boat can drive, accelerate and lift weights using these attributes that are present. Without attribute you cannot perfectly explain the method and without methods attributes have no use.

In syntactic terms, properties are destined to be adjective that valuate the noun that is an object. All the methods are called as verbs that does a work in general.

When we collect different type of boat objects and group them then these are called classes. We can also create sub classes with more concrete evaluation to create much more efficient programs.

In python :

```
class car
// This is a class in Python
car.driving()
// This is an object calling a method in Python
```

INHERITANCE IN PYTHON CLASSES

Inheritance is an Object oriented concept by which the child classes can inherit all the properties and methods that the parent class possess. You can implement this principle in Python to create much more efficient programs. All you need to do is call the parent and child class together using a single apostrophe between them.

WHAT NEXT?

With this, we have completed a brief introduction to functions and OOP concepts in Python. As a wannabe hacker, these concepts are usually more than enough to get you started to create programs that can automate work flows and maintaining some basic tasks as a Linux system administrator. We will now talk about bash scripting in detail in the next chapter. Let us go!

CHAPTER FIVE
INTRODUCING BASH SCRIPTING

IN THE PREVIOUS chapters we have discussed about the important details and basics of python scripting in detail. In this final chapter of this module we will introduce the concepts of Bash scripting. Bash is a shell and is an essential part of the Linux environment. We will discuss the fundamentals of Shell programming in detail in the next module of the book. For now, try to understand how bash script works with the help of few examples that are provided. Let us go!

WHAT IS A SHELL IN GENERAL?

Linux uses system resources such as hardware equipment to interact with the software kernel so that everything would function in a perfect way. Linux kernel is complex and provides use case for almost every essential driver that has been created. All the communication between the user and the kernel is performed in a command line environment known as a shell.

In Linux, shell is like a crossover bridge between the user and the operating system. Whatever you enter will be sent to the Linux kernel. However, remember that Linux kernel responds to the only shell commands that seem correct to it. It will not perform any malicious self destructive commands that can shutdown the system or wipe it down completely. If you do have root privileges and are performing commands using shell environment then be sure because any wrong command can mess up your system pretty badly.

WHAT ARE THE ADVANTAGES OF SHELL?

You might have often heard about shell and should be confused why they are so

special in Linux systems. Their popularity among Sysadmins and hackers is due to the fact that they make very less clutter unlike GUI applications and installers that take away a lot of system resources. You may feel that it doesn't matter much as a normal PC user but while maintaining servers and databases allocating memory resources and constantly monitoring them to not crash is a petty task.

This is where shell comes in with its minimal system usage. It performs even complex tasks with very less code and time. Also, it should be remembered that shell can run in both Linux and UNIX operating systems. As macOS is built upon UNIX you can run all of your linux written shell code in it. For windows, there are other sandbox alternatives that create a shell environment within windows.

Some of the famous examples of shell are Korn shell, C shell and Bourne-again shell (Bash). In the next section of this chapter, we will be talking about the basics of bash shell with few examples.

HOW TO WRITE A SIMPLE BASH SCRIPT?

Bash script can be used to interact with kernel as said as before. It acts like an interpreter to execute the shell code that is written. Difference between Bash and other shell interpreters are that it has its own in-built commands that makes the execution process easy and less time-consuming.

To understand Bash better read this code:

```
#! /bin/bash
echo ( " This is a sample example")
# This is a comment
```

Explanation :

The above bash shell program that can be executed from a Linux computer consists of three important concepts of Bash that every hacker needs to understand and implement them in their programs.

I) First, we will talk about the Line that starts with a Shebang (#!) symbol. If a

text file starts with a Shebang command then it directly means that the file is asking the Linux kernel to look for the system file that can run this program as an interpreter.

Here, /bin/bash is provided as the interpreter Location. So, now Linux knows that the mentioned text file needs to be run as a Bourne-again shell.

If you want to run the program as a Python interpreter then you need to enter the command as below mentioned.

```
#! /bin/python
// This runs using the Python interpreter
```

ii) In the second line, we used echo to display a message on the computer screen. Echo is a shell built-in-program that can be used to display or print information according to the user wish.

iii) And the third line that is followed after the '#' is known as a comment. Comments are usually used by programmers to make others understand what they has written. However, it is important to remember that everything that is written in a comment will be ignored by the bash interpreter to save compiling time and system resources.

In the next section, we will know how to run a Bash program in a Linux machine.

HOW TO RUN A BASH SCRIPT?

Linus respects user privacy and is highly restricted to run potential system harming programs such as bash scripts. Even if you are the owner of the file you will not be allowed to run the bash program without certain tweaks that confirm the system that you are well aware of the risks that may happen if you mess up any system files. But that is what hackers like us do right ? Take risks, mess up the system and fix them.

Whenever you want to run any script file in Linux operating system you need to first check the permissions the file has by default.

Use the following command:

```
root@server : ls -l sample.sh
// This shows all the permissions
```

If you are unaware of the permissions in a Linux system we recommend you to head over to the second module of this book where we have discussed about Linux file permissions in detail.

The above command will display an output with all the permissions that the file possess. Generally when a file is created the owner of the file will have read(r) and write(w) permissions but not execute (x) permission. To run the bash script we need to have execute permission which we can achieve using the below command.

```
root@server : chmod +x sample.sh
// Now, the sample file gets the execute permissions
```

You, can check the permissions again using the ls -l command to confirm that you have done the procedure right. After checking, head over to the directory where file is present and enter the following command to run the bash shell script.

```
root@server : ./sample.sh
```

Here ./ represents that the designated file should be searched only in the current directory. This is a good practice it reduces the search time and is therefore recommended by all Linux system administrators to run a Bourn-again shell in a terminal.

When you run, the following output will be displayed on the computer screen :

```
This is a simple example
```

WHAT NEXT?

In this chapter, we have looked at an example that described the basic concepts of a bash shell. In the next chapter we will discuss other examples that increases the complexity that we are dealing with while writing programs that interact with the shell interface. Follow along!

CHAPTER SIX
VARIABLES IN BASH

THE PREVIOUS CHAPTER introduced Bash scripting and provided a simple example to understand the basic flow and philosophy of the shell programming and Bourne-again shell principles. In this chapter, we will reach out to the complex options that Bash will provide for Linux enthusiasts. Don't forget to experiment these written programs in your Linux machine. Practicality is different from theoretical knowledge. As a system administrator and a Linux enthusiast from years I recommend you to be gain practical knowledge first and spread all the theoretical information about it parallel to become an expert in Linux systems. Let us go!

WHAT ARE VARIABLES IN BASH SCRIPTING?

Variables hold the same principle in any programming or scripting languages. They are just a representation of storage in memory of the computer. They can be used to fill any type with data such as numbers, strings or even Boolean types.

Variables are dynamic and can be changed manually or automatically by the program. The destroyed data in the variable can be sent to cache memory or can be utilised using the garbage mechanism. But, most of the time the variable data will be destroyed with the new data when replaced. Hackers need to learn to implement variables in their scripts to exploit and crack systems.

We will give an example that will help you to understand how to use variables effectively while writing script files that can be run in a bash interface.

Here is the code :

```
#! /bin/bash
# This is an example program that takes input from the user
# and store them in variables and later display them in a string
Echo “What is Your name?”
Read name
Echo “ What is Your city?”
Read city
Echo “ Why are you here for?”
Read reason
Echo “ Hi $name, You are from $city and You are here for $reason “
# End of the bash program
```

Explanation :

The program mentioned maybe somewhat overwhelming if you are new to programming or scripting by any change. But don’t worry because we will explain each line of the program with precision in simple terms for your understanding.

i) As said before, the first line of the program which starts with a Shebang(#!) command says to the Linux machine that it needs to run the below code using the interpreter that is present in the location “/bin/bash”. If by any chance bash is not installed in the system then this location will be empty and the program will exit with an error. Remember that all Linux systems come pre installed with Bourne-again shell as it is default and recommended shell environment by Linux enthusiasts. If you are by any chance using other shell environments such as K-shell make sure that you have installed it on the system and the location of the interpreter is correct.

ii) The second line of the program comes with a comment. In the above comment the main purpose of the program is written and you can observe that you can even write multiple lines of code in a shell script file that can be run using Bourne-again shell.

iii) Now, starts the main logic of the script file we are dealing with. In this line an echo is introduced to display a question that can be used to get an input from the end user. ‘Echo’ not only displays the static information but can also be used

to interact with the user.

A message will be displayed on the computer screen as

```
What is your name?
```

And now the user will have an option to enter data because in the next line a variable is created using the ‘read’ identifier to store that data in this particular variable.

Let us assume that the user entered his name as

```
Tom
```

Now, the entered data will be read by the shell program and will be saved in the variable ‘name’. In the similar way obtain information from the end user for the next two questions that is about his city and his reason to use this program. Let us assume that the user entered this data as ‘New York’ and ‘Knowledge’.

Now, we have three variables ‘name’, ‘city’,’reason’ with data which we need to be displayed using a string. We will learn how to do that now.

iv) In the next line, we display all the stored data in the variables using the ‘$’ behind the variable name.

```
echo “ Hi $name, You are from $city and You are here for $reason “
Here name = Tom, City = New York , reason = knowledge
```

So, the displayed output will be as following :

```
Hi Tom, You are from New York and You are here for Knowledge
```

That’s it. Getting user input, storing them in variables and using them whenever you needed is the basic foundation of the scripting to interact with different

components of the target system. Experienced hackers use different complex components such as conditionals, loops and templates tp increase the complexity of the program.

WHAT NEXT?

In the next chapter, we will provide an example to help you understand the complex scripting skills such as using functions and loops to create efficient automated programs. Before heading to the next chapter, take a quick glance about operators and conditional from the Python lessons we discussed before. Shell scripting uses the same philosophy, so we need you to be aware of these topics so that we can help you with an example. Let us go!

CHAPTER SEVEN

ADVANCED BASH SCRIPTING TECHNIQUES

THE PREVIOUS CHAPTER is a tour to the much essential variables in scripting languages. Variables are often called as the building blocks of the programming because they are used in almost every part of the implementation of a program. And also, they are often easily misunderstood because of the simplicity they comes with. With the gained knowledge about variables we move farther to talk about the advanced topics that make Bash scripting a must learn for hackers. All the examples provided will help you to gain a practical knowledge on the subject. Let us go!

NMAP AND ITS IMPORTANCE

Before starting to understand the complexity of bash scripting we need you to know a bit about Nmap the tool we are using to scan the open ports. Nmap is a penetration testing tool that has been being used by hackers to understand about the target system.

What does Nmap do?

Nmap basically automates the process of scanning the ports you have provided in the command. If not with tools like Nmap, you need to manually enter the port and IP address each time and send packets to analyze whether we are receiving any response or not. It is time consuming and thus these tools are life saving for hackers.

Nmap usually works with TCP scans to analyze the target system and will give an output information about the operating system and services it is using. What we are going to do now is to automatically scan all the addresses using Nmap

with a specified open port and display them in a text file for our reference. If you are still confused look below the step by step procedure what the program does.

WHAT DOES THE SCANNER DOES?

Step 1 : We scan all the network addresses in a Local network using the Nmap Tcp scanner

Step 2 : Now we send all the output to a specified text file in a way that it can interact with the GREP format. This is essential because it is troublesome to analyze a bunch of network code with a lot of information. As a hacker we need to simplify things. We needs what we only need. That is a hackers philosophy.

Step 3 : Now using group we extract the addresses that starts with open port specification after the scanning procedure and will send then all to a new text file and will display as results for the end user.

That’s it. That’s the use case of the scanner we are going to develop now using bash script components. Follow along!

Here is the program code :

```
#! /bin/bash

# This program creates a scanner that scans a specified port number to check whether they are open or not in a local network

Nmap -sT 191.123.111.32 -p 2378

-oG resultsfile1

Cat resultsfile1 | grep open > resultsfile2

Cat resultsfile2

# End of the bash script file
```

Explanation:

The above program may seem complex for beginners bu believe us it is simple and does what it is entitled to.

I) Life every time, we start the script file with a #! Command with the location of bash interpreter. And in the next line we follow with a comment that explains

the role of the scanner that we are trying to create.

ii) In the third line, we actually get into some business by invoking a TCP scan using the nmap command. Here carefully follow what each of the command does while scanning.

A) nmap - This starts the scanning program. It is essential to scan the open ports

B) -sT - This command informs to the Nmap hat it needs to perform a TCP scan

C) 191.123.111.32 - This is the address of the local network we are trying to attack with nMap

D) -p - This says that nmap is searching for the open ports. Nmap also provides other options such as -T, -V to know about other details of the target system. Also, make sure that you are using a TOR network or other relay network with a lot of proxies to not being stopped by efficient intrusion detection systems.

E)2378 - This is the number of the port that we are trying to scan in all the network addresses in Local area network

The bash script automatically scans all the network addresses in the given network and will look out for any ports with 2378 and if they are open will print a log with ‘open’ modifier.

iii) By now, the shell terminal reaches the next line of the code a result will be generated by the nmap on command line. You can even stop displaying the output if you want to. After the scanning completes we reach the next line of the script file where it says to export all the output into a text file with GREP format enabled.

The file name that the script asks to be created is ‘resultsfile1’. If you wonder what a GREP format is let us explain you. Usually shell terminal prints a lot of code while performing actions. GREP allows us to extract a part of the output for easy reading or analyzing purposes. It is a handy tool that is often used by hackers and system administrators to search and filter things. You will learn about it in detail in the next module of this book.

iv) The next line of the bash script separates the file content that is preceded using the ‘cat’ command. After that it asks all the extracted lines to be exported to ‘resultsfile2’.

V) That’s it. Now a new file will be created with all the network addresses with

open ports 2378.

Now, enter the directory and execute it with root permissions in the manner we have learnt before. This is how you create complex shell scripts that can be run using Bourne-again shell (Bash).

WHAT NEXT?

With this, we have completed a brief introduction to the complexity the Bash interface offers. Most of the information is given by us regarding Bash scripting. Before heading out to learn about shell programming in the next module we Will explain some of the built-in commands that Bash comes with in Linux in the bonus chapter of this module. Let us go!

CHAPTER EIGHT
IN-BUILT BASH COMMANDS

IN THE PREVIOUS chapters we have talked about the building blocks of bash scripting language with few example programs and detailed explanations. In this bonus chapter we will be discussing some of the inbuilt commands Bash offers for us in Linux. You might be surprised because a lot of these commands are well known even for Linux beginners who have just started to learn about the wonders of it. Without wasting time let us start exploring now.

WHAT ARE IN-BUILT COMMANDS?

In-built commands are commands that come themself while we install the software or shell interface. For example, when we install a Python interpreter we can run the ‘py’ command in the shell. To say in other words, ‘py’ is an inbuilt command that comes because of installing Python.

Just like that, bash also comes with these commands and some of them are extensively used by the Linux operating system because of its efficiency in communicating with the kernel resources.

Some of the famous commands:

I) echo

We already discussed about this command and also used in some of our programs. Echo command is like the ‘print’ command in programming languages. All it does is to display the output on the screen. It can take variable or function instances with the help of ‘$’ symbol.

Here is an example :

```
Echo “ This can display output on computer screen”
```

Output:

```
This can display output on computer screen
```

ii) cd

This is one of the famous Linux commands that is developed by bash. All it does is to change the directory of the user in the terminal. You many need to change the directories while performing executions because some may run only in the current directory.

Here is an example:

```
Cd /bin/home
```

After you execute the above command in the shell interface you will be in that particular directory. You can now use ls to check all the files in the current directory

iii) pwd

Also remember that you can also find information about the current directory you are in using the pwd command

Here is a command :

```
root@ server : pwd
```

This will give an output with a bunch of information about the directory you are in along with number of files and the location of the directory.

iv) Process management commands

We all know that in Linux everything run as processes. There are background

and foreground processes that improves the accuracy of the system configurations that are available. Bash has some in-built commands that can help users better organize the processes that are available or processes that are just now created.

A) bg

This a built-in command that makes the software or process that you are trying to start to run in background instead of being in foreground and consuming the system resources.

Here is an example :

```
root@server : bg vmware
// This starts the software in background
```

As a hacker you need to be aware which software are running in the background as most anti virus software work as background processes. Also, when you plant an exploit in the target system you need to make sure that it will be not be found out by the system administrator.

