

THE NEW PENETRATING TESTING FOR BEGINNERS



WILFRED DAWSON

Contents

[Presentation](#)

[What is Penetration Testing?](#)

[Setting the Stage](#)

[Introduction to Kali and Backtrack Linux: Tools. Lots of Tools](#)

[Working with Your Attack Machine: Starting the Engine](#)

[The Use and Creation of a Hacking Lab](#)

[Phase's of a Penetration Test](#)

[Where Do I Go from Here?](#)

[FINAL THOUGHT](#)

Copyright ©2020.WILFRED DAWSON

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

Presentation

The goal of this book is to provide general knowledge that will allow you to tackle progressed points and books. When you have a strong handle of the basics, you can always go back and learn the specific subtleties and advanced highlights of a tool. What's more, each chapter will end with a list of suggested tools and topics that are outside the scope of this book however can be utilized for further study and to advance your information.

Past just being written for fledglings, this book actually presents the information in an interesting way. All the tools and techniques we use in this book will be carried out in a specific request against a small number of related targets (all target machines will have a place with the same subnet, and the reader will be ready to handily reproduce this "target" organization to follow along). Readers will be indicated how to decipher tool output and how to utilize that output to continue the attack from one part to the next. The book will cover both local and remote assaults as well as a discussion of when each is appropriate.

The use of a consecutive and singular moving example all through the book will help readers see the enormous picture and better comprehend how the various devices and phases fit together. This is different than many other books on the market today, which often discuss various instruments and assaults yet fail to explain how those tools can be effectively fastened together. Presenting data in a way that shows the client how to clearly move from one phase to another will give valuable experience and permit the reader to complete a whole penetration test by simply following along with the models in the book. This concept should allow the peruser to get an unmistakable understanding of the fundamental information while learning how the different tools and phases interface.

What is Penetration Testing?

Penetration testing can be characterized as a lawful and authorized endeavor to find and successfully misuse PC systems for the purpose of making those systems more secure. The process incorporates testing for vulnerabilities as well as providing evidence of concept attacks to exhibit the vulnerabilities are real. Proper penetration testing always closes with

specific recommendations for addressing and fixing the issues that were discovered during the test. On the entire, this process is used to help secure PCs and organizations against future assaults. The general idea is to discover security issues by utilizing the same instruments and techniques as an assailant. These findings can then be mitigated before a real programmer exploits them.

Infiltration testing is also known as:

- Pen testing
- PT
- Hacking
- Ethical hacking
- White cap hacking
- Offensive security
- Red joining.

It is important to spend a couple of seconds examining the difference between penetration testing and weakness assessment. Many individuals (and vendors) in the security community incorrectly use these terms conversely. A weakness assessment is the process of reviewing services and systems for potential security issues, whereas a penetration test really performs exploitation and Proof of Concept (PoC) attacks to prove that a security issue exists. Penetration tests go a step beyond vulnerability assessments by simulating hacker activity and delivering live payloads. In this book, we will cover the measure of vulnerability assessment as one of the means used to complete an infiltration test.

Setting the Stage

Understanding all the different players and positions in the world of hacking and infiltration testing is central to appreciating the big picture. Let us start by painting the picture with broad brush strokes. Please understand that the

following is a gross oversimplification; in any case, it should help you see the differences between the different gatherings of people involved.

It might help to consider the Star Wars universe where there are two sides of the "power": Jedis and Siths. Great vs Evil. Both sides have access to an incredible power. One side utilizes its power to secure and serve, whereas the other side uses it for personal gain and exploitation.

Figuring out how to hack is much like learning to use the force (or so I envision!). The more you learn, the more power you have. Eventually, you will have to choose whether you will use your power for good or terrible. There is an exemplary poster from the Star Wars Episode I movie that depicts Anakin as a young kid. On the off chance that you look closely at Anakin's shadow in the poster, you will see it is the outline of Darth Vader. Try looking the Internet for "Anakin Darth Vader shadow" to see it. Understanding why this poster has appeal is critical. As a kid, Anakin had no aspirations of becoming Darth Vader, however it happened in any case.