B) exec

This is a built in bash command that starts a software as a new process. Sometimes even when a software or utility is opened you can use this command to open a new instance of the software.

Here is the command :

```
root@server : exec vmware config
// This starts the config file of Vmware software as a new process
```

You can confirm whether or not a new process had been created using the following command

```
root@server : pid
```

Check for vmware in this using the grep command to see two processes running in the same time

C) wait

Usually processes are started automatically and are ended whenever they are feasible for the system resources. Sometimes processes even get stuck and end before they needs to. Sometimes, you can force wait the software to wait for a process to be completed using the wait command

Here is an example :

```
root@server : wait vmware
```

V) umask

This in-built command is used to change the default permissions that are already present. We can usually use chmod to give execute permissions but umask is much more complex and can be used to perform complex changes whenever needed.

Here is a command :

```
root@server : umask 777 bash.sh

// This makes the files as executable even though by default it has just read and write permissions.
```

Vi) In built command for variables

Variables are memory locations that are used by programming languages to point to the memory address of computer. They are simple and are needed for the efficient functioning of scripts. Bash provides two simple commands for tweaking their possibilities.

A) read-only

This bash command make the variable not change . Usually variables are temporary but when they are given this particular bash command their values cannot be changed.

Here is a command :

```
readonly first
```

B) export

Also, whenever you create variables or functions in computer programs or in scripting files they cannot be used by other files. Only way to use these components in other files is by using the export command that is available in bash.

Here is a command :

```
export first
```

WHAT NEXT?

With this, we have completed a brief introduction to Bash scripting in Linux. All the examples should have helped you understand the foundations of scripting. With that in mind, now you are ready to learn in much detail about the shell programming that Linux runs upon in the next module. Follow along and experiment with different complex topics to interact with the system resources that Linux is capable of. Let us go into an another exciting module of this book.

SHELL PROGRAMMING FOR HACKERS

INTRODUCTION

WELCOME to the fourth module of this Linux for hackers booklet which is designed to help beginners interested in hacking to achieve essential skills using various techniques. In the previous module we have already introduced a bit about bash scripting, which is a shell interpreter. We hope you will have fun reading the advanced skills that a shell programmer need to master.

Also, In the previous modules of this book bundle we coherently discussed about the fundamentals of Linux with various in-deep examples. In this module we will talk about shell scripting along with few tips and tricks that can help you create well versed programs.

WHY DO HACKERS NEED SHELL PROGRAMMING?

Hackers especially need shell programming as an expertise to crack target systems. While bash and Python scripting is an essential prerequisite for hackers, shell programming is often neglected because of the fact that it can coherently and fundamentally be used among Linux systems instead of the much popular Windows systems.

As a front-line Linux expert lecturer with more than ten years industry experience I , the author of this book has summed up my years of teaching and practical experience as the essence for this book.This book not only explains various grammar and functions of Shell, but also contains a large number of interesting cases, which are accumulated by my philosophy of teaching and have great reference value for any scope of the reader.

At present, there are many IT and Linux books on the market, but many of them are hard to read or works that are rushed to completion, which may be lacking in

content professionalism and coherent writing style, and even can cause more confusion. This book as far as we know provides consistent writing that can help programmers and Linux enthusiasts attain knowledge all the while without any semantic disturbances. In the previous modules we have already discussed about various concepts of Linux that are essential for hackers.

Nowadays, well-known IT books all come from the author's long-term research and thinking in this major. Fortunately, this Linux Programming Guide belongs to this kind of books that enrich the author's experience, which is why we solemnly recommend this book to any Linux enthusiast of any expertise.In today's IT field, it is really important to master automatic operation and maintenance skills. For this exact purpose learning shell scripting is a definite must.

WHY SHELL SCRIPTING IS IMPORTANT?

Regardless of the basic Linux or cloud platform used, the operation and maintenance, development and testing personnel all use DevOps to guide and carry out their work. Various automatic operations and maintenance tools such as Python, Perl and Puppet are constantly emerging. Linux is used by tons of companies to implement their servers and other IT infrastructure. Even though being 0pen-source software Linux holds a good chunk of the IT market and offers reliable support from hundreds of thousands of developers looking forward to help others.

Shell can combine every dedicated and efficient task command in Linux to complete complex and wonderful transactions.Every Linux engineer and student understands the importance of scripts, especially in this era when there is a growing demand for automation operation and maintenance development engineers.

Shell can be simple and efficient.Similar to all programming languages, if you want to master the essence of Shell programming, you should be familiar with various command parameters in Linux, be diligent in practice, and refer to code examples written by masters of the technology.On the basis of reading this book, beginners can first simulate the example case code, then reproduce it through memory, and finally draw inferences from others.

Experienced engineers can directly resonate and get inspiration from this book.I believe that every reader can find surprises in the module of this book.I hope

everyone can love Shell programming, Linux and open source. Open-source not only moves technology forward but also helps us getting rid of being controlled by greedy and selfish multi-national companies.

WHY SHELL BECAME AN IMPORTANT PART OF OPEN-SOURCE DEVELOPMENT?

Computer technology has both profound theory and strong practicality. Many related operations must be experimented by themselves, and even after many failures, they can achieve their ideal goals.The Shell involved in this book is an old and young technology that is especially developed by open-source programmers for open-source systems.

From the initial stage of UNIX and Linux use, the Shell is accompanied by users.Nowadays, when IT market is full of new terms such as digital transformation, Shell script still plays a vital role in many aspects.This book is characterized by its simplicity and emphasis on practicality and examples.

Many people who have just started to learn Shell script programming, after learning the basic grammar, have no idea to write scripts because of lack of script cases, and many people give up before starting real programming, which is also the defect of other similar books on the market at present.

The highlight of this book is that it not only explains the syntax format of Shell, but also enables readers to verify their knowledge through a large number of case scripts, and will build the idea of writing scripts, which is commendable.

Before Going to the real part…….

In today's intelligent data era, automation and intelligence have become the inevitable choice for enterprises, whether it is for the improvement of efficiency or the operation and maintenance of large-scale systems.Shell script has become one of the necessary skills for every engineer.

Beginners (novices) can systematically learn and master from this book about how to standardize the preparation and use of Shell scripts, and how to draw inferences from actual combat cases through existing knowledge points, and apply them to the production environment with less detours.

For the old experienced shell wannabes, this book systematically expounds the knowledge points of Shell and a large number of actual combat cases, which can

help you get new inspiration and guidance, and can make you finish your work more efficiently, intelligently and automatically. It is a rare reference book worth reading frequently.

HOW TO USE THIS MODULE?

When choosing the operating system distribution, this book integrated the characteristics of each distribution, and finally chose LinuxMint as the basic system platform of this book. Linux mint is one of many Linux distributions, but because it comes from Debian framework and is completely open source, including open software YUM source, it can bring more convenient upgrade methods for users.In addition, many domestic enterprises are also very keen on LinuxMint distribution, which also increases the practicality of this book.

CHAPTER ONE
INTRODUCING SHELL

THIS CHAPTER IS the beginning of this module and will help to introduce you the basics of shell scripting files along with various advantages they come with. We recommend you to thoroughly read this chapter before skipping to the other ones. Also, it is recommended to experiment the given code samples in your Linux machine. There is no better way to learn Linux more than doing yourselves in a Linux system. Do a simple research about different Linux distros and settle with the one that you feel most comfortable with. From a simple Arch Linux system to a complex and visually stunning Debian based Linux distro you have everything to chose from.

WHAT IS SHELL?

Shell is a very easy and powerful programming language. Many Linux system maintainers often use Shell scripts in their work, but not everyone is good at writing Shell scripts. Once you master the rules and skills of writing Shell scripts, your work will be easier and more efficient in the future!

Since 1991, Linux has rapidly grown into the preferred operating system for enterprise server products, and more and more IT enterprises have adopted Linux as their server platform operating system to provide customers with high-performance and high-availability business services. Linux uses Shell programming and is therefore a must for enthusiasts who are willing to make a career in Linux server and database administration. Ethical hackers and application developers also use shell for their own purposes.

WRITING FORMAT OF SCRIPT FILE

Before talking about the importance of Shell scripting and how they work it is better if you have a good grasp of the format of a Shell scripting file. This is why we are introducing it in this chapter with detailed examples.

What is a Shell script file?

To briefly explain in layman terms, the commands of Linux or UNIX-like system are written into a file, which is popularly known as a Shell script file. It is written in a way that the Shell script file we write must run in Linux or UNIX-like operating systems. There are also a ton of tweaks to run shell files in windows or macOS operating systems.

Note:

Before starting the fundamentals we remind you that the operating system platform used in this book is Linux Mint. Linux mint is not only a Debian based light-weight operating system but also can perform tons of complex operations.

After giving the script file the necessary execution authority (most commonly called as root privileges) and running the script file, the computer will execute the commands in the script file content from top to bottom. While running it can look in other shell files or programs according to the instructions. Shell files are an easy and recommended way to automate operations that can define your work.

Compared with manually executing system commands on the command line, the advantage of script file is that once written, all commands in script file can be automatically completed later (with higher efficiency).Moreover, the same script file can be called and executed repeatedly, avoiding unnecessary manual and repeated input of commands.

As understood by the previous section, script is a file. So what tools do we use to create this file?

It needs to be understood by a Linux enthusiast that a script file is just an ordinary text file. For this reason you can create a script file by using any text editor software such as vim, gedit, Emacs, Notepad\++, Sublime, Atom and other tools. You can do a simple research to find out the text editors for your use case.

We use VIM editor in the cases in the following chapters.It is recommended to use .sh as the file extension when creating a new file, so that people can see that

the file is a Shell script file at an instance. If you are worried you can first just create the program code in a .txt extension and afterwards can export into a .sh extension simply by renaming the file. Until you change the file extension and provide execution permissions the script file cannot be run.

Here is the command:

```
root@server : vim sample.sh

// This will create a shell file
```

WHAT ARE THE WRITING FORMAT REQUIREMENTS FOR SCRIPT FILES?

First, the first line of the script file requires shebang(#!) symbol which specifies an interpreter for a script, such as #!/bin/bash, #/bin/sh, # ! /usr/bin/env python, etc.

This line is annotated by #, so it will not be executed as a command, but the computer knows what interpreter should be used to interpret all valid codes in the whole script file through this annotation information (the interpreter used in this case is /bin/bash).

Secondly, the script file uses # or < < symbol to realize single-line or multi-line annotation. It needs to be understood that the annotated keywords or codes will not be executed, and the annotation is mainly for people to see!

By reading the notes, we can quickly understand the function, version and author contact information of the script file. The core function is to explain the function of the script file or code block.Finally, the most important content is the code part.

Generally, a line of code is a command, and all valid code commands in the script file are executed from top to bottom.

Let's write the first script file and look at the composition of the script file.

```
#!/bin/bash

<< COMMENT
```

```
This is a comment
( Mostly multi-line)
Written for Linux for Hackers
COMMENT
# Now we will show an echo command
echo “ This is an example program”
```

Note that the keyword behind the < < symbol can be any string, but the same keyword must be used when ending the comment.If you start commenting from < < ABC, you must also use ABC (letters are case sensitive) when you finish commenting information.

In the next section of this chapter, we will have a brief discussion about the various execution modes of shell or for that matter any script files.

VARIOUS EXECUTION MODES OF SCRIPT FILE

After the script file is written, the next step is to execute it.There are many ways to execute script files, including those that require execution permission, those that do not, those that open subprocesses, and those that do not. We will explain each of these instances with examples.

1) If the script file itself has no executable permission

In this case , the default script cannot be executed directly, but an interpreter like bash or sh can use the script file as a parameter (read the contents of the script file) to execute the script file.

Here are the commands:

```
root@server : ./sample.sh
// Will display errors because there is no permission
root@server : bash sample.sh
root@server : sh sample.sh
// Other main ways to execute but without errors
```

From the output information of the above three commands, we can see that when the sample.sh script file under./(current directory) is executed without execution permission, an error message will appear, while when bash and sh are used to execute the sample.sh script as parameters, the correct message “This is a sample” will be displayed as output.

2) If the script file has an executable permission

The execution permission can be assigned to the script file through the chmod command. Once the script file has the execution permission, it can be executed using absolute path or relative path.

The following example assumes that a script file has the absolute path of /root/sample.sh, and the effect of executing the script file is as follows.

Here are the commands :

```
root@server : chmod +x sample.sh
// This will give executable permissions
root@server : ./sample.sh
// Will display output without errors
```

3) How to open a sub-process for execution

Regarding to make a descision whether to open or not a sub-process we must first understand what a sub-process is. Generally, we can view the process tree through the pstree command to understand the relationship between processes.

Here is the command:

```
root@sample : pstree
```

From the above command output, we can see that the first process started by the computer is systemd, and then N subprocesses are started under this process, such as NetworkManager, atd, chronyd and sshd, all of which are systemd subprocesses.Under the sshd process, there are two sub-processes of sshd. Under the two sub-processes of sshd, the bash interpreter sub-process is started, and a

pstree command is executed under one of the bash processes.

Your system may have different other sub processes. All you need to do is verify using the pstree command. As we said just now, whether the script is executed directly or by using an interpreter such as bash or sh, the subprocesses will be started.

The following example demonstrates the effect through a script file.

First, open a command terminal, write a script file in the command terminal, and execute the script file.Then, open a command terminal, and observe the process tree through pstree command in this terminal.

Here are commands:

```
// sleep.sh
#!/bin/bash
Sleep 5000
root@server :chmod +x sleep.sh
rooot@server : ./sleep.sh
root@server : pstree
```

It can be seen from the output that a subprocess script file is opened under the bash terminal, and a sleep command is executed through the script file.Back to the first terminal, use Ctrl+C to terminate the script file executed before, and use bash command to execute the script again.

```
root@server : bash sleep.sh
```

At last, use the pstree command on the second terminal to observe the experimental results.The result is similar, a bash subprocess is opened under the bash process, and a sleep command is executed under the bash subprocess.

4) Execution mode without opening subprocesses

Next, let's take a look at the case of execution mode without opening subprocesses.

Similar to the previous experiment, we need to open two command terminals.First, open the first terminal, and this time use the source or .(dot) command to execute the script file.

Or then, we can open the second terminal and observe the results through the pstree command.It can be seen from the experimental results that the sleep command in the script file is directly executed under the bash terminal.Finally, we can write a special script file with the following contents.For this script file, use open subprocess and do not open subprocess respectively.

Here are the command:

```
root@server : . sleep.sh
root@server : bash sleep.sh
root@server : source sleep.sh
```

With this, we have completed a brief introduction about the shell scripting execution capabilities. As linux system deals with lot of users and subgroups it is important to not run a script file that is not supposed to run. So, with these extra abilities to run programs we can now start learning about input and output statements in shell programming. Let us go!

CHAPTER TWO

INPUT AND OUTPUT IN SHELL PROGRAMMING

IN THE PREVIOUS CHAPTER, we gave a small introduction to shell scripting and talked about execution permissions with various scenarios. In this chapter we will provide in-depth information about input and output statements and their importance in shell scripting.

HOW TO IMPLEMENT THE PROCESS OF INPUT AND OUTPUT OF DATA IN SCRIPT FILES?

In Linux system, both echo command and printf command can implement the function of information output that is necessary for the program. let's look at the application cases of these two commands respectively.

1) First, we will use echo command to create a script file menu

Function description of the command:

echo command is mainly used to display string information.

The syntax of the echo command is as follows:

```
echo “ This is the 1st sentence”
echo “ This is the 2nd sentence”
echo “ This is the 3rd sentence”
```

As can be seen from the above script file, the echo command can output any string of messages. Multiple echo commands can be used to output multiple

messages, or one echo command can be used to output multiple messages together with quotation marks.However, the output information is usually in default black font and cannot be displayed in the center.

When we need to display the information prominently to prompt the user's attention, the output may be slightly monotonous.The echo command supports the -e option, which allows the echo command to recognize the meaning of the escape symbol after \t.

```
root@server : echo “\t”
```

Among these additional commands, \033 or \e can be followed by terminal code and can be used to define font color, background color, positioning cursor, etc. All these extra information can enhance how you display messages according to your convenience using the echo command.

Example Application case:

Here is a certain way to display the classic “hello world” using various advanced shell output techniques.