It is likely safe to accept that very few individuals get into hacking to turn into a super villain. The problem is that journey to the dim side is a slippery slope. Notwithstanding, on the off chance that you want to be great, have the regard of your friends, and be productively utilized in the security labor force, you need to submit yourself to utilizing your forces to protect and serve. Having a felony on your record is a one-way ticket to another calling. It is genuine that there is right now a shortage of qualified security specialists, however even thus, not many businesses today are willing to take a possibility, especially if those crimes involve PCs. The guidelines and restrictions become even more tough if you need a PC job which requires a security leeway.

In the pen testing world, it is normal to hear the terms "white hat" and "dark hat" to describe the Jedis and Siths. Throughout this book, the terms "white cap", "moral programmer", or "entrance tester" will be utilized interchangeably to describe the Jedis or good folks. The Siths will be referred to as "black hats", "crackers", or "malicious assailants".

It is significant to note that ethical hackers complete many of the equivalent activities with numerous of the same tools as malignant attackers.

In nearly each situation, a ethical hacker should strive to act and think like a genuine black hat hacker. The closer the penetration test simulates a certifiable assault, the more value it provides to the customer paying for the penetration testing (PT).

Please note how the previous passage says "in nearly each circumstance". Even though white caps total many of the same assignments with many of the same tools, there is a world of difference between the different sides. At its center, these differences can be boiled down to three key points: approval, motivation, and intent. It should be focused on that these focuses are not all inclusive, yet they can be valuable in determining if a activity is ethical or not.

The first and easiest approach to differentiate between white hats and black caps is approval. Authorization is the process of obtaining approval before conducting any tests or attacks. Once authorization is got, both the penetration analyzer and the company being audited need to agree upon the scope of the test. The scope includes specific information about the assets and systems to be remembered for the test. The degree explicitly defines the authorized targets for the entrance analyzer. It is important that both sides completely understand the authorization and extent of the PT. White hats must always respect the authorization and remain within the scope of the test. Black hats will have no such imperatives on the target list.

Note: Clearly characterizing and understanding the scope of the test is crucial. The scope formally defines the standards of engagement for both the entrance tester and the client. It ought to incorporate an objective rundown as well as specifically posting any systems or attacks which the client doesn't want to be included in the test. The scope ought to be recorded and signed by authorized personnel from both the testing team and the client. Occasionally, the scope will need to be amended during a penetration test. When this occurs, be sure to update the scope and leave before proceeding to test the new targets.

The second way to differentiate between an moral hacker and a malicious hacker is through examination of the attacker's inspiration. In the event that the attacker is motivated or driven by close to home gain, including profit through coercion or other naughty methods of gathering cash from the

victim, revenge, fame, or the like, the individual should be considered a black hat. Be that as it may, if the attacker is preauthorized and his or her motivation is to help the association and improve their security, he or she can be considered a white cap. In addition, a black cap hacker may have a critical amount of time focused on attacking the association. In most cases, a PT may last multi week to several weeks. Based on the time allocated during the PT, a white hat may not have discovered more advanced time-intensive exposures.

Finally, if the intent is to give the organization a realistic attack reproduction so that the company can improve its security through early discovery and mitigation of vulnerabilities, the attacker ought to be considered a white hat. It is also significant to comprehend the critical nature of keeping PT findings classified. Moral hackers will never share delicate information discovered during the process of a penetration testing with anyone other than the client. However, if the intent is to leverage information for personal profit or acquire, the attacker should be considered a black cap.