Because there is no -e option and the character \ is not supported, the screen will output the original content \t directly. For this example, we will Output hello, then use Tab indent, and finally output world, and then will make sure that the the final result will be displayed as “hello world”.

Also in an another way we can Output hello, then move the cursor to the left by one bit, and then output world. The original letter e is usually replaced by the new letter o, so the final output result will still be displayed as “hello world”.

Here are the commands:

```
root@server : echo -e “hello\fworld”
root@server : echo -e “hello\tworld”
```

Note:

There is at least one space between the -e option and the content to be given as

output later. Similar to the above case, move 2 bits to the left, and the final output will be shown as “hello world”.

```
root@server : echo -e “hello/bo world”
```

We can also output hello and wrap the line but the cursor still stays at the original position, that is, the position behind the letter o, and then will output world.We will have Bold display OK. \033 or \e can be followed by different codes to set different terminal attributes.

```
root@server : echo -e “hello \bo world”
```

1m is for the terminal to display the character string in bold, followed by OK is the content of the character string to be displayed, and finally \ 033. [0m is to close the terminal attribute setting after outputting OK in bold.]

If the attribute setting is not closed with 0m in the end, then all strings in the terminal are displayed in bold.After executing the following command, you will find that all strings that output in the terminal are displayed in bold except OK.

Here is a command:

```
root@server : echo -e “\w 0m”
// This displays with the specified options
```

Try it:

Do you have any other colors? Such as 92m? You can try it yourself!

In addition to defining the font color, style and background of the terminal, you can also use h to define the location attribute.For example, you can display OK in the third row and tenth column of the screen by the following command.

Here is the command:

```
root@server : echo -e “\033”
```

Finally, we use the echo command to write a more interesting script file menu. In the following script file, clear command is used to clear the whole screen, and then echo command is used to set terminal properties, and a personalized menu with color and layout is printed. As for the specific color matching, readers can personalize the design according to their own needs.

Here is the script:

```
#!/bin/bash
clear
echo -e “\033m”
```

With a complete introduction to the much discussed “echo” command we will now shift our focus to the “printf” command in Linux.

2) Expand knowledge and create a script menu by using printf command.

In Linux system, besides echo command, you can also use printf command to achieve the same effect.

Function description:

printf command can print format data.

The syntax format of the printf command is as follows:

```
printf “details”
```

Note: The parameters of the general printf command are what needs to be used as an output.Commonly used format strings and function descriptions are shown for your better understanding of what you are dealing with. In the next section, we will further expand our knowledge using an application case.

Application Case:

The format %8d of this command sets the print width to 8, and displays the integer 64 in a right-aligned manner.

```
root@server : printf “%8d” 64
// Output will be 64 with width as 8
```

Note that there are five spaces in front of the output information 64 of this command.5 spaces +3 numbers together are 8 characters wide.If you need left alignment, you can use %-8d to achieve the effect, such as the following command.

root@server : printf “%-8d” 64

Note, after the printf command outputs information, it defaults to no new lines. You can use the \n command if you need to wrap lines.

In order to better observe the left-right alignment effect, the following example will print two symbols to determine the position.

root@server : printf “ | %-11d| \n” 64

Left-aligned output 64, the output content takes up 11 characters wide, 12 takes up 2 characters wide, followed by 8 spaces.

The default printf command will not wrap after the output, but can wrap after the output by using \n the command symbol.

This displays the octal value of 10, and the conversion from octal 12 to decimal is exactly 10.0x11 represents hexadecimal 11, and the printf command converts hexadecimal 11 into decimal integer output (17).011 represents octal 11, and printf command converts octal 11 into decimal integer output (9).

```
root@server : printf “%d \n” 17.011
```

When using \d to print a large integer, the system prompts that it is out of range and that the maximum number that can be printed is 92798329882.

If you need to print such a large integer, you need to use the \u command, but the \u command also has the maximum display value (29290390238009), and when it is greater than the maximum value, it cannot be printed.

```
root@server : printf “%d\n” 9279832988
```

In the next section, we will talk about reading the information available using the ‘read’ command that is famous now a days.

3) Read the user's input information by using the read command.

Before that, we learned how to output data in the Shell script, and then discussed how to solve the input problem. In the Shell script, read command is allowed to realize the data input function.

Function description:

The read command can read a line of data from standard input.

The read command has the following syntax format.

```
root@server : read \{ Enter parameter here }
```

If no variable name is specified, the default variable name is REPLY.

Here, 672 is entered through keyboard, while Read command reads this 672 from standard input and assigns this string to variable key1. For key1, we can use echo $key1 to display the value of this variable.

```
root@server : read something
672
```

Application case:

Note that after the password is prompted here, when the user enters the password 672, the computer displays the plaintext of the password on the screen, which is not what we want to see!

What to do?

The read command supports the -s option, which can make any data input by the user not displayed, but the read command can still read the data input by the

user, but the data is not displayed.Let's look at a script case written with the read command.

Here is the command:

```
read -s -p “ Use this to save”
```

This script reads the user name and password entered by the user through the read command, and when reading the password entered by the user, the content of the password is not directly displayed on the screen, which is safer.

The user name and password entered by the user are stored in two variables, user and pass respectively. Here, call the values in the variables with $,use the useradd command to create a system account, and use the passwd command to configure the password for the user. Passwd is directly used to modify the password.

Here is the script:

```
#! /bin/bash
# Read the data
read -p “ Enter the name “ user
read -s “ Enter the password” pass
user add “$something”
#print password
```

By default, the password is configured by man-machine interaction, which requires manual input and repeated input twice.Here, we use a | symbol, which is like a pipeline. Its function is to pass the output result of the previous command to the next command through the pipeline as the input of the latter command.

```
echo “$pass” |passwd —stdin “$something”
```

WHAT ARE COMMAND PIPELINES?

Sometimes, in Linux system, we need to complete a complex task, but a certain command may not be able to complete this task. At this time, we need to combine two or more commands together to complete such a task.

As said from the Linux documentation, similar to the pipeline for transporting water, the pipeline of Linux system can store the output result (data) of command 1 into the pipeline, and then let command 2 read the data from the pipeline and further process the data.

In the next section, we will talk about the advanced abilities of giving output and input to the functions that are available.

a) who

The who command can help us check which accounts log in to the computer at what time.However, when there is a lot of login information on the computer, it is inconvenient to record the number of logins manually.

The wc command in Linux system can count the number of rows, but the wc command needs data. If you give wc several rows of data, this command can automatically count the number of rows of data.We can use pipelines to combine who and wc commands.

```
root@server : who
```

b) ss

For another example, ss command can view the list of ports that all services in Linux system listen to.However, ss command itself has no flexible filtering function, while grep command has powerful and flexible filtering function, so these two commands can be used together through pipeline.Obviously, there are a lot of data that are not filtered by grep command, which is not clear enough.

After ss command stores its output data into the pipeline, grep command reads the data from the pipeline again, and filters out data lines containing sshd from many data, and the final output result is only two lines of data.

```
root@server : ss
```

In a conclusion….

In this way, we can see the data we need more simply and clearly.Some commands are special, such as the passwd command we used earlier, which is used to modify the system account password, but the command can only read the password from the keyboard by default.

If you want the command to read the data from the pipeline and use it as the password, you need to use the --stdin option.As shown, the echo command will display the output results on the screen by default. After having a pipeline, the echo command can store the output 123456 in the pipeline, and passwd can read 123456 from the pipeline to modify the password of the system account ‘sample’.

In the next section of this module, we will discuss about the advanced topics that are related to input and output statements in shell scripting. Follow along!

CHAPTER THREE

REDIRECTING INPUT AND OUTPUT IN SHELL

IN THE PREVIOUS chapter we gave a satisfactory introduction to the importance and implementation of input and output statements in shell scripting. In this chapter we will talk about advanced expertise details such as Redirection of input and output statements with sufficient examples. Follow along!

WHY IS REDIRECTION IMPORTANT IN SHELL?

In most systems, the output information is usually displayed on the screen by default, while the standard input information is obtained through the keyboard.

However, when writing scripts, we can't or don't want the output information of some commands to be displayed on the screen (when the scripts are executed, a large amount of output information will make users feel confused). It also uses system resources more than what we ought to.

At this time and for this reason, it is better to write the output information into the file temporarily, and then read the file and extract the required information when needed later.There are similar problems with the default standard input information. When we use the mail command to send mail in Linux system, the program needs to read the text of the mail.

By default, the input data from the keyboard is used as the text, which will make the script enter interactive mode, because reading the keyboard information requires manual input.

For example, if at this time the default input mode can be changed, instead of reading the data from the keyboard, the data can be read from the file prepared in advance, so that the mail program can automatically read the file content and

send the mail automatically when needed without manual interaction.In this way, the automation effect of the script will be a better experience.

OUTPUT IN LINUX SYSTEMS

In Linux system, the output can be divided into standard output and standard error output.The file descriptor of standard output is 1, and that of standard error output is 2. While the standard input file descriptor holds for 0.

If you want to change the direction of the output information, you can use the > or > > symbol to redirect the output information to a file. Use 1 > or 1 > > to redirect standard output information to a file (1 can be ignored without writing, the default value is 1), or use 2 > or 2 > > to redirect wrong output information to a file.

Here, use the > symbol to redirect the output information to the file. If the file does not exist, the system will automatically create the file. If the file already exists, the system will overwrite all the contents of the file (the original data will be lost!).

Use the > > symbol to redirect the output information to the file. If the file does not exist, the system will automatically create the file. If the file already exists, the system will append the output information to the end of the original information of the file.

In the following example, the echo command would have displayed the data output on the screen, but if redirection is used, the output information can be exported to a file.

```
root@server : echo “ This is a sample” > sample.txt

// This exports the file to a text file named sample
```

The previous echo command will not give an error message.However, when using ls command, the final output information is divided into standard output and error output according to whether the file exists or not. At this time, if we only use > or > >, we cannot redirect and export the error information to a file.

Here we need to use 2 > or 2 > > to redirect the error output.

If a command has both standard output (correct output) and error output, how to redirect it?

In fact, we can redirect standard output and error output to different files, or redirect them to the same file at the same time. Use the &> symbol to redirect both standard output and error output to one file (overwrite), or use the &> > symbol to achieve additional redirection.

Finally, we can also use 2 >&1 to redirect error output to standard correct output, or 1 >&2 to redirect standard correct output to error output.Although the following commands all show the results on the screen.Although the first command is an error message, it is displayed on the screen from the standard correct channel.

```
root@server : ls -l /something >2 sample.txt
```

While the second command has no error message originally, the final hello is displayed on the screen through the error output channel by redirecting the correct information to the error output.

Under normal circumstances, because the system does not have a /something file, the ls command will report an error, and the error information will be transmitted to the display through the error output channel.

```
root@server : ls -l /etc/home > sample.txt 2> err.txt
```

However, when we use the 2 >&1 command, the error message will be redirected to the standard correct output. Although the screen will eventually display the error message, it is transmitted to the monitor through the standard output channel.

Under normal circumstances, the echo command displays messages on the screen through standard output.When we use 1 >&2, the system will redirect the correct output information to the wrong output, although the screen will eventually. Hello is also displayed, but it is transmitted to the monitor through the wrong output channel.Finally, the correct and wrong information is imported into the file.

Note:

There is a special device /dev/null in Linux system, which is like a black hole.

```
root@server : echo “sample” > /dev/null
```

No matter how much data is written in this file, it will be swallowed and discarded by the system.If there is some output information that we no longer need, we can use redirection to import the output information into the device file.

Note:

Once the data is imported into the black hole, it will not be retrieved.In addition to redirecting the output, you can also redirect the input.

The default standard input is keyboard and mouse.But the keyboard needs human interaction to complete the input.

For example, the following mail command, after executing the command, the program will enter the state of waiting for the user to input the mail content. As long as the user does not input the content and uses an independent line to indicate the end of the mail content, the mail program will stay in this state.

```
root@server : mail -s Thisislatest
```

All the above email texts need to be manually entered, but in the future, when we need to use scripts to send emails automatically, there will be problems.

To solve this problem, we can use the < symbol for input redirection.< the symbol needs to be followed by a file name, so that the program can read the data from the file instead of reading the input data from the keyboard.

```
root@server : mail -s < /etc/home
// This is an input redirection
```

If we want to send emails automatically and interactively without preparing files

in advance, can we?

You can use the << symbol to achieve the same effect.In this way, the script can run independently without depending on the file of the mail content.Use the << symbol to redirect data content to the previous command as the input of the command.

```
root@server : mail -s warning << EOF
```

The << symbol (also called Here Document) means that the content you need is here.Let's take a look at an example where cat reads data through Here Document and then exports the data to a file through output redirection.

Here is the script:

```
#! /bin/bash
Mail -s error
This is where we want to send
<< fax.txt
# Ends the program after an input redirection
```

In Linux system, fdisk command is often used to partition disks, but this command is interactive, and now we need to write scripts to realize automatic partition, automatic format, automatic mount of partitions and so on.To solve this problem, Here Document can also be used.Let's write such an automatic partition script.

Warning:

This script will delete all data on the disk, and all data will be lost!

Here is the script:

```
#! /bin/bash
mkfs.xfs /dev/sdb
# This is used to format
```

```
EOF
mount -a
```

In order to improve the readability of the code when writing scripts, it is often necessary to add extra indents to the code.However, when using << to import data into a program, if there is indentation in the content, it will be passed to the program along with the indented content.

At this time, the Tab key only serves as indentation, and we don't want to pass it to the program.If necessary, you can use the <<-symbol to redirect the input, so that the system will ignore all data contents and Tab key in front of EOF.In this way, only the Tab key can be ignored, and if the body content of Here Document is indented with spaces, it is invalid.

```
cat << EOF
```

With this, we have completed a brief introduction to the redirection of input and output statements in Shell programming with the help of linux system commands. Before proceeding to discuss about variables and other advanced concepts we will talk about the posture of various equation marks while dealing with shell scripts. This may usually cause mistakes even being a steep learning curve. Follow along!

CHAPTER FOUR

DIFFERENT QUOTATION MARKS FOR SHELL SCRIPTING

PROGRAMMING LANGUAGES often use quotation marks to display the end of the line in a program or will be used to determine the indentation principles of the programming language. In shell, we also use different type of quotation marks to improve the efficiency of the program. In this chapter, we will describe some of these using examples. Follow along!

CORRECT USE POSTURE OF VARIOUS QUOTATION MARKS

1) single quotation marks and double quotation marks

We often need quotation marks when writing scripts. Shell supports various quotation marks, such as ""(double quotation marks)," ‘(single quotation marks)’, (reverse quotation marks) and \ (escape symbols).

Here are some commands :

```
root@server : touch x y z
// This is without quotation marks
root@server : touch “ x y z “
// This is with quotation marks
```

Under what circumstances are so many symbols used?

Let's look at a few cases.It can be seen here that the function of double quotation marks is to quote a whole, and the computer will treat all the contents in

quotation marks as a whole.Instead of using double quotes, three different files are created.When files need to be deleted later, similar problems will occur. So, we need to use the first command instead of the second command as they are individual files.

```
root@server : rm -rf x y z
// This will delete all the three files created
root@server : rm x y z
// This will display an error
```

Also we should analyse how many files are there here?What exactly is the file name?

```
root@server : rm “ x y z”
```

Because double quotation marks are not used here, the system understands that three files X, Y and Z need to be deleted, but in fact, there is only one file named “x y z” in the system, and the space is also a part of the file name.

In the next section, we will further discuss about how files can be deleted in practical terms.

HOW FILES CAN BE DELETED?

The files usually are successfully deleted by using double quotation marks.In Linux system, besides double quotation marks, single quotation marks can also be used to quote a whole. At the same time, single quotation marks have another function, that is, special symbols can be shielded (the special meaning of special symbols can be shielded and converted into the name of the character surface).

```
root@server : echo #
root@server : echo ‘#####’
```

The above two commands have no special symbols, so the use of double quotation marks or single quotation marks has the same effect.However, when there are special symbols, single quotation marks and double quotation marks cannot be interchanged, such as the following example explained above.

In the first Shell, the # symbol has a special meaning and is a comment symbol.

The # symbol and the content behind the # symbol will be interpreted as comments by the program and will not be executed. This command originally wanted to output a # symbol on the screen, but the actual output result is a blank line.

If we want to output this # symbol, we can use single quotation marks as in the second command to mask the special meaning of the # symbol.In addition, in the Shell, the symbol $ has the special meaning of extracting variable values, and when we need to use the symbol $ directly, we also need to use the shielding function of single quotation marks.

Actually, in Linux, besides single quotation marks, there is also the \ symbol. Although the \ symbol can also realize the function of shielding escape, the \ symbol can only escape the first symbol behind it, while single quotation marks can shield all special symbols in quotation marks, as shown below.

```
root@server : echo $$
root@server : echo '$$'
```

2) Command Substitution

Finally, we will explain the symbol (back quotation mark). Back quotation mark is a command substitution symbol, which can replace the command with the output result of the command.

Let's take an example below.

```
root@server : /root/name.tgz /var/log
```

Use the above command to back up all the data in the /var/log directory to the /root directory, but the file name of the backup is fixed.If the system needs to

perform the scheduled tasks, the data will be backed up once every Friday, and then the new backup will overwrite the original backup file (because the file name is fixed).In the end, you will find that only the last week's data was backed up, and all the previous data was lost.

How to solve this bizarre problem?

This command still uses the tar command for backup.However, because the \ symbol is used to replace the command, the file name backed up here is no longer the date, but the output result after the date command is executed, that is, the string of the date command itself is replaced by the output result of the command, and the last file name backed up is similar to the log-name.tar.gz.

The specific time in the file name depends on the computer system time when the command is executed.Look at a few more examples.Although back quotation marks are easy to use, they also have their own defects, such as being easily confused with single quotation marks and not supporting nesting (back quotation marks cannot be used in back quotation marks).

In order to solve these problems, people have designed the $ () combination symbol, which is also a command replacement function and supports nesting function, as shown in the following case.

```
root@server : ping -c2 $(url)
```

With this, we have completed a brief introduction to the basics of shell programming with definite examples. In the next section of this module we will talk about variables in detail. Follow along!

CHAPTER FIVE

VARIABLES IN SHELL PROGRAMMING

IN THIS CHAPTER we will talk about the efficient implementation of variables while writing shell programs. They may seem easy but are usually complex and can change the shell implementations tremendously. We will also discuss about expressions in shell programming with detailed examples. Follow along!

WHAT ARE VARIABLES?

Let us introduce variables with a small analogy. We all know that water is stagnant water if it does not flow. If the constants used in the script are all immutable, then the function of the script is not flexible enough, and it is only a fixed script that can meet specific needs.If the water flows, there will be various forms. In the same way, if the script uses variables, it will become more flexible and changeable.

Just as air temperature and air pressure are real-time changing data in real life, the computer data that scripts need to deal with often changes in real time.In Linux system, variables are divided into system preset variables and user-defined variables.

CUSTOM VARIABLES

First, let's look at how custom variables are defined and called.In Linux system, the definition format of a custom variable is “variable name = variable value”, and the variable name is only used to find an identifier of the variable value, and it has no other function.

When defining a variable, the variable name can only use a combination of letters (both upper and lower case), numbers and underlined lines (_). And can not start with numbers.In addition, it is best to use easy-to-understand words when defining variable names in your work. Remember not to use random characters to name variables. Irregular variable names will make the readability of scripts extremely poor.

It should be noted that there should be no spaces on both sides of the equal sign when defining variables.

Tips for variable names:

When you need to read the variable value, you need to add a dollar sign "$" before the variable name. When variable names are mixed with other characters that are not variable names, they need to be separated by {}.Finally, if you need to undefine a variable, you can use unset command to delete it.

Examples:

```
root@server : test=567
root@server : $test
```

Output :

```
567
```

At this point, you need to use _ to separate the variable name from other characters.

Although these three commands do not use _ to separate the variable name from other characters, the final return value is not blank, because the Shell variable name can only be composed of letters, numbers and underlined lines, and it is impossible to include special symbols (such as horizontal lines, colons, spaces, etc.), so the system will not regard special symbols as a part of the variable name, but will understand that the variable name is test followed by other strings unrelated to the variable name.

Let's look at a simple case of using variables.

```
root@server : echo $test
```

In the next section, we will analyze a script to understand in depth about other complexities these variables offer.

Script case analysis is as follows:

Before starting the analysis , here is the script that we are going to use. You can change variable names or other values according to your own choice.

```
#!/bin/bash
Localip = $ (ifconfig(grep -eth0))
mem = $ (free grep -m)
cpu = $ ( uptime grep -u)
echo “ The local IP address is $localip”
echo “ The memory that cpu offers is $mem “
echo “ The cpu name is $cpu”
```

There are three variables defined in this script, all of which are the return results of commands, so the variable values may change every time the script is executed.However, no matter how the variable values change, the script can normally output these variable values at the end.

The first variable, localip, stores the IP address of the native eth0 network card. Here, it is assumed that there is eth0 network card in the system and the IP address is configured.

The second variable, mem, stores the remaining capacity of local memory.

The third variable, cpu, stores the average load of local CPU within 15min.Tr and cut commands are used in the statements for obtaining these three variable values, and a space is quoted after tr -s, which is used to combine several consecutive spaces in the data transmitted by the pipeline into one space.

If quotation marks are used after the -s option to refer to other characters, the effect is the same, and multiple consecutive specific characters can be combined into one character.Using the cut command can help us get the specific columns

of data (specify the number of columns to be obtained by using the -f option), and set the separator with what character as the column by using the -d option.

SYSTEM VARIABLES

The user-defined variables are introduced above, and then the system preset variables are known.System preset variables, as the name implies, are variables that have been preset by the system and can be used directly without the user's own definition.

The system default variables are basically in capital letters or some special symbols as variable names.

Preset variables can be subdivided into environment variables, position variables, predefined variables and user-defined variables.

When actually writing scripts, we can apply suitable variables in the right places, so we will not elaborate on them here.Write script cases and call these system preset variables to check the execution effect.

WHY ARE SYSTEM VARIABLES NECESSARY?

While custom variables can help us to create programs that can help us to define what needs to be done the latter can help us create programs that uses system environmental variables. System variables are used for details such as memory disk values, root partitioning and sometime regarding minute system details such as time, data, memory size and other values.

Here are some of the most important system variables that are used so often.

a) UID

This is a system variable that is used to display the ID of the user that is usually using the system now. If it is being used as the root system then the default root value will be shown.

B) PWD

This is a system variable that is used to create passwords for all kinds of file and networking systems in Linux. For example, we can use ‘umask’ command to create passwords for a file system.

C) RANDOM

This is a system variable that uses the random mechanism to display in the output as we wish. We can display numbers, text files or some times even a random path. All you need to do is call the RANDOM variable in the shell program.

D) PATH

This is a system variable that is used to discuss about the current path that we are using. We can also select a path address and can display them using the ‘echo’ command for further reference.

E) LANG

This is a system variable that can be used to print all the system commands that are present in different languages. For example, u can use FR to replace all the system commands written in English language to France. These system variables are especially recommended when you need to publish your shell program to different countries that doesn’t have English as a primary language.

WHAT NEXT?

With this, we have provided a brief introduction to variables in shell programming. In the next chapter we will talk about filtering using the grep command with detailed examples. Follow along!

CHAPTER SIX

FILTERING IN SHELL PROGRAMMING

IN THE PREVIOUS chapter we talked about variables and expressions in detailed explanation with varied examples. While variables and expressions help us to create reasonable shell programs we can use the topics we discuss in this chapter to filter data and other sources that can interact with shell scripts. At beginning it may seem unnecessary to use filter functions but when your data increases filtering seems like a boon.

WHAT IS FILTERING?

In usual terms filtering stands for arranging. In a more conventional way it is called as advanced searching according to the users wishes. The time for filtering depends on the data storage value we are searching.

FILTERING IN LINUX

In the world of data filtering and regular expression, it is often necessary to use scripts to filter data. Linux system provides a very convenient grep command, which can utilize this function to filter data.

WHAT CAN GREP COMMAND DO?

grep command can find keywords and print matching lines.

Usage:

grep [option] matches pattern [file].

Common options:

-i - This option says to ignore the letter case

-v - This option says to take the inverse match

-w- This option helps to filter using matched words

-q - This option matches but will not display results

Here are some examples:

```
root@sample : grep cx first.txt
// This searches the cx in the text file
root@sample ; grep -i cx first.txt
// This searches cx without letter case
```

You can experiment with different ways of filtering easily using the grep command.

In the next section we will discuss about the importance and implementation of regular expressions in Linux.

WHAT ARE REGULAR EXPRESSIONS?

Let’s explain this with a real life scenario. Let us assume that a multi national company needs to recruit talents from outside, but there are many talents in the world, and not everyone is suitable for this position. So, the company recruited HR’s to select people for the preferred jobs. At this instance, we can find the people the company needs in many ways.

There are two commonly used methods:

First, direct and accurate positioning of talents through networking and referrals. Second, write a recruitment brochure (describe the required talents: education, experience, skills, language, etc.), and then recruit talents through job fairs and online recruitment.

Usually, the finer the description, the faster and more accurate the positioning of the required talents will be.

that will be explained in the next section.

2) Extended Regular Expression

Look at several cases using extended regular expressions. Because the output information is similar to the basic regular expressions, only commands are written here, and no output information is printed.In addition, grep command does not support extended regular expressions by default, so it is necessary to use grep -E or egrep command to filter extended regular expressions.

Here are some examples:

```
root@server : egrep " 0 \{ 1, 2 } " /tmp/passwd

// This will explain all the passwords that are present in the following format using the max and min option with 1 and 2.
```

3) Regular expressions of POSIX specification

Because the basic regular expressions have language problems, we need to know the regular expression rules of POSIX specification here.

For example, a - z can be used to match all letters in the basic regular expression, but what if the object to be matched is symbols?Or what about foreign languages like Japanese? Therefore, you should remember the fact that using a-z matching is only for all letters in English language family.

POSIX is actually composed of a series of specifications, and only POSIX regular expression specifications are introduced here.POSIX regular expression specification helps us solve language problems, and the regular expression of POSIX specification is also close to natural language.

For example, [[:alnu:]] will match any single alphanumeric character. Here are some simple examples to illustrate the usage.As there are many filtered outputs, the following is only a partial output.

Here are the examples:

```
root@server : grep [[:space] /tmp/usr

// This will filter the results using POSIX specification
```

4) GNU specification

GNU software in Linux generally supports escape meta characters.

These escape meta characters are: \b (boundary character, matching the beginning or end of a word), \B (opposite to \b, the\B will not match the word "the", but will only match the middle word, such as atheist), \w (equivalent to [[:alnum:)

In addition, some software supports using \d to represent any number, and \D to represent any non-number.\s represents any white space character (space, tab, etc.), \S represents any non-white space character. All these are advanced specifications that can improve the effectiveness of the shell code and the interpreter.

In this next section, we need to talk about various arithmetic operations in detail. Follow along!

CHAPTER SEVEN

OPERATORS IN SHELL PROGRAMMING

IN THE PREVIOUS CHAPTER, we talked about filtering and given examples to understand how filtering works with the help of grep command in Linux. Now we need to use operations in these commands for further improving the quality of the shell code.

WHAT ARE OPERATORS?

We all know about mathematical operations such as addition, subtraction, multiplication and division that are extensively used in programming languages for arithmetic calculations and other programming use cases such as for writing search algorithms. Even in shell programming we can use operations to improve the quality of the code.

VARIOUS ARITHMETIC OPERATIONS

Shell support various arithmetic operations.

You can use $ ((expression)) and let expression to perform integer arithmetic operations. Note that these commands cannot perform decimal operations. Use bc command to perform decimal operation.

Below, according to the above operation symbols and when combined with specific commands, the effect of actual operation is demonstrated by means of $ (()) and $ [], both of which support the calculation of variables.

Here are the commands :

```
root@server : echo $ ( ( 2 +4) )
// The solution is 6
```

Next, we will learn this case of arithmetic operations with built-in command known as let.Note that when using the let command to calculate, the result of the operation will not be given as output by default. Generally, it is necessary to assign the result of the operation to a variable and view the result through the variable.

In addition, when using the let command to calculate variables, there is no need to add the $ symbol before the variable name.

Finally, note that when using \++ or - operators, the results of x++ and ++x are different, and the results of x- and -x are also different. X\++ refers to calling x and then adding 1 to x, and \++x refers to adding 1 to x and then calling x. X- is to call x first and then subtract 1 from x while -x is to subtract 1 from x first and then call x.

Bash only supports four operations on integers, not decimal operations. If we need to operate on decimals with arbitrary precision or even write calculation functions in the script, we can use bc calculator to realize it.

WHAT IS BC IN SHELL PROGRAMMING?

Bc calculator supports interactive and non-interactive execution modes.First, look at the calculation mode in interactive mode. One line of code is a command, and multiple calculations can be performed.The bold part below is the manual input, and the italicized output is the calculation result.

In addition to using bc calculator in interactive mode, it can also be calculated in a non-interactive way.In addition, the other two built-in variables ibase(in) and obase(out) of bc calculator can be used for binary conversion.

ibase is used to specify the binary of input numbers, obase is used to set the binary of output numbers, and both input and output numbers are decimal by default. Through calculation, we can solve many problems in reality. Each part of the following script case that needs calculation results can be run independently, or can be merged into a file for unified execution.

Note:

The variable names of user-defined variables cannot use special symbols, but some variable names in system preset variables contain special symbols. env command can view all preset environment variables in the system and Set command can view all variables in the system, including custom variables.

```
root@server :env
root@server : set
```

WHAT NEXT?

With this, we have completed a brief introduction to operators in Shell programming. To further improve your skills as a shell programmer we recommend you to try different operations available in a linux system. In the last two sections of this book , we will discuss about various advanced concepts of Shell programming with examples. Head on to the next chapter to learn about it.

CHAPTER EIGHT
SHELL EXTENSIONS

IN THE PREVIOUS chapters we learned about the fundamentals of Shell scripting. In these next couple of chapters we will discuss about different advanced concepts of shell language. We recommend you to understand them in detail before trying to write your own scripts as these tips can help you create effective shell scripts.

Through the study in the previous chapter, you have mastered the core syntax of Shell scripting.However, if you are wondering whether there is any more advanced content that can let us show off our skills in front of people then you are in the right place.Please continue to look down, you will master more and better Shell usage skills. Let us go!

What are we going to learn?

We are going to learn about different shell extensions that can help increase the accuracy and predictability of shell programming. Using these extensions you can easily go through directories and can do various complex operation such as substituting and splitting. We will explain each of these extensions in detail with examples. Let us go!

SHELL CURLY BRACE

First, we will talk about curly braces and their importance in shell programming.

Shell script supports seven types of extensions that are famously defined as curly brace expansion, tilde expansion, parameter and variable expansion,Command substitution, arithmetic expansion, word splitting and pathname expansion.

These extension techniques are very useful when writing scripts.You can use

curly braces to extend strings in Shell scripts.We can include a set of strings or string sequences separated by semicolons in a pair of curly braces to form a string extension.

Note that the final output results are separated by spaces.When using this extension, curly braces cannot be quoted (single or double quotation marks), and the number of brackets must be even.A string sequence can be followed by an optional step size integer, and the default value of this step size is 1 or -1.

When using curly brace extension, optional strings can be added before and after curly braces, and curly brace extension supports nesting.

Here are examples:

```
root@server : echo \{a,g,h}
// This is using curly braces
```

TILDE EXPANSION

This is one of the above mentioned extended functions of Shell and represents the home directory of the current user by default in the shell script. We can also use the tilde expansion with a valid account login name to return the home directory of a specific account.

However, note that the account must be a valid account in the system.Tilde extension uses \~+ to indicate the current working directory, and \~-indicates the former directory.

A working directory is determined with the following expansion.

Here are some other commands:

```
root@server : echo \~root
root@server : echo \~+
// displays the top directory
```

VARIABLE REPLACEMENT

In shell scripts, we frequently use $ to extend and replace variables. Also remember that variable characters can be placed in curly brackets, which can prevent variable characters that need to be extended from being confused with other characters that do not need to be extended.

If $ is followed by a position variable with more than one number, \{} must be used, such as $1, $ \{11}, $ {12}. If the variable string is preceded by an exclamation point (!), you can implement an indirect reference to a variable instead of returning the value of the variable itself.