It is also critical to understand that not all penetration tests are carried out in the same manner or have the same purpose. White box penetration testing, also known as "overt" testing, is very thorough and thorough. The goal of the test is to examine every alcove and cranny of the target's framework or network. This kind of test is valuable in assessing the overall security of an organization. Because stealth isn't a worry, many of the devices we will inspect all through this book can be run in verbose mode. By disregarding secrecy in favor of thoroughness the infiltration tester is often capable to discover more weaknesses. The downside to this type of test is that it does not provide a very accurate simulation of how most present day, talented aggressors exploit networks. It likewise does not provide a chance for the organization to test its incident reaction or early-alert systems. Keep in mind, the analyzer is not trying to be stealthy. The tester is attempting to be thorough.

Black box penetration testing, also known as "secretive" testing, employs a fundamentally different procedure. A discovery test is a much more realistic simulation of the way a skilled aggressor would attempt to access the target frameworks and network. This type of test trades thoroughness and the ability to detect various weaknesses for stealth and pin-point precision.

Black box testing typically only requires the tester to find and exploit a single vulnerability. The advantage to this type of test is that it more intently models how a real-world attack takes place. Not many attackers today will filter all 65,535 ports on a target. Doing so is uproarious and will almost certainly be detected by firewalls and interruption discovery frameworks. Skilled malicious hackers are much more discrete. They may only examine a single port or interrogate a single administration to find a way of compromising and

Introduction to Kali and Backtrack Linux: Tools. Lots of Tools

A few years back, the open discussion or teaching of hacking techniques was considered a bit taboo. Fortunately, times have changed and people are beginning to understand the value of offensive security. Offensive security is now being embraced by organizations regardless of size or industries. Governments are also getting serious about offensive security. Many governments have gone on record stating they are actively building and developing offensive security capabilities.

Ultimately, penetration testing should play an important role in the overall security of your organization. Just as policies, risk assessments, business continuity planning, and disaster recovery have become integral components in keeping your organization safe and secure, penetration testing needs to be included in your overall security plan as well. Penetration testing allows you to view your organization through the eyes of the enemy. This process can lead to many surprising discoveries and give you the time needed to patch your systems before a real attacker can strike.

One of the great things about learning how to hack today is the plethora and availability of good tools to perform your craft. Not only are the tools readily available, but many of them are stable with several years of development behind them. May be even more important to many of you is the fact that most of these tools are available free of charge. For the purpose of this book, every tool covered will be free.

It is one thing to know a tool is free. It is another to find, compile, and install each of the tools required to complete even a basic penetration test. Although this process is quite simple on today's modern Linux operating systems (OSs), it can still be a bit daunting for newcomers. Most people

who start are usually more interested in learning how to use the tools than they are in searching the vast corners of the Internet to locate and install tools.

To be fair, you really should learn how to manually compile and install software on a Linux machine; or at the very least, you should become familiar with apt-get (or the like).

More Advanced: Advanced Package Tool (APT) is a package management system. APT allows you to quickly and easily install, update, and remove software from the command line. Aside from its simplicity, one of the best things about APT is the fact that it automatically resolves dependency issues for you. This means that if the package you are installing requires additional software, APT will automatically locate and install the additional software. This is a massive improvement over the old days of “dependency hell”.

Installing software with APT is very straightforward. For example, let us assume you want to install a tool called Paros Proxy on your local Linux machine. Paros is a tool that can be used (among other things) to evaluate the security of web applications. We will discuss the use of a proxy in the Web Based Exploitation chapter but for now let us focus on the installation of the tool rather than its use. Once you know the name of the package you want to install, from the command line you can run: apt-get install followed by the name of the software you want to install. It is always a good idea to run: apt-get update before installing software. This will ensure that you are getting the latest version available. To install Paros, we would issue the following commands:

```
Apt-get update
```

```
Apt-get install paros
```

Before the package is installed, you will be shown how much disk space will be used and you will be asked if you want to continue. To install your new software, you can type “Y” and hit the enter key. When the program is done installing you will be returned to the # prompt. At this point you can start Paros by entering the following command into the terminal:

Paros

For now, you can simply close the Paros program. The purpose of this demo was to cover installing new software, not in running or using Paros.