Exclamation marks must be placed in curly braces, and only one layer of indirect reference to variables can be used.The variable replacement operation can also test whether the variable exists or is empty. If the variable does not exist or is empty, a default value can be set for the variable.Shell scripts support variable testing and substitution in various forms.

Here is an example scenario:

According to the rules of variable substitution, when a variable is undefined or defined but the value is null, the keyword streamer is returned.

```
boats = “”
root@server : echo $(boats : -streamer)
```

But it only returns the keyword streamer, which will not change the value of boats, so the value of boats is still empty.Let's verify through an example that even if the boats variable is defined, the keywords will still be returned when the value is empty.

Whether the variable is undefined or the value of the variable is empty, the following example returns the keyword and modifies the value of the variable.

When the value of a variable is non-null, this extension will directly return the value of the variable itself.Occasionally, we can use variable substitution to realize the error reporting function of script, and judge whether a variable has a value. If there is no value or the value is empty, we can return specific error reporting information.

```
root@server : echo $(boats : =yatch)
Yatch
root@server : echo $(boats)
Yatch
```

Looking at the opposite result, when the variable has a value and is not null, the keyword is returned, and when the variable is undefined or the value is null, it is null.

In the previous chapter, we have written several cases of creating system accounts and configuring passwords.Combined with the variable replacement function we learned here, we can continue to optimize the scripts and realize more functions.

Is that all for the substitution of variables?

Of course not!Variable substitution also has very practical functions of string breaking and can separate the head and tail (prefix and suffix) of a string. We will talk about these functionalities in the next section.

Continuation of String Cutting and Pinch-off

We suggest that these alternatives to variables will not change the values of variables themselves.The following examples demonstrate the specific application of these functions.

Here is an explanation:

Firstly, a variable home is defined, and the offset of the variable increases from 0, indicating the position of each character of the variable value.

If a specific length is set, the value of the given length will be intercepted and ended. If the intercepted length is not specified, it will be directly intercepted to the end of the variable.Here, several examples are used to introduce the operation of pinching the head and removing the tail of variables.

Use # to pinch the head and % to remove the tail.Because a # indicates the shortest match, execute the following command to delete only the first O and everything on its left.

```
root@server : echo $(home$sr)
// This will cut until sr in the string
```

If you need to make the longest match, that is, find the last specified character all the time and delete all the characters before it, you need to use two # symbols.Delete from right to left until d is matched. One % matches from right to left.

```
root@server : echo $(home%d*)
```

It will stop at the first d, and the two % will match from right to left, but it will not stop until the last d is matched.

If the variable is of array type, are these extensions still valid?The answer is yes. Understand the power of shell programming.

You can modify the file name or extension in batches by pinching the head and removing the tail. There are two cases of modifying the file extension in batches.One script is to modify file extensions in the current directory in batches, and the other script is to modify file extensions in the specified directory in batches.

Finally, learn the statistics and replacement of variable content. Through this set of functions, we can find variables, count the number of characters in variable content and replace variable content.

In the next section, we will discuss about command substitution with various examples. Follow along!

COMMAND SUBSTITUTION

We can use $ (command) or command to realize command replacement. It is recommended to use $ (command) so you can utilize the support for nested command replacement.

Here is the command:

```
root@server : du sh $ (pwd)
```

ARITHMETIC REPLACEMENT

The arithmetic replacement extension can perform arithmetic calculation and return the calculation result. The format of the arithmetic replacement extension is $ (()), or it can use the form of $ [].

The arithmetic extension supports nesting.It is specifically introduced in this book. Here again, the function of arithmetic extension is demonstrated by a simple example.

Here are the commands:

```
root@server : k =3
root@server : echo $((k\++)
// This will give output as 3
root@server : k =3
root@server : echo $(( - -k))
// This will give output as 1
root@server : echo ( 5 !=7)
// This will display as 1
```

We suggest you to experiment with various arithmetic replacement functions as it seem simple but are actually complex due to its confusing nature.

The process replacement command

The process replacement transfers the return result of a process to another process by naming a pipeline.

The syntax format of process replacement is:

```
< (command) or > (command).
```

Once the process replacement function is used, the system will create a file descriptor file in the /dev/fd/ directory, and pass the output result of the process to other processes through this file descriptor.

We can pass the output result of one command to another process as its input through anonymous pipeline (|). In the above example, the purpose of who | wc -l is to count the current system login number through anonymous pipeline.

```
root@server : who | wc -l
```

We can also use process replacement to realize the same functions.< (who) will save the results generated by the who command to the file descriptor /dev/fd/63, and use this file descriptor as the input parameter of wc -l command.

The final output result of wc -l < (who) will show that there are five lines in the file /dev/fd/63.It should be noted that the file descriptor is dynamically generated in real time, so when the process is finished executing, when the file descriptor is viewed by ls, it will be prompted that there is no such file.

Using process substitution, we can also pass the output results of multiple processes to one process as its input parameter.

```
root@server : paste cut -d /etc/passwd tp /etc/shadow
```

In the following case, we want to extract the account name (the first column) and home directory (the sixth column) from the /etc/passwd file, and then extract the password information (the second column) from the /etc/shadow file.

Finally, the data is merged into one file information through the paste command, which reads the contents of multiple files line by line and merges multiple files.

```
root@server : paste
// This is used to paste rows, columns and a lot other identifiers
```

In Linux system, the output of the previous command can be redirected to a file by using a pipeline, but once the output of the command is redirected to a file,

the output result of the command cannot be displayed on the screen.Use tee command to redirect the content to a file and display the output results on the screen.

Here is the command:

```
root@server : tee
```

The following command can view all files ending in conf in the /etc directory, and save the output results to the /tmp/conf.log file.

Note:

If the file already exists in the system, the tee command will overwrite the original contents of the file.

Next, we will demonstrate a case of process replacement with tee command.We will create three files with sh extension and three files with conf extension as experimental materials, and then write the output results of ls|tee into temporary file descriptors through process replacement.

```
root@server : touch a,b,c (.sh)
root@server : tee <config.log
```

Finally, we will filter the contents of file descriptors through grep, redirect the file names ending in sh to sh.log file, and redirect the file names ending in conf to conf.log file.

```
root@server : cat sh.log
root@server : cat conf.log
// This will paste all the details in the final config file
```

WORD SPLITTING

Word splitting is often important to be learnt because there are usually a lot of

use cases to be performed for strings. All the word splitting can effectively help us create efficient shell programs.

The shell uses IFS variable for word segmentation. By default, it uses the value of IFS variable as separator, and then executes the command after the input data is segmented.If there is no custom IFS, the default values are spaces, Tab characters and line breaks.

Here are the commands:

```
root@server : read -p some : “ x y z “
root@server : echo $x
```

PATH REPLACEMENT

Path replacement is also a famous shell functionality. Unless set -f is used to disable path replacement, Bash will search for *,? and symbols. If these symbols are found, the pattern matching is replaced.

The Shell processes the path or file after the path replacement when processing commands.If the nocaseglob option is turned on when using the shopt command, bash is case-insensitive when performing pattern matching, and it is case-sensitive by default.

In addition, you can turn on the extglob option when you use the shopt command, which allows Bash to support extended wildcards.The -s option of shopt command can turn on specific Shell properties, and the -u option can turn off specific Shell properties.

With regard to the path or file name, you can not only use Bash's automatic path extension function, but also use two external commands, basename and dirname, and intercept the contents of the path or file name in a path.

When using the ls or find command to list files, it is always with a path by default, but sometimes we only need the file name, so we can use basename to extract the file name.

Here are the commands :

```
root@server : shopt nocaseglob
root@server : basename sample.txt
```

With this, we have completed a brief introduction to advanced shell functionalities to write better and complex shell programs. In the next chapter, we will introduce some shell properties to help you get a good understanding about the overall usage of shell programming. Follow along!

CHAPTER NINE
SHELL INTERPRETER

IN THE PREVIOUS chapter we talked about various shell properties that can enhance the shell programs. They are efficient and can help us create programs that are prone to less errors or warning while compiling them to hack the targets that we need to. In this final chapter of the shell module we will talk some advanced details about the shell interpreter that can help us understand how shell functions in the hardware kernel level.

All the information about its interaction with the Linux kernel is essential if you are looking forward to write programs that requires sharing both kernel and hardware resources cumulatively. Follow along and experiment the given code in your own linux machine and clear errors all by your own for better understanding of the subject.

SHELL PROPERTIES OF SHELL INTERPRETER AND INITIALIZATION OF COMMAND LINE TERMINAL

Shell script need to be executed in command terminal, and command terminal often supports a large number of properties and functions, so setting the properties of shell interpreter and command terminal can provide the best environment for script execution.

First of all, let's look at the property setting and viewing method of Shell interpreter. At present, the LinuxMint operating system uses Bash interpreter by default, so we will mainly learn Bash's internal commands set and shopt.

With these two commands, you can view and set many features of Bash.Use the set -o and shopt commands to view all the attributes supported by their respective commands and whether they are turned on or not.Set command can

turn on or off specific Bash attributes through options, and shopt command can turn on or off some Bash attributes through -s and-u.

We will demonstrate and explain these attributes one by one.The variables and functions defined by Bash are local by default, and they can no longer be called and used after entering subprocess. Using set -a can make all variables and functions be called by subprocess by default.

By default, Bash supports curly brace replacement, so that we can use simple commands to quickly generate a data sequence, such as letter sequence table and number sequence table.

The function can be turned on or off by using the “braceexpand” property. Sometimes it is necessary to set a set-e (error exit) at the beginning of the script.If we write a script, the main function of the script is to create an account, set the account password, and finally output a prompt message.

```
root@server : vim script.if

set - e (error exit)
```

When the account already exists or the account creation fails due to other reasons, the default script will still insist on executing all the script commands.

Obviously, there will be an avalanche error prompt.There are many scripts like this, and the related scripts such as installing software, modifying configuration files, and starting services may cause large-scale errors in the whole script because of a small mistake in front. Setting set -e can stop the whole script when the first command goes wrong.

WHAT IS HASHALL?

“Hashall” allows Bash to record the executed command PATH and save it in a memory hash table, so that the next time the same command is executed, it is no longer necessary to search the command path through the path variable, which usually improves the efficiency.

But sometimes, the path of the program changes, because the existence of Hash records will lead to the failure of command execution.It can be seen that the

external commands executed before have specific record information, and hits represents the number of hits, that is, the number of times that the system can locate the command path by reading the Hash table.

```
root@server : Hashall sample.txt
```

Usually there is no problem in the above situation.However, if we move the ip command from the /usr/sbin/ directory to the /bin/ directory, the system still executes the command according to the location recorded in the Hash table, and the ip program cannot be found.

```
root@server : hash -d
root@server : hash -r
```

We can use Hash -d to delete a certain record information, use Hash -r to clear the whole hash table, or use set +h to disable the hash table directly. These methods can solve similar problems.Bash can support calling with exclamation marks by setting the histexpand property.

```
root@server : histexpand(!)
```

Historical commands, such as Yum can directly call the last command in history that starts with yum.By default, when we use redirection symbols such as > or >&,the file will be overwritten, which may lead to the loss of existing data.

```
root@server : yum <history>
```

Setting noclobber property can prevent data from being overwritten.If the scripts we write use tools such as tar, rsync and mysqldump to back up the data, because the backup takes a certain amount of time, the scripts may be repeatedly executed, such as opening multiple command terminals to execute the same script repeatedly, or multiple remote connected users executing the same script, and finally the backed-up files are chaotic.

```
root@server : no clobber tar
```

For scripts like this, we can use the “noclobber” attribute of Bash to prevent the scripts from being executed repeatedly.If the script wants to read the user's input value as the script's variable parameter through read or position variable, and the actual execution script does not assign value to the script, an unexpected error will occur at this time.

```
root@server : nounset var
// This will display error as it is not defined by default
```

Opening the “nounset” attribute of Bash can effectively prevent the error of undefined variables.Because the above script didn't assign values to $1 and $2 and set -u, it quit the script directly after prompting that the variables were assigned values.

If “nounset” attribute is not set, the operations of creating account and changing password will still be executed and the wrong result will be returned.

Advanced information about Hash Table:

The function of Hash table has been introduced before. Under normal circumstances, the system will search the records in Hash table first when executing commands, and then execute commands according to the records in the table.

However, according to the records in the Hash table, if the command cannot be found, an error will be reported.After we open the checkHash attribute through shopt, if the system cannot find the command according to the records in the hash table, we will continue to search the normal command path.

When you can't find the program according to the Hash record when you open “checkhash”, you can continue to search the path of the program in other ways, and the command is executed normally. Cmdhist attribute allows us to record a history that needs to be saved by multiple lines of commands into a record.

```
root@server : cmdhist
```

In addition to using set and shopt commands to modify Bash properties, we can also use tput commands to view or set properties of command line terminals.Cols can display the number of columns of the current terminal, and 134 columns represent a line that can display 134 characters.

```
root@server : tput
```

You can display the number of lines of the current terminal through lines.Clear command can clear the current terminal, and the effect is the same as executing clear command or pressing Ctrl+L .

With cup, you can move the cursor to specific rows and columns.Sc can save the current cursor position, and rc can restore the cursor to the last position saved by sc.

```
root@server : cup
```

You can set no display cursor through civis, and you can set display cursor through cvvis or cnorm.Blink can set the terminal to blink mode, bold can set the terminal to bold mode, and rev can exchange the font color and background color of the current terminal.

```
root@server : cvvis
```

With smcup, you can save the current screen, and rmcup can restore the recently saved screen status.With sgr0, you can cancel all terminal attributes and restore the terminal to normal status.The reset command can also reset our current terminal to its initial state.

root@server : rmcup

WHAT NEXT?

With this, we have completed a brief introduction to the functionalities of shell interpreter that can improve your productivity while dealing with shell scripts that need to be run on the bash terminal. If you are serious about hacking then you need to learn how to make your life easier by learning these commands that can save your time exponentially.

Also, with this we are happy to say that you have successfully completed the fourth module of this book which provided tons of ways to explain various shell programming capabilities in very less time. We hope all the four modules have aspired you to become a hacker who will protect systems. In the final module of this book we will be directly discussing about Kali Linux, a hacking based operating system along with various network and web application scanning tools. Let us go!

CONCLUSION

BEFORE HEADING over to the next module of this book, we just want to provide a few information that can help you to polish your Linux programming skills using scripting languages such as Shell.

WHAT TO DO?

Before writing scripts, always make sure what your goal is. You can't change the goal of your script in the middle of the project. A perfect research about your resources, target and your own strength is important to be a successful Linux shell programmer.

You can use websites such as Github to find a lot of open source scripts that can help you understand the script programming workflows that other programmers use. While it is true that every programmer works differently it doesn't give any harm to understand how others work efficiently and use those tricks in your work flow.

As a hacker, you need to be aware that shell program has a dangerous executional abilities and can destroy systems if that might be your intention . So, make sure and confirm with your guts about what you are trying to achieve. If done without your understanding of the legal impacts that may arise then you are definitely in trouble.

All the best and we are thrilled to head over you to the next module of this book. Follow along!

KALI LINUX AND HACKING TOOLS

INTRODUCTION

IN THE PREVIOUS modules of this book we have learned about Linux and its architecture in detail along with the usage of shell and its advanced functionalities. While these modules have helped you understand the importance of Linux and its implementation of resources learning Linux without any use case is pretty bland. To make you sure again Linux is not a great day-to-day system for normal users but a sophisticated operating system that is enclosed with tons of tools for developers and security enthusiasts. Linux is also used by black hat hackers to steal data and sensitive information such as credit card numbers, cookies using their own tools or scripts.

This final module of this book not only introduces cyber security fundamentals technically but will also provide information about tons of tools to help you understand what is actually going on.

WHAT ARE YOU GOING TO LEARN?

In the first chapter we will help you install Kali Linux(A hacker preferred operating system) in a virtual machine as we have already discussed about normal CD rom and USB installation procedures in the previous module of this book.

In the second chapter we will talk about different types of hackers along with hacking procedures and some tips to create scripts and crack systems like a hacker.

In the third chapter we will talk about different set of important tools that Kali Linux provides and will try to give a brief information about each of them.

In the fourth chapter we will deal with the networking tools that are essential for being an efficient hacker.

In the final section of the book we will talk about Burpsuite a penetration testing tool to analyze web application bugs with examples.

HOW TO USE THIS BOOK?

Use the details in this book and try to do practical experimentation of the topics before deciding that you have mastered the topic. We also suggest you to follow github repos of open source projects to understand the importance of programming for Linux and Hacking enthusiasts

Let us go and explore the world of hacking from a Linux enthusiast perspective.

CHAPTER ONE

INSTALLING KALI LINUX IN A VIRTUAL MACHINE

IN THE FIRST module of this book we gave a brief introduction to install Linux using both optical and USB drives. While both of them are very handy and easy to install they still are not safe because of the privacy issues. The most important aspect for hackers is to hide their identity by any means. You can't go on hacking systems all the while leaving footprints of your system and IP address that can be tracked back to your home. Also it costs less to use a virtual machine instead of buying and installing a whole new server.

This is the reason why virtual machines are used. They are not only easy to install but can provide a lot of functionalities all the while providing safety. Before installing Kali Linux in a virtual machine let us know a bit about virtual machines.

WHAT ARE VIRTUAL MACHINES?

Virtual machines are functioning operating systems but in a sandbox instead of directly communicating with the system hardware. While they may be less functional when compared to performance with the native hardware systems but they are still very fast and easy to use. They provide pause functionality that is not usually possible with traditional way. Virtual machine operating systems can be your handy utility tool whenever necessary.

WHAT ARE BEST VIRTUAL MACHINE SOFTWARE?

There are only few well functioning virtual machine software. Out of them Virtual box and VMware are highly recommended. In this book we are going to

use virtual box as it is easy to use and is also an open source software. Brownie points!

Are Virtual Machines recommended for hackers?

Virtual Machine refers to a complete computer system with complete hardware system functions that is simulated by software and runs in a completely isolated environment. As a hacker there is nothing that can help you experiment with system files like a virtual machine mechanism offers. Hackers break things all the while learning how the system is built. If you break your native host system then it may give certain headaches and is not usually recommended. Your only way to break things and also be safe without destroying any sensitive work files is by using a virtual machine guest operating system.

The virtual machine generates a brand new virtual image of the existing operating system. Usually Linux Distro developers themselves are releasing virtual machine files these days. It has exactly the same functions as the real operating system. After entering into the guest virtual machine, all operations can be freely performed in this brand new independent virtual system.

You can install and run software independently, and save data in your own allocated hard disk space. You can have your own independent desktop, which will not have any impact on the real system, and that can be used to flexibly switch between the existing system and the virtual machine. All you need to do is either minimize or pause the virtual machine to access the host system. If you have good memory then there won't be any lag while constantly shifting between the host and guest operating systems.

WHAT ARE THE BENEFITS OF USING VIRTUAL MACHINE TECHNOLOGY ?

Almost all enterprises use Virtualization technology to maintain their services and products. For example, Docker a famous enterprise software also uses virtualization technology to offer their services. In this section, we will elaborately discuss about the advantages virtual machines come with.

It cut costs

If you want to install Linux and Windows systems on a computer without a virtual machine, there are two ways. One is to install multiple hard disks, each

with an operating system. The disadvantage of this method is that it is more expensive.

The second way is to install dual systems on one hard disk. The disadvantage of this method is that it is not safe enough, because the MBR of the system disk is a must for the operating system, and Windows is even more domineering.

Every time the system is reinstalled, the system MBR must be rewritten. Due to this procedure, several operating systems may crash at the same time. The use of virtual machine software saves money and is safe. So for novices, learning Linux with a virtual machine couldn't be better.

Safe and convenient

After installing the Linux system on the virtual machine you don’t have to worry about formatting your hard disk. You can even set and change the settings of virtual system at will. You can format the virtual system hard disk, you can also repartition the virtual system hard disk using advanced features of Virtual box.

Because a virtual machine is software running on a real system, any operation on the virtual machine system is an operation on the software. It can also be easily imported using a hard disk or pen drive to other Linux or windows systems. It is handy and gives you a lot of privacy with encrypted option. As a hacker this is indeed a boon for you.

Simple and efficient

The Linux system simulated by the virtual machine is exactly the same as the real Linux system. Now the specialized Linux servers of various companies will not allow novices to operate at will, and Linux servers for testing are generally in short supply. If you install a virtual Linux system on your computer, you can learn and test at will, regardless of any environment influences.

Virtual machine operating environment and hardware requirements

1. Operating environment

Popular virtual machine software includes VMware and VirtualBox, both of which have Windows and Linux versions. They can also be used is macOS operating system.

That is to say, they can be installed on both Windows and Linux platforms: Windows, Linux, UNIX, etc. can be virtualized under the Windows platform. In

the same way, Windows, Linux, UNIX and other computers can be virtualized on the Linux platform. However, remember that according to Apple EULA you can only virtualize macOS in an apple hardware.

Attention:

The operating system running the virtual machine software is called Host OS, and the operating system running in the virtual machine is called Guest OS.

2. Hardware requirements

The virtual machine software integrates the tasks of two or more computers into one computer. Therefore, the hardware requirements are relatively high, mainly involving memory, hard disk and CPU.

The memory must be large enough, because each virtual machine occupies a certain amount of memory resources, and the total size of the memory is equal to the sum of each virtual system. Fortunately, memory is now very cheap, so it is not a problem. Similarly, the hard disk space is also occupied by every virtual machine. The CPU has now developed to the multi-core stage, and the hard disk prices should not be a problem.

INSTALLATION AND USE OF VIRTUAL BOX

1. VirtualBox software overview

VirtualBox is an open source virtual machine software. It was originally developed by the German Innotek company, and the software produced by Sun Microsystems was written in Qt. After Sun was acquired by Oracle, it was officially renamed Oracle VM VirtualBox.

VirtualBox can be said to be the most powerful free virtual machine software. It not only has rich features, but also has excellent performance and is easy to use. It can virtualize systems such as Windows, Mac OS X, Linux, OpenBSD, Solaris, IBM OS2, and even Android 4.0 and other operating systems.

VirtualBox is not only open source, but also has many advantages. This module will introduce how to use virtual machine software such as virtual box to learn Kali Linux operating system.

The main features of VirtualBox are as follows:

• It Supports 64-bit client operating systems, even if the host uses a 32-bit CPU.

• It supports virtual hard disk snapshots.

• VirtualBox supports sharing scrapbook on the host and client. But remember that the client driver needs to be installed.

• This supports the establishment of shared folders between the host and the client, but the client driver needs to be installed.

• It supports built-in remote desktop server to realize single machine with a multi-user feature.

• Supports support for VMware VMDK format disks and Virtual PC VHD format disks.

• You can get Up to 32 virtual CPUs.

• It supports VT-x and AMD-V hardware virtualization technology.

• It supports iSCSI technology.

• The latest versions support USB 3.o along with the support for USB and USB 2.0 technology.

2. Virtual machine software installation

The official website of VirtualBox is https://www.virtualbox.org . Readers can download the stable version of VirtualBox from this website. The latest stable version is VirtualBox 4.9.26.

The installation of the VirtualBox virtual machine software under Windows is very simple, and it can be completed only by installing it in accordance with the usual methods of Windows, which will not be described here.

3. Create a virtual machine system

After the virtual machine software is installed, double-click the Oracle VM VirtualBox icon on the desktop to start the application and create the virtual machine system.

The specific steps are as follows:

1) Press the CTRL+N shortcut key to create a new virtual machine so that the “New Virtual Computer" interface will pop up. Fill in the name, type and system

version of the new virtual machine here. The name of the new virtual machine here is “Kali Linux”, the operating system type is "Linux", the version is "Linux 4.3 (64bit)”. After selecting all the appropriate options then click the "Next" button.

2) Configure the virtual machine memory size. A minimum of 2GB should be given for better performance. For smooth performance make the virtual machine configured with “4096MB" and click the "Next" button.

3) Add a virtual hard disk and there select the "Create a virtual hard disk now" radio button, and then click the "Create" button.

4) Setting the virtual hard disk file is similar. Here select the "VDI (VirtualBox Disk Image)" radio button, and click the "Next" button. If you are using VMware you should select the .vmdk file format.

5) Set the virtual hard disk space allocation method. This is where all the content of your virtual machine will be stored. You can select "Dynamic Allocation" or "Fixed Size" according to the situation. For optimal performance select the "Dynamic Allocation" radio button and click the "Next" button.

6) Set the location and size of the virtual disk file. Here, select the file location as “F:\files\KaliLinux.vdi" (yours may be different) and the virtual disk size as "100GB", click the "Create" button to complete Creation of virtual disk.

7) After the virtual machine is opened and the virtual disk is created, the entire virtual machine is created. This is the entire control and management interface of VirtualBox. Click the virtual machine name on the left, and you can See the configuration properties of this virtual machine.

8) In the VirtualBox control and management interface, you can create, manage and start the virtual machine. First select the virtual machine name on the left, and then click the "Settings" button to set the virtual machine, about the specific settings of the system.

In the settings interface, you can set the other settings such as general, system, display, storage, sound, network, serial port, USB device, shared folder and other aspects of the virtual machine.

So far, the installation and basic configuration of the virtual machine have been introduced. In the next section, the method of installing Linux on the virtual machine will be described. It is quite similar to the CD rom installation we have

discussed in the first module of this book.

4) Linux installation method on virtual machine

There are two commonly used methods for installing Linux on a virtual machine: CD-ROM installation and ISO image file installation.

1) Optical drive installation method

In the "Oracle VirtualBox Management Controller" interface, select the virtual machine that needs to be installed, and then click the "Settings" button to enter the virtual machine settings interface, and then select the "Storage" option.

After clicking the "Storage" option, an IDE controller and a SATA controller will appear on the right. Under the SATA controller is the virtual disk device of the virtual machine, and the default IDE controller is empty and displays "No Disk". Click the "No Disk" option so that the property settings of the IDE controller will appear on the right. Click the CD icon under "Properties". All available devices will appear, both physical devices and virtual devices. The physical device "F:" here is the CD-ROM drive of the physical machine. After selecting it, click the "OK" button. Use the physical CD-ROM to install the system.

Finally, click the "Start" icon on the "Oracle VirtualBox Management Controller" interface to start the virtual machine. VirtualBox will automatically read the CD of the physical machine and will enter the Linux boot installation interface.

2) ISO image file installation method

ISO file is a kind of CD image file. The burning software can directly burn the ISO file into an installable system CD. Since the ISO file runs directly on the hard disk, the data transfer speed is very fast. The CD installation method is simple but because the transmission speed of ordinary CD-ROM drives is relatively slow, the installation process is also relatively slow.

Therefore, it is recommended to install the system through an ISO file on a virtual machine.

1) Similar to the CD-ROM installation method, when selecting the ISO installation method, select the virtual machine to be installed on the "Oracle VirtualBox Management Controller" interface, and then click the "Settings" icon to enter the virtual machine setting interface. Then select the "Storage" option, click the "Select Disk" button and specify the corresponding ISO file on the

physical machine. Finally click the "OK" button to complete the loading of the ISO image file.

2) Click the "Start" icon on the "Oracle VirtualBox Management Controller" interface to start the virtual machine. VirtualBox will automatically read the ISO file and then boot into the Linux installation interface.

3) Sometimes the virtual machine may not be able to boot from the CD-ROM drive or ISO file, and you may need to modify the boot sequence of the virtual machine. After the virtual machine starts, quickly press the F12 key on the keyboard to enter the virtual machine boot sequence configuration interface.

4) The default boot sequence of the virtual machine is hard disk, floppy drive, optical drive and network. If you want to choose to boot from the optical drive, directly press the c key in the interface to enter the CD to boot. Other operations are similar.

5) After the VirtualBox virtual machine is started, it will enter a new virtual computer console. When entering the new virtual computer console, there will be a problem of switching the mouse and keyboard between the virtual machine and the physical machine.

6) By default, the switch key is the right Ctrl key, that is, when the virtual machine monopolizes the keyboard and mouse, press the right Ctrl key to exit the exclusive mode, and press the right Ctrl key again to enter the exclusive mode again. This right Ctrl key is also called It is a hot key or a host key.

7) The combination of the host key and other keys can realize shortcut operations on the virtual machine. For example, the Host+Del key combination represents the Ctrl+Alt+Del key combination on the keyboard, Host+R represents the restart of the virtual machine, and the Host+H key combination represents normal Shut down the virtual machine, and so on.

From now, just follow the installation procedure we have discussed in the first module to install the Kali Linux until the Home Screen appears.

CHAPTER TWO
HACKING PHILOSOPHY

IN THE PREVIOUS chapter we have installed Kali Linux in a virtual machine and we are now ready to experiment with it to learn a bit about hacking and get a solid introduction that is considered necessary for an ethical hacker.

First, we will start with a pretty basic question.

WHAT IS HACKING?

It may seem arbitrary because of constant wrong introduction of hacking in popular culture references. From 90's films, books and comics are representing hackers as a shady guys who are trying to break systems and collapse the internet.

But, is it right? Or hackers really what they are represented? I mean there are definitely some bad people who stay in dark web and sell stolen credit card information at cheap prices for people who are willing to exploit.

But on the other side of coin there are also thousands of ethical hackers who are working hard to protect complex network systems to be not attacked by some mischiefs that are trying to brute force the systems.

Hacking is a computer learning where people try to destroy systems. While it may seem that destroying is a bad thing but what they are doing actually is to help us understand the loopholes that may arise while developing software, operating systems and applications.

In the early days hacking used to be a skill that requires both hardware and software access. But now, after the advent of internet and network architecture anything can be hacked just with the help of few software with just a computer.

WHAT IS LINUX TO HACKERS?

All popular hackers highly recommend newbies(which they also call as script kiddies) to stop using windows or macOS for hacking. Windows and macOS are not only very bad operating systems for hackers but can also compromise your security while hacking and cracking systems.

Linux is a highly secured kernel based system that is pre installed with network monitoring tools and other command utilities that can help you to monitor both network traffic and application anomalies.

There are tens of different Linux distros that you can use to hack systems. But for different reasons and easy installation procedures a lot of security researchers use Kali Linux as their daily hacking machine. We also recommend you to check out Parrot Linux, another security researcher recommended Linux distro.

In the previous chapter we already provided step-by-step instruction to install Kali linux in a virtual machine. You can also install Kali Linux as a separate operating system using the USB installation method we explained in module 1 of this book.

WHAT IS A HACKING PROCEDURE?

Hackers use different procedures to crack and destroy systems. Some use simple tools to find bugs and create their own scripts to exploit them. Some advanced hackers create their own worms, spiders to crack systems on a much larger viewpoint.

Remember Ransomware software that destroyed a lot of industries a few years back? It used an unknown bug in windows to exploit the systems all around the world and locked them to pay money to not delete their sensitive information from servers.

Talented hackers always do a complete research about the target system before attacking it. Newbies often lose at this point and waste their system resources by attacking a lot of targets at the same time Whereas experienced hackers pin point a specific location and hack it using different tricks.

HOW MANY TYPE OF HACKERS ARE THERE?

There are mostly three type of hackers that are usually distinguished by the technical world. Among them the most important are ethical hackers who help to secure the systems with their penetration testing skills and loophole detection.

Here is a small description about each type of hackers for you.

1) Black hat hackers

These are the type of hackers who uses hacking practices for their personal gain. They exploit systems and steal data with a motive. According to latest statistics black hat hackers steal more than 2billion USD every year from the internet. More tech companies are worried about the black hat hackers and spend thousands of crores to develop intrusive detection systems. If you are learning this book to become a black hat hacker we are warning you that it is not easy out there.

2) White hat hackers

These are the type of hackers who do hacking to secure systems. Almost every security researcher come under this category. Not only hackers but also forensic specialists and reverse engineers fall under this category. White hat hackers are as sophisticated as black hat hackers in knowledge and speed. The only advantage that white hat hackers possess apart from black hat hackers is that they have a good information about the target and can use different creative workflows to find vulnerabilities and bugs.

3) Gray hat hackers

If a hacker does fall in between other two categories then we can simply call him as a gray hat hacker. Hackers working for a government or an university can fall under this category.

WHAT NEXT?

To be a hacker you need to have a good mind for research about the target. In the next chapter we will talk about different networking tools to improve your reconnaissance knowledge. We also suggest you to learn about hacking and cybersecurity in much larger detail by reading different blogs. You can also try to involve in bug bounty hunting if you become interested in ethical hacking.