If you prefer not to use the command line when installing software, there are several Graphical User Interfaces (GUIs) available for interacting with APT. The most popular graphical front end is currently aptitude. Additional package managers are outside the scope of this book.

One final note on installing software, APT requires you to know the exact name of the software you want to install before running the install command. If you are unsure of the software name or how to spell it, you can use the apt-cache search command. This handy function will display any packages or tools which match your search and provide a brief description of the tool. Using aptcache search will allow you to quickly narrow down the name of the package you are looking for. For example, if we were unsure of the official name of the Paros package from our previous example, we could have first run:

Apt-cache search Paros

After reviewing the resulting names and descriptions, we would then proceed with the apt-get install command.

Please note, if you are using Kali Linux, Paros will already be installed for you! Even so, the apt-get install command is still a powerful tool for installing software.

A basic understanding of Linux will be beneficial and will pay you mountains of dividends in the long run. For the purpose of this book, there will be no assumption that you have prior Linux experience, but do yourself a favor and commit yourself to becoming a Linux guru someday. Take a class, read a book, or just explore on your own. Trust me, you will thank me later. If you are interested in penetration testing or hacking, there is no way of getting around the need-to-know Linux.

Fortunately, the security community is a very active and very giving group. There are several organizations that have worked tirelessly to create various

security-specific Linux distributions. A distribution, or “distro” for short, is basically a flavor, type, or brand of Linux.

Among the most well-known of these penetration testing distributions is one called “Backtrack”. Backtrack Linux is your one-stop shop for learning hacking and performing penetration testing. Backtrack Linux reminds me of a scene from the first Matrix movie where Tank asks Neo “What do you need besides a miracle?” Neo responds with “Guns. Lots of Guns”. At this point in the movie, rows and rows of guns slide into view. Every gun imaginable is available for Neo and Trinity: handguns, rifles, shotguns, semiautomatic, automatic, big and small from pistols to explosives, an endless supply of different weapons from which to choose. That is a similar experience most newcomers have when they first boot up Backtrack or Kali Linux. “Tools. Lots of Tools”.

Backtrack Linux and Kali Linux are a security tester’s dream come true. These distributions are built from the ground up for penetration testers. They come preloaded with hundreds of security tools that are installed, configured, and ready to be used.

Note: Best of all, Kali and Backtrack are free!

Additional Information: In the spring of 2013, the Offensive Security crew released a redefined, re-envisioned version of Backtrack called “Kali Linux”. Like Backtrack, Kali Linux is freely available and comes preconfigured with loads of security auditing tools. Kali can be downloaded from www.kali.org. If you are new to the penetration testing and hacking world, the differences between Backtrack and Kali may seem a bit confusing. However, for understanding the basics and working through the examples in this book, either distribution will work. In many cases, Kali Linux may be easier to utilize (than Backtrack) because each of the tools are “built into the path” meaning they can be run from anywhere. Simply, open a terminal and enter the tool name along with the desired switches. If you are using Backtrack, you often need to navigate to the specific folder before running a particular tool. If all this talk about navigating, paths, switches, and terminals sounds confusing, do not worry. We will cover everything in the coming chapters. For now, you simply need to decide

which version you would like to learn with. Kali or Backtrack. Remember, there is no wrong choice.

Navigating to the Backtrack (or Kali) link will allow you to choose from either an .iso or a VMware image. If you choose to download the .iso, you will need to burn the .iso to a DVD. If you are unsure of how to complete this process, please Google “burning an iso”. Once you have completed the burning process, you will have a bootable DVD. In most cases, starting Linux from a bootable DVD is as simple as putting the DVD into the drive and restarting the machine. In some instances, you may have to change the boot order in the BIOS so that the optical drive has the highest boot priority.