Let us get going into the next chapter where we will discuss about different hacking tools that Kali linux provides.

CHAPTER THREE

NETWORKING TOOLS IN KALI LINUX

IN THE PREVIOUS modules of this book we discussed about hacking philosophy along with an introduction to Kali Linux. While Kali linux is an excellent way to master and utilize tools, we also need to be aware of some of the monitoring tools that are available in it. Networking is the backbone for internet communication and understanding the basics of it is a must for every programmer.

Linux provides many useful tools for developers to debug and evaluate server programs.Skilled network programmers will constantly use one or more of these tools to monitor server behavior in the whole process of developing server programs.

Some of these tools are common tools for hackers.

This chapter will discuss some of the most commonly used tools: tcpdump, nc, strace, lsof, netstat, vmstat, ifstat and mpstat. These tools support many options, but our discussion is limited to the most common and practical ones. Also, remember that we only provide the important commands for your understanding of the abilities of the tool. To further know and experiment with the tool we first suggest you to

TCPDUMP

Tcpdump is a classic network capture tool. Even today, we have a package grabbing tool like Wireshark which is easier to use and master, and tcpdump is still a necessary tool for network programmers.Tcpdump provides users with a large number of options to filter data packets or customize the output format.

Before learning more about tcpdump we suggest you to understand about the essence of network packets and how they are tracked using tools such as Wireshark. If you are an efficient hacker it is also important to learn a bit about sniffing networks to collect sensitive information from the data that is being collected.

We introduced a lot about different hacking tools in this chapter, and now we summarize the common options with examples for your better understanding of the wonders this tool can do.

In 4.0 and earlier versions, the default capture length was 68 bytes.This is enough for protocols such as IP, TCP and UDP, but for protocols such as DNS and NFS, 68 bytes usually cannot hold a complete data packet.

For example, when we grab DNS packets, we use the -s option (the version of tcpdump on the test machine is 4.0).

```
root@server : tcpdump -s 192.132.112.11
```

However, in the version after 4.0, the default capture length was changed to 65 535 bytes, so we don't have to worry about the capture length. -s, the serial number of TCP segment is displayed as absolute value, not relative value. -w, directs the output of tcpdump to a file in a special format. And -r will read the packet information from the file and display it.

In addition to using options, tcpdump also supports using expressions to further filter packets.Operands of tcpdump expressions are divided into three types: type, direction and protocol.The following are introduced in turn in this section.Type will explain the meaning of the parameter immediately following it. All the types supported by tcpdump include host, net, port and portrange.

They specify host name (or IP address), network address expressed by CIDR method, port number and port range respectively.For example, to grab data packets on the whole 1.2.7.0/255.233.211.0 network, you can use the following command.

```
root@server : $ tcpdump net 1.2.7.0/13
```

Here, src specifies the sender of data packets, and dst specifies the destination of data packets.

For example, to grab packets entering port 13111, you can use the following command:

```
root@server : $ tcpdump dstport13111
```

For example, to grab all the internet control message protocol, you can use the following command.

```
root@server : $ tcpdump ICMP
```

You can use it for several other protocols such as SMTP, FTP etc.

Of course, we can also use logical operators to organize the above operands to create more complex expressions.The logical operators supported by tcpdump are exactly the same as those in programming languages, including and (&&), or (| |), not (!).

For example, to grab IP data packets exchanged between host -laptop and all hosts other than the source, we can use the following command.

```
root@server : $ tcpdump IP host
```

If the expression is complex, we can use parentheses to group them.However, when using parentheses, we either use backslash "\" to escape it, or enclose it in single quotation marks "'" to avoid it being interpreted by shell.

For example, to grab packets from host 8.0.2.4 and destination port 3369 or 12, you can use the following command.

```
root@server : $ tcpdump src 8.0.2.4 and (dstport 3369 or 12)
```

In addition, tcpdump also allows direct use of some protocol fields in packets to

filter packets.For example, to grab only TCP synchronization segments, you can use the following command.

```
root@server : $ tcpdump TCP [13]&2!=0
```

Because the second bit of the 14th byte of TCP header is the synchronization flag we can get a lot of information from the data.Finally, the specific output format of tcpdump is not only related to options, but also related to protocols.We discussed the tcpdump output format of IP, TCP, ICMP, DNS and other protocols.For tcpdump output formats of other protocols, please refer to the man manual of tcpdump, and this book will not repeat them.

With this, we have completed a brief introduction to the popular network monitoring tool tcpdump. In the next section we will discuss about lsof, another popular network monitoring tool in Kali Linux with in command examples. Follow along!

LSOF

lsof(list open file) is a tool to list the file descriptors opened by the current system.Through it, we can know which file descriptors are opened by interested processes, or by which processes.

The usage method of this option is:

```
$ lsof-I [46] [protocol] [@ hostname | ipaddr] [:service | port]
```

where 4 represents IPv4 protocol and 6 represents IPv6 protocol.Protocol specifies the transport layer protocol, which can be TCP or UDP. Hostname specifies the host name and Ipaddr specifies the IP address of the host. Service specifies the service name and Port specifies the port number respectively.

For example, to display the socket file descriptors of all ssh services connected to the host 192.168.1.108, you can use the command.

```
root@server : $lsof-i@192.168.1.108:22
```

If no parameters are specified after the -i option, the lsof command will display all socket file descriptors.-u, displays all file descriptors opened by all processes started by the specified user and -c, displays all file descriptors opened by the specified command.

For example, to see which file descriptors are opened by the websrv program, you can use the following command.

```
root@server :$ lsof-c websrv-p
```

To display all file descriptors opened by the specified process use the man command. -t, only the PID of the process with the target file descriptor turned on is displayed.We can also directly take the file name as an argument of the lsof command to see which processes have opened the file.

It can be seen from the code listing that the program files and dynamic libraries on the test machine are stored in the equipment "8,3". Where "8" indicates that this is a SCSI hard disk;"3" indicates that this is the third partition on the hard disk, namely sda3.

The equipment corresponding to standard input, standard output and standard error output of websrv program is "136,3".Among them, "136" indicates that this is a fake terminal;"3" indicates that it is the third pseudo terminal, i.e. /dev/pts/3.

For FIFO-type files, such as pipes and socket, this field will display the address of a kernel reference target file, or its inode number.If the field is displayed as "0t" or "0x", it means that it is an offset value, otherwise it means that it is a file size.It is meaningless to define the file size for a character device or FIFO type file, so this field will display an offset value.

For socket, it shows the protocol type, such as "TCP".Name, the name of the file.If we use the telnet command to initiate a connection to the websrv server, when we execute the lsof command again,The following line will be added to its output.

```
websrv6346shuang5u IPv4 442880t0tcp localhost: 13579-> localhost: 48215 (ESTABLISHED).
```

This output indicates that the server has opened a socket of IPv4 type with a value of 5 and it is in established state.The local socket address of the connection corresponding to this socket is (127.0.0.1, 13579), and the remote socket address is (127.0.0.1, 48215).

While it may not have advanced capabilities like tcpdump, lsof still serves the purpose it is made for. In the next section of this chapter we will talk about nectar in detail. Follow along!

NC(NETCAT)

netcat command is short, capable and powerful, and has the reputation of "Swiss Army Knife".It is mainly used to quickly build network connections.We can make it run as a server, listen to a port and receive client connections, so it can be used to debug client programs.We can also make it run as a client, initiate a connection to the server and send and receive data, so it can be used to debug the server program, which is a bit like a telnet program.

The transport layer protocol used by nc command by default is TCP protocol.-w, if the nc client does not detect any input within the specified time, exit.-x, which specifies the communication protocol used between nc client and proxy server when they communicate.At present, the proxy protocols supported by nc include "4" (socks v.4), "5" (socks v.5) and "connect" (https proxy).The proxy protocol used by nc by default is SOCKS v.5.-x, specify the IP address and port number of the target proxy server.

For example, to connect to squid proxy server on laptop from source and access Web services in www.google.com through it, you can use the following command.

root@server : $ NC-x source: 1080-x Connect www.google.com 80-z, and scan whether one or some services on the target machine are turned on (port scanning).

For example, to scan services with port numbers between 20 and 50 on the machine ernest-laptop, you can use the following command.

```
root@server : $nc-z source 20-50.
```

For example,We can use the following methods to connect to the websrv server and send data to it.

```
root@server : $nc-C 127.0.0.1 13579 (server listening port 13579)
gethttp://localhost/a.htmlHTTP/ 1.1 (carriage return)
```

It will provide with requested file was not found on this server. Here we use the -C option, so that every time we press the enter key to send a line of data to the server, the nc client program will send an extra < Cr > < lf > to the server, which is exactly the HTTP line terminator expected by websrv server.

After sending the third line of data, we got the response from the server, which is exactly what we expected. The server did not find the requested resource file a.html.Therefore, nc command is a convenient and quick testing tool, through which we can quickly find out the logic errors of the server.

In the next section of this chapter, we will discuss about strace tool that can conveniently help us to track the routes of the network packets. Follow along!

STRACE

Strace is an important tool for testing server performance.It tracks the system calls and signals received during the program running, and outputs the system call names, parameters, return values and signal names to the standard output or specified files.

The format is:

```
[qualifier=][!]value1[,value2]...。
```

The qualifier can be one of trace, abbrev, verbose, raw, signal, read and write, and the default is trace.Value is a symbol or numerical value used to further limit the tracked system calls.Its two special values are all and none, which respectively mean to track all system calls of the type specified by qualifier and not track any system calls of this type.

For example, -e read=3, 5 means to output all data read from file descriptors 3

and 5.-o, writes strace's output to the specified file.Each line output of strace command contains these fields: system call name, parameter and return value.

For example, the following example:

```
root@server :$ stracecat/dev/nullopen ("/dev/null", rdonly | largefile) = 3.
```

This line of output indicates that the program "cat/dev/null" executed the open system call during running.

Open call opened the large file /dev/null in a read-only manner, and then returned a file descriptor with a value of 3.It should be noted that this example command will output a lot of content, and here we omit a lot of secondary information. In the following examples, we only display the content related to the topic.

When there is an error in the system call, the strace command will output the error identification and description, such as the following example.

```
root@server : $ stracecat/foo/baropen ("/foo/bar", rdonly | large file) =-1
```

This will display the output as enoent (no such file or directory)

strace command will have different output modes for different parameter types, for example, for C-style strings we will get a different output depending on its parameter. This helps us to monitor data in a dynamic way using the strace command utility tool.

The default maximum output length is 32 bytes, and the excessive strace will be omitted with "…".

For example, the ls-l command will read the /etc/passwd file:

```
root@server : $ stracels-lread (4, "root: x: 0: 0: root:/root:/bin/bash \ n" ..., 4096) = 2342.
```

It should be noted that the file name is not regarded as a C-style string by strace. For a structure, strace will output each field of the structure with "{}" and

separate each field with ",".For structures with more fields, strace will omit some output with "…".

For example:

```
root@server : $ stracels-l/dev/null lstat64 ("/dev/null", {mode = ifchr | 0666, rdev = makedev (1,3), ..}) = 0
```

The strace output above shows that the first parameter of lstat64 system call is the second parameter and is the output parameter (pointer) of stat structure type. strace only shows two fields of the structure parameter: mode and rdev.It should be noted that when the system call fails, the output parameters will be displayed as the values before passing in.

For bit set parameters (such as signal set type sigset), strace will output all bits set to 1 in the set with "[]", and separate each item with a space.

Suppose there is the following code in a program:

```
sigaddset( &set,SIGUSR1);sigprocmask(SIGBLOCK, &set,NULL);
```

Then strace command for this program will output the following contents:

```
rt sigprocmask (SIG block, [quitusr1], null, 8) = 0.
```

For the output modes of other parameter types, please refer to strace's man manual, and do not repeat them here.

For the signal received by the program, strace will output the value of the signal and its description.For example, we run the "sleep 100" command on one terminal, then use the strace command to track the process on another terminal, and then use "Ctrl+C" to terminate the "sleep 100" process to observe the output of strace.

The specific operation is as follows:

```
root@server : $ sleep 100 $ PS-EF | grep sleep
```

The specific operation is as follows:

```
root@server : $./websrv 127.0.0.1 13579 $ PS-EF | grep websrv
```

Next, use the method to initiate a connection to the server and send an HTTP request. Therefore, this event indicates that a new client connection is coming, so the websrv server makes an accept call to the listening socket, and accept returns a new connection socket with a value of 5.

Then, the server clears the errors on the new socket, sets its REUSEADDR attribute, then registers EPOLLRDHUP and EPOLLONESHOT events on the socket in the epoll kernel event table, and finally sets the new socket as non-blocking.

The wait call detected the epoll event on file descriptor, which indicated that the first line of data from the client had arrived, so the server executed two recv system calls to receive the data.The first recv call reads 38 bytes of customer data, that is, "gethttp://localhost/a.htmlhttp/1.1 \ r \ n".The second call to recv failed, and errno is EAGAIN, which means there is no more customer data to read at present.

After that, the server called futex function to unlock the mutex to wake up the thread waiting for the mutex.It can be seen that pthread_mutex_unlock function in POSIX thread library calls futex function internally.The contents of the third and fourth parts are similar to those of the second part, so we will not repeat them here.

In the fifth part, the wait call detects the EPOLLOUT event on file descriptor 5, which indicates that the worker thread correctly processed the client request and prepared the data to be sent, so the main thread started to execute the writev system call and write the HTTP response to the client.

Finally, the server removes all registered events on file descriptor 5 from the epoll kernel event table and closes the file descriptor.Therefore, strace command enables us to clearly see the timing of each system call and the values of related parameters, which is more convenient than debugging with gdb.

With this, we have described a brief and complex introduction to strace command. We recommend you to read it with attention. While there may be different ways to understand the importance of network packets in strace we should also use other tools such as netstat to further improve the impact of network monitoring tools as a hacker.

NETSTAT

Netstat is a powerful statistical tool for network information.It can print all connections, routing table information and network card interface information on the local network card interface.For this book, we mainly use the first of the above functions, that is, display TCP connection and its status information.

After all, to get routing table information and NIC interface information, we can use route and ifconfig commands with richer output content.

Next, we run the websrv server and execute the telnet command to initiate a connection request to it.

```
root@server : $./websrv 127.0.0.1 13579 & $ telnet 127.0.0.1 13579
```

And then execute the command to check the connection status.

```
netstat-NAT | grep 127.0.0.1: 13579
```

The result is as follows:

```
protorecv-qsend-q local address foreign address state
TCP 0 0 127.0.0.1: 13579 0.0.0.0:
```

In the above output, line 1 indicates that the local socket address 127.0.0.1:13579 is in the LISTEN state, and waits for any remote socket (represented by 0.0.0.0:) to initiate a connection to it.Line 2 indicates that the server has established a connection with the remote address 127.0.0.1:48220.

The third line only repeatedly outputs the connection indicated by the information in the second line from the client's point of view, because we are running the server program (websrv) and the client program (telnet) on the same machine.In the process of server program development, we must ensure that every connection is in the state we expect at any moment.Therefore, we should get used to using netstat command.

We can display a lot of information about different network servers, their traffic packets and other important information using the netstat command. In the next section, we will talk about other fundamental networking tool which is called as vmstat. Follow along!

VMSTAT

Vmstat is the abbreviation of virtual memory statistics, which can output the usage of various resources of the system in real time, such as process information, memory usage, CPU usage and I/O usage.

By default, the output of vmstat is quite rich.

Look at the following example:

```
root@server : $vmstat 5 3

# outputs the results every 5 seconds.
```

Note that the output in the first line is the average result since the system was started, while the output in the next line is the average result in the sampling interval.

If these two values change frequently, there is not enough memory.io, the usage information of block devices, and the unit is block/s. bi indicates the rate at which blocks are read from the block device;"bo" indicates the rate at which blocks are written to the block device."in" indicates the number of interrupts per second and “cs" indicates the number of context switches (process switch) per second.

However, we can use iostat command to get more information about disk usage, and we can also use mpstat to get more information about CPU usage.The

vmstat command is mainly used to check the usage of system memory.

A lot of hackers use this to understand about memory resources and temporarily stop those programs that are consuming high memory. While performing tasks like brute force attacks we need to understand the importance of these tools that can decrease the work load of the system. In the next section, we will talk about ifstat networking tool in detail. Follow along!