If you choose to download the VMware image, you will also need software capable of opening and deploying or running the image. Luckily enough, there are several good tools for accomplishing this task. Depending on your preference, you can use VMware’s VMware Player, Sun Microsystem’s VirtualBox, or Microsoft’s Virtual PC. In reality, if you do not like any of those options, there are many other software options capable of running a virtual machine (VM) image. You simply need to choose one that you are comfortable with.

Each of the three virtualization options listed above is available free of charge and will provide you with the ability to run VM images. You will need to decide which version is best for you. This book will rely heavily on the use of a Backtrack VMware image and VMware Player. At the time of writing, VMware Player was available at <http://www.vmware.com/products/player/>. You may need to register for an account to download the software, but the registration process is simple and free.

If you are unsure if you should use a live DVD or VM, it is suggested that you go the VM route. Not only is this another good technology to learn, but using VMs will allow you to set up an entire penetration testing lab on a single machine. If that machine is a laptop, you essentially have a “travelling” PT lab so you can practice your skills anytime, anywhere.

If you choose to run Backtrack using the bootable DVD, shortly after the system starts, you will be presented with a menu list. You will need to review the list carefully as it contains several different options. The first couple of options are used to set some basic information about your system’s screen resolution. If you are having trouble getting Backtrack to

boot, be sure to choose the “Start Backtrack in Safe Graphical Mode”. The menu contains several other options, but these are outside the scope of this book. To select the desired boot option, simply use the arrow keys to highlight the appropriate row and hit the enter key to confirm your selection.

Kali Linux works in much the same way. You need to choose between downloading an ISO and burning it to DVD or downloading a preconfigured VMware image. Regardless of which version you selected, you can simply accept the default option (by hitting the Enter key) when presented with the Kali Linux GRUB bootloader boot menu.

The use of Kali or Backtrack is not required to work through this book or to learn the basics of hacking. Any version of Linux will do fine. The major advantage of using Kali or Backtrack is that all the tools are preloaded for you. If you choose to use a different version of Linux, you will need to install the tools before reading the chapter. It is also important to remember that because this book focuses on the basics, it does not matter which version of Kali or Backtrack you are using. All the tools we will explore and use in this book are available in every version.

Working with Your Attack Machine: Starting the Engine

Regardless of whether you choose to run Kali or Backtrack as either a VM or Live DVD, once the initial system is loaded you will be presented with a login prompt. The default user name is root and the default password is toor.

Notice the default password is simply “root” spelled backward. This default user name and password combination has been in use since Backtrack 1, and most likely it will remain in use for future versions. At this point, if you are running Backtrack, you should be logged into the system and should be presented with “root@bt: ~ #” prompt. Although it is possible to run many of the tools, we will discuss in this book directly from the terminal, it is often easier for newcomers to make use of the X Window System. You can start the GUI by typing the following command after the “root@bt: ~ #” prompt:

Startx

After typing this command and hitting the Enter key, X will begin to load. This environment should seem vaguely familiar to most computer users. Once it has completely loaded, you will see a desktop, icons, a taskbar, and a system tray. Just like Microsoft Windows, you can interact with these items by moving your mouse cursor and clicking on the desired object. If you are utilizing Kali Linux, after logging in with the default root/toor user name and password you will be automatically loaded to the GUI-based Gnome desktop environment.

Most of the programs we will use in this book will be run from the terminal. There are several ways to start the terminal. In most Linux distributions, you can use the keyboard shortcut: Ctrl + Alt + T. Many systems also include an icon represented by a black box with a: >_ inside of it. This is often located in the taskbar or menu of the system.

Unlike Microsoft Windows or many of the modern-day Linux OS's, by default, some versions of Backtrack do not come with networking enabled. This setup is by design. As a penetration tester, we often try to maintain a stealthy or undetected presence. Nothing screams “Look at Me!! Look at Me!! I’m Here!!!” like a computer that starts up and instantly begins spewing network traffic by broadcasting requests for a Dynamic Host Configuration Protocol (DHCP) server and Internet protocol (IP) address. To avoid this issue, the networking interfaces of your Backtrack machine may be turned down (off) by default.

The easiest way to enable networking is through the terminal. Open a terminal window by clicking on the terminal icon or (if you are using Backtrack) by using the keyboard shortcut Ctrl + Alt + T. Once the terminal opens, enter the following command:

Ifconfig -a

This command will list all the available interfaces for your machine. At the minimum, most machines will include an eth0 and a lo interface. The “lo” interface is your loopback interface. The “eth0” is your first Ethernet card. Depending on your hardware, you may have additional interfaces or

different interface numbers listed. If you are running Backtrack through a VM, your main interface will usually be eth0.

To turn the network card on, you enter the following command into a terminal window:

Ifconfig eth0 up

Let us examine this command in more detail; “ifconfig” is a Linux command that means “I want to configure a network interface”. As we already know, “eth0” is the first network device on our system (remember computers often start counting at 0 not 1), and the keyword “up” is used to activate the interface. So we can roughly translate the command you entered as “I want to configure the first interface to be turned on”.

Now that the interface is turned on, we need to get an IP address. There are two basic ways to complete this task. Our first option is to assign the address manually by appending the desired IP address to the end of the previous command. For example, if we wanted to assign our network card, an IP address of 192.168.1.23, we would type (assuming your interface is “eth0”):

Ifconfig eth0 up 192.168.1.23

At this point, the machine will have an IP address but will still need a gateway and Domain Name System (DNS) server. A simple Google search for “setting up network interface card (NIC) Linux” will show you how to enter that information. You can always check to see if your commands worked by issuing the following command into a terminal window:

Ifconfig -a

Running this will allow you to see the current settings for your network interfaces. Because this is a beginner’s guide and for the sake of simplicity, we will assume that stealth is not a concern at the moment. In that case, the easiest way to get an address is to use DHCP. To assign an address through DHCP, you simply issue the command:

Dhclient

Please note, dhclient will attempt to automatically assign an IP address to your NIC and configure all required settings including DNS and Gateway

information. If you are running Kali or Backtrack Linux from VMware Player, the VMware software will act as the DHCP server.

Regardless of whether you used DHCP or statically assigned an address to your machine, your machine should now have an IP address. If you are using Kali Linux, your networking should be preconfigured. However, if you have any issues the preceding section will be helpful. The last thing to address is how to turn off Backtrack or Kali. As with most things in Linux, there are multiple ways to accomplish this task. One of the easiest ways is to enter the following command into a terminal window:

Poweroff

Alert! It is always a good idea to power-off or reboot your attacking machine when you are done running a pen test. You can also run the command “shutdown” or “shutdown now” command to power-off your machine. This good habit prevents you from accidentally leaving a tool running or inadvertently sending traffic from your network while you are away from your machine.

You can also substitute the power-off command with the reboot command if you would prefer to restart the system rather than shut it down.

Before proceeding, you should take several minutes to review and practice all the steps discussed thus far including the following:

- Power on/Start up Backtrack or Kali
- Login with the default user name and password
- Start X (the Windows GUI) if you are using Backtrack
- View all the network interfaces on your machine
- Turn up (on) the desired network interface
- Assign an IP address manually
- View the manually assigned IP address
- Assign an IP address through DHCP
- View the dynamically assigned address

- Reboot the machine using the command line interface
- Power-off the machine using the command line interface.

The Use and Creation of a Hacking Lab

Every ethical hacker must have a place to practice and explore. Most newcomers are confused about how they can learn to use hacking tools without breaking the law or attacking unauthorized targets. This is most often accomplished through the creation of a personal “hacking lab”. A hacking lab is a sandboxed environment where your traffic and attacks have no chance of escaping or reaching unauthorized and unintended targets. In this environment, you are free to explore all the various tools and techniques without fear that some traffic or attack will escape your network. At the minimum, the lab is set up to contain at least two machines: one attacker and one victim. In other configurations, several victim machines can be deployed simultaneously to simulate a more realistic network.