IFSTAT

Ifstat is the abbreviation of interface statistics, which is a simple network traffic monitoring tool.

For example, we execute the following command on the test machine:

```
root@server : $ifstat-a 2 5

# outputs the result every 2 seconds.
```

output of ifstat shows the rate of receiving and sending data on each NIC interface in KB/s.Therefore, using ifstat command can roughly estimate the total input and output traffic of the server in each period.

In the following section of this chapter, we will talk about mpstat which is a popular networking tool for hackers to analyze different processor statistics. Follow along!

MPSTAT

Mpstat is the abbreviation of multi-processor statistics, which can monitor the usage of each CPU on multi-processor system in real time.The mpstat command and iostat command are usually integrated in the package sysstat, which can be obtained by installing sysstat.

The typical usage of the mpstat command is (there are not many options of the mpstat command, which are not specificALLy introduced here)

```
mpstat [-p {| all}] [interval [count]]
```

option p specifies the CPU number to be monitored (0 ~ CPU number -1), and its value "all" means to monitor all CPUs.

The interval parameter is the sampling interval (in s), that is, the statistical information is output once every interval.Count parameter is the number of sampling times, that is, a total of count times of statistical information are output, but mpstat will finally output the average value of these count times of sampling results.

Like the vmstat command, the first output result of the mpstat command is the average result since the system was started, while the next (count-1) output result is the average result within the sampling interval.

For example, we execute the following command on the test machine:

```
root@server : $mpstat-P ALL 5 2
# outputs the result every 5 seconds.
```

The output of each sampling contains three pieces of information, and each piece of information contains the following fields: CPU, which indicates which CPU the piece of information is."0" indicates the data of the first CPU, "1" indicates the data of the second CPU, and "all" indicates the average value of the data of the two CPUs.

% usr, except for the process with negative nice value, the proportion of the time that other processes on the system run in user space to the total CPU running time.

% nice, the proportion of the time that a process with a negative nice value runs in user space to the total CPU running time.

% sys, the ratio of the time that all processes on the system run in kernel space to the total CPU running time, but excluding the CPU time consumed by hardware and software interrupts.

% iowait, the proportion of CPU waiting for disk operation to total CPU running time.% IRQ, the proportion of CPU time spent processing hardware interrupts to the total CPU running time.

% soft, the proportion of CPU time spent processing software interrupts to the total CPU running time.

% steal, a physical CPU can contain a pair of virtual CPUs, which are managed by the hypervisor.When a hypervisor is processing a virtual CPU, another virtual CPU must wait for its processing to finish before running.This waiting time is called steal time.The field indicates the proportion of steal time to the total CPU running time.

%guest, the time spent running virtual CPU accounts for the total CPU running time.

% idle, the ratio of system idle time to total CPU running time.Among all these output fields, we are most concerned about %user, %sys and %idle.

They basically reflect the proportion of business logic code and system call in our code, and how much load the system can bear.Obviously, in the above output, executing system calls takes more CPU time than executing user business logic.This is because we have run the stress testing tool introduced on this machine, which constantly executes recv/send system calls to send and receive data.

WHAT NEXT?

With this, we have provided a perfect introduction to a couple of network monitoring tools that can help you understand the essence of hacking. To become a hacker you need to personally try out all these commands you have learnt in this chapter on your own networks. If not, anything that you do cannot be understood. Linux is not about using the written commands but it is about using commands in a definite way that can change the way system interacts with the hardware resources and software kernel.

In the next chapter of this module we will discuss about different Kali Linux tools in detail. Follow along!

CHAPTER FOUR
KALI LINUX TOOLSET

KALI LINUX CONSISTS of more than 350 hacking tools pre installed. All these tools provide different use cases and can help hackers spoof network data and can even monitor and document them for future proof cases.

In this chapter we will discuss about some of the most important tools that are available in Kali Linux along with what they do. Grab your seat and have fun exploring different branches of hacking.

STAGE 1 - RECONNAISSANCE TOOLS

As said before to start cracking and hacking systems you need to first have a lot of information about the target. For example, knowing what server architecture the target is using can help you find vulnerabilities for that particular server architecture.

But, how do you actually know about the target?

To explain this we need to explain how hackers work first. Hackers are not only smart geeks but are also good conversationalists that try to acquire information from normal conversations from the users that are using the target.

If a hacker is looking to target a bank server then he persuades few of the bank employees and tries to gather information about the server. While it may seem fishy but just installing a monitoring software can help them get all the resources and information they need.

For this reason we are dividing this branch of hacking into two categories. One is social engineering attacks and the other is digging the valuable information.

A) Social engineering attacks

These are the attacks that hackers use to the target users, employees and system administrators. One of the famous attacks is to create a phishing page for expecting the target to enter their username and password. After obtaining the credentials they use this privilege to reach to the root level.

There are tons of tools for these type of attacks. There are different command line utilities to create fake phishing pages of social networking websites.

B) Digging the valuable information

A lot of information about the target is usually available in the open world. A couple of intelligent google searching can give you tons of information about the target. Reaching out to the system administrators using e-mail can also be a good way to get some sensitive information.

However, the most important tool hackers use for this hacking step is nmap. nmap is an open-source tools that scans the open ports that are available and outputs tons of information that is related to the target.

If done correctly you can find the server architecture and the operating system the target using with different automated commands. However, remember that the internet security researchers have learned a lot in these years and have developed different intrusive detection systems to counter the requests that are coming form newbies who are trying to scan the open ports.

If you are an experienced hacker the only thing you will do is to not draw attention from the system administrators to ban your IP address or MAC address permanently.

Here are some commands:

```
root@sample : nmap
// This displays the menu
root@ sample : nmap www.google.com
// This scans the open ports of this particular address
root@sample : nmap -V 192.32.222.12
// This sends requests for open port detection for this input address
```

Nmap is also very efficient to know about other advanced stuff such as pinging time and the presence of firewalls. If there are firewalls present you can find open vulnerabilities to exploit them.

After scanning and finding potential loopholes you can proceed to the next stage of hacking. That is exploitation.

2) EXPLOITATION TOOLS

What is fun in hacking systems if you don’t know a way to hack them and exploit them?

After finding vulnerabilities hackers usually insert trojans or spider worms to complete the dirty work for them. A lot of hackers now are also implementing bitcoin mining in different systems to make some quick bucks. But how do they do it?

Experienced hackers write their own software for exploits whereas moderate hackers use kali linux tools such as Metasploit.

What is Metasploit?

Metasploit is the most popular exploitation tool and can be used for tracking the users or for binding trojans in an image or pdf files. And when the target clicks the metas-loot modified file we can track them and exploit them.

Here are some command:

```
root@server : metasploit
// This is will start the msf console
root@server : msf bind image.jpg
// This binds the image file with metasploit trojan code
```

You can also use metasploit to hack servers and mobile devices. However, remember that you can’t hack an apple device without hardware access whereas you can hack android files even remotely.

STAGE 3 - MONITORING TOOLS

In the previous stages we used Kali Linux tools to scan and exploit the systems. In this section we will learn about Monitoring tools such as Wireshark to understand the hacker attacks that may happen anytime.

What is Monitoring?

Network data travel in the form of packets. Packets if not encrypted can be spoofed and can be used to steal information by hackers. Tools like Wireshark can help ethical hackers to monitor the packets to analyze any suspicious traffic that is coming.

Wireshark is one of the most downloaded packet analysis software. It provides a pane where every packet, its headers and the host information will be displayed. You can analysis these records from the logs that wireshark leave.

Here are some tips to use Wireshark:

1) Always make sure that you are aware of the target host address you are trying to monitor.

2) Use utility tools such as net stat and ping to analyze the network traffic and use this information to capture the packets effectively.

3) Understand the difference between encrypted and non-encrypted network packets. If you are forced to decrypt the encrypted packets then make sure that you understand other complex stuff such as HTTP protocols and cookie information.

Wireshark provides premium features for network administrators for a nominal price. In the professional version you can automatically make reports using the captured information where as in community version you need to manually make them.

DICTIONARY ATTACK TOOLS

While there is a ton of complex stuff for hackers to deal with what they are mostly interested in is to steal sensitive information such as passwords and email addresses. The only way to steal a lot of passwords and accounts at once is by using dictionary attacks. Also known as brute force attacking in technical terms.

Kali linux provides tools such as Hydra and Jack the Ripper to crack websites

and databases. Some crackers also exploit SQL injection to crack databases to obtain sensitive information.

A) Hydra

Hydra is a dictionary attack tool that is available in kali linux. Using hydra all you need to do is provide the url page and a text file that consists of username and passwords separated by a colon.

All the successful logins will be printed in a log file and can be used to login to the website.

Here are some commands:

```
root@server : hydra www.google.com/login user.txt

// This will attack the login page with the information provided in text file
```

B) Jack the Ripper

Jack the Ripper is same type of tool which can be used to crack login pages. While hydra can brute force only static login pages this tool can help you crack login pages dynamically.

That is, you can customize how a brute force attack can be performed according to the ongoing results. For example, if the login page is giving a “ Server not found “ errors then Jack the Ripper will automatically change the proxy servers to mitigate a proxy ban that is performed by intrusive detection systems.

Here is a command:

```
root@server : jtr -w pass.txt www.samepleurl.com -p proxy.txt
```

Here w stands for the wordlist that carries the username and password where as proxy.txt has information about different proxy servers that can be randomly used if they are live by the brute force engine.

WIRELESS NETWORK TOOLS

The last set of tools we are going to discuss is about wireless tools. We all know that the world is rapidly changing and network ethernets are now not the only way to transmit information in a network. Wireless networks include wifi hotspots , mobile phones, laptops and even smart watches.

There is a lot of privacy at stake and even a small vulnerability in the network protocol can be used to steal a lot of sensitive information. This is the reason why you need to update your system regularly.

Here are some of the important tools :

A) aircrack-ng

This is a famous wireless network tool that can be used to brute force all the available Wpa/wp2 networks. It also can automatically find vulnerabilities in the WPS network. All you need to do is to find the network address of the Wireless network using net stat command and insert in the tool to crack them automatically.

Here are commands:

```
root@server : aircrack-ng 192.23.11.234 text.txt
```

B) Airdump

Airdump is a network monitoring tool that does the same work as the above mentioned tool but with more options and customizations. Airodump even provided fake phishing pages to obtain information from the target users using social engineering attacks.

With this, we have completed a brief introuction to Kali Linux tool set. In the next chapter of this module we will talk about different network monitoring tools that can help to analyze the networks and hack them. Let us go!

CHAPTER FIVE

BURPSUITE FOR PENETRATION TESTING

IN THE PREVIOUS chapters we have talked about network monitoring tools that can help us to understand different network modules that are present in the target system. While Network hacking is a popular hacking procedures until few years back now due to the increase of internet and web applications we are now on a verge to getting stolen just by browsers.

WHAT ARE WEB APPLICATIONS?

Web applications are software that run in a browser and can be accessed with just an URL. For example www.gmail.com is a web application that can be used to send messages in an Email across the world to whoever has an Email address.

It is very easy to find bugs in a web application for experienced programmers. For finding these bugs they use certain software like Burp suite. Burp suite is at this point the most popular web application vulnerability finder software. It comes in both community and professional versions. Community version is free and will not provide automatic scanning whereas professional version is available for a nominal price and provides automatic spidering and scanning of targets with documentation availability for enterprises. Burp-suite also has a plugin store that consists of tons of free and premium add-ons to make web application penetration tester life much easier.

HOW TO USE BURPSUITE?

To install Burp-suite head on to the official website and install it after giving permission using the chmod command.

If you are using Kali Linux then you are lucky because burp suite community edition is pre installed in this hacking distro. Open Burp-suite from the “ Web application’ Menu bar in the start menu.

After opening Burp-suite follow these instructions:

1) Use the proxy options and enter ’127.0.0.1” in both Firefox and burp suite as host address. By doing this procedure all the HTTP requests that are coming from the browser will be first sent to burp suite.

2) You can also be able to run HTTPS protocol requests using the certificate that is present from the port swigger security website.

3) After receiving requests you need to enter “ Accept’ or “ Reject” to send it to the Burp spidering. You can also enter the website url to make burp spiders scan the whole website. However be realistic when the websites are scanned because you will not get fascinating results with the scanners. To find vulnerabilities you need to a much harder digging and a lot of testing.

4) Burp-suite also has other interfaces such as Burp suite intruder, Burp suite comparer to automate brute force attacks and analyzing cookie information.

5) Burp-suite also provides a documentation panel where you can write POC for the bugs that you have found.

In the next section, we will provide a detail instruction to hack online web page passwords using the Burp intruder option available in Burp-suite. We want you to carefully follow the exercise to understand how to crack accounts of any website you like. It is tricky sometimes but if done correctly you can have some fruitful results.

USING BURP INTRUDER TO CRACK PASSWORDS

Burp intruder is an important component in the Burp suite software. Using intruder you can individually send a request to brute force it with available dictionaries and options. Intruder can send both GET and POST requests and can easily be used to analyze the responses it get.

Using intruder security analysts often understand how well the intrusion detection systems are. A lot of other web penetration bugs such as XSS and CSRF attacks can also be detected using the Burp intruder component.

How to use it?

Step 1:

Find a webpage that you want to crack. It can be either HTTP or HTTPS protocol supported website. If you are not sure about the attacking restrictions the website is imposing then we suggest you to try with sample test websites that are developed specifically for the practice of web application penetration testers.

Step 2 :

Enter the webpage with burp intercept proxy in On. Within seconds you will get a request in the Burp interface. Just take a look at it and accept all the requests that appear. If you deny any requests sometimes the web page may not open properly. So, make sure that the webpage has appeared in the Burp site menu.

Step 3:

Now carefully observe all the requests the burp inspector has analysed. In those requests find out the requests that have a “POST” identifier. Normally, all the requests will have “GET” identifier because they are asking certain information from the server. Whereas login pages and other sensitive and vulnerable indexes such as comment boxes, New post boxes and login forms use “POST” identifier as they are sending information to the server.

Step 4:

After finding the login form that you are wishing to brute force left click on it so that a menu will appear. In that burp menu click on the option “ Send to burp intruder”. Now you can go to intruder interface to see that the selected page is all ready to perform a brute force (dictionary) attack using the options available.

Step 5:

Now, select the parameters using the HTML code that appears in the bottom box. This is where you select the “ login” and “password” boxes for easy attacking. After selecting enter the options menu and select the type of dictionary that you are willing to use.

From numbers to common passwords there are tens of options for you to consider. You can also upload your custom wordlist in professional version of Burp-suite. Also don’t forget to select the type of brute force attack you are willing to perform. I normally use “pitchfork” as it is simple and easy.

If you are not sure about the different type of brute force attacks that are available in Burp-suite we suggest you to visit port swigger online labs for more understanding. They are awesome.

Step 6:

Now click start and within seconds the requests will start flooding your results screen. After the brute force ends carefully observe the results. All the results with “404” errors are bad results. If you get any successful messages then they are potentially good results. Cross check the good results and make a documentation about the attack scenario.

With this, we have provided a clear cut instruction to Burp suite along with various important information.

WHAT NEXT?

Congratulations! You have now successfully completed the last module of this book. In these five modules you have learned a lot of new information that can help you to become an efficient hacker.

To make use of the knowledge you have acquired please write your own shell and python scripts that can automated brute forcing. Also, read tons of open-source code from Github so that you can challenge yourself with complexities that may arise while dealing with catastrophic situations as an ethical hacker.

Finally, a friendly suggestion. We are not here to talk about ethics. Do whatever it makes you happy. But it genuinely feels better to stop destroying things instead of destroying thing. Cheers and All the Best!

ACKNOWLEDGMENTS

This book is possible because of the tremendous effort by my editor DAN GUIND. He is an awesome gentleman who helped me edit and format the book.

Also, a large thanks to my cover designer David. He is just awesome.

And to all my family who made my life a lot better this book is for all of you.

ABOUT THE AUTHOR

TYE DARWIN is a cyber security specialist and has written tens of books about hacking. He is very accurate about all the tools that he learned and explains them with concise words. His first book “ Hacking for beginners” has become a successful book with more than thousand copies sold. He is a programmer by day and a penetration tester by night.