The proper use and setup of a hacking lab is vital because one of the most effective means to learn something is by doing that thing. Learning and mastering the basics of penetration testing is no different.

The single, most crucial point of any hacker lab is the isolation of the network. You must configure your lab network in such a way that it is impossible for traffic to escape or travel outside of the network. Mistakes happen and even the most careful people can fat-finger or mistype an IP address. It is a simple mistake to mistype a single digit in an IP address, but that mistake can have drastic consequences for you and your future. It would be a shame (and more importantly illegal) for you to run a series of scans and attacks against what you thought was your hacker lab target with an IP address of 172.16.1.1 only to find out later that you actually entered the IP address as 72.16.1.1.

The simplest and most effective way to create a sandboxed or isolated environment is to physically unplug or disconnect your network from the Internet. If you are using physical machines, it is best to rely on hardwired Ethernet cables and switches to route traffic. Also be sure to double-and triplecheck that all your wireless NICs are turned off. Always carefully inspect and review your network for potential leaks before continuing.

Although the use of physical machines to create a hacking lab is an acceptable solution, the use of VMs provides several key benefits. First, given today's processing power, it is easy to set up and create a mini hacking lab on a single machine or laptop. In most cases, an average machine can run two or three VMs simultaneously because our targets can be set up using minimal resources. Even running on a laptop, it is possible to run two VMs at the same time. The added benefit of using a laptop is the fact that your lab is portable. With the cheap cost of external storage today, it is easily possible to pack hundreds of VMs on a single external hard drive. These can be easily transported and set up in a matter of minutes. Anytime you are interested in practicing your skills or exploring a new tool, simply open up Kali Linux, Backtrack, or your attack machine and deploy a VM as a target. Setting up a lab like this gives you the ability to quickly plug-and-play with various OSs and configurations.

Another benefit of using VMs in your pen testing lab is the fact that it is very simple to sandbox your entire system. Simply turn off the wireless card and unplug the cable from the Internet. As long as you assigned addresses to the network cards like we covered in the previous section, your physical machine and VMs will still be able to communicate with each other and you can be certain that no attack traffic will leave your physical machine.

In general, penetration testing is a destructive process. Many of the tools and exploits we run can cause damage or take systems offline. In some cases, it is easier to reinstall the OS or program rather than attempt to repair it. This is another area where VMs shine. Rather than having to physically reinstall a program like SQL server or even an entire OS, the VM can be quickly reset or restored to its original configuration.

In order to follow along with each of the examples in this book you will need access to the three VMs:

- **Kali or Backtrack Linux:** the screenshots, examples, and paths in this book are taken from Kali Linux but Backtrack 5 (and any previous edition) will work as well. If you are using Backtrack 5, you will need to locate the proper path for the tool being discussed. With Backtrack most tools can be located by navigating the Applications → Backtrack menu on the desktop or by using the terminal and moving into the/pen test directory. Regardless

of whether you choose Backtrack or Kali, this VM will serve as your attacker machine for each exercise.

- **Metasploitable:** Metasploitable is a Linux VM which was created in an intentionally insecure manner. Metasploitable is available for free from SourceForge at <http://sourceforge.net/projects/metasploitable/>. Metasploitable will serve as one of our targets when we cover exploitation.

- **Windows XP:** while most of the exercises in this book will run against Metasploitable, Windows XP (preferably with no service packs installed) will also be used as a target throughout the book. With its wide deployment base and past popularity, most people have little trouble getting a valid copy of Windows XP. A default installation of Windows XP makes an excellent target for learning hacking and penetration testing techniques.

For the duration of this book, each of the systems listed above will be deployed as a VM on a single laptop. Networking will be configured so that all machines belong to the same subnet and are capable of communicating with each other.

Alert! Even if you cannot get your hands on a Windows XP VM, you can still follow along with many of the examples in this book by utilizing Metasploitable. Another option is to simply make a second copy of Backtrack (or Kali). If you use two copies of your attack machine, one can serve as the attacker and one as the target.

Phase's of a Penetration Test

Like most things, the overall process of penetration testing can be broken down into a series of steps or phases. When put together, these steps form a comprehensive methodology for completing a penetration test. Careful review of unclassified incident response reports or breach disclosures supports the idea that most black hat hacker's also follow a process when attacking a target. The use of an organized approach is important because it not only keeps the penetration tester focused and moving forward, but also allows the results or output from each step to be used in the ensuing steps.

The use of a methodology allows you to break down a complex process into a series of smaller, more manageable tasks. Understanding and following a methodology is an important step in mastering the basics of hacking. Depending on the literature or class you are taking, this methodology usually contains between four and seven steps or phases. Although the overall names or number of steps can vary between methodologies, the important thing is that the process provides a complete overview of the penetration testing process. For example, some methodologies use the term “Information Gathering”, whereas others call the same process “Reconnaissance” or “Recon” or even “OSINT”. For the purpose of this book, we will focus on the activities of the phase rather than the name. After you have mastered the basics, you can review the various penetration testing methodologies and choose one that you like best.

To keep things simple, we will use a four-step process to explore and learn penetration testing. If you search around and examine other methodologies (which is important to do), you may find processes that include more or less steps than we are using as well as different names for each of the phases. It is important to understand that although the specific terminology may differ, most solid penetration testing methodologies cover the same topics.

There is one exception to this rule: the final step in many hacking methodologies is a phase called “hiding”, “covering your tracks”, or “removing evidence”. Because this book focuses on understanding the basics, it will not be included in this methodology. Once you have a solid understanding of the basics, you can go on to explore and learn more about this phase.

The remainder of this book will be dedicated to reviewing and teaching the following steps: Reconnaissance, Scanning, Exploitation, and Post Exploitation (or Maintaining Access). Sometimes, it helps to visualize these steps as an inverted triangle. The reason we use an inverted triangle is because the outcome of initial phases is very broad. As we move down into each phase, we continue to drill down to very specific details

The inverted triangle works well because it represents our journey from the broad to the specific. For example, as we work through the reconnaissance phase, it is important to cast our nets as wide as possible. Every detail and every piece of information about our target is collected and stored. The

penetration testing world is full of many great examples when a seemingly trivial piece of information was collected in the initial phase; and later turned out to be a crucial component for successfully completing an exploit and gaining access to the system. In later phases, we begin to drill down and focus on more specific details of the target. Where is the target located? What is the IP address? What OS is the target running? What services and versions of software are running on the system? As you can see, each of these questions becomes increasingly more detailed and granular. It is important to note that asking and answering these questions in a particular order is important.

Additional Information: As your skills progress beyond the basics, you should begin to wean yourself off the use of “vulnerability scanners” in your attack methodology. When you are starting off, it is important to understand the proper use of vulnerability scanners as they can help you connect the dots and understand what vulnerabilities look like. However, as you become experienced, vulnerability scanners may become a crutch to the “hacker mentality” you are trying to hone. Continuous and exclusive reliance on this class of tool may eventually hinder growth and understanding of how vulnerabilities work and how to identify them. Most advanced penetration testers I know rarely use vulnerability scanners unless they have no other options.

However, because this book covers the basics, we will discuss vulnerability scanners and their proper use in the Zero Entry Hacking methodology

It is also important to understand the order of each step. The order in which we conduct the steps is very important because the result or output of one step often needs to be used in the step below it. You need to understand more than just how to simply run the security tools in this book. Understanding the proper sequence in which they are run is vital to performing a comprehensive and realistic penetration test.

For example, many newcomers skip the Reconnaissance phase and go straight to exploiting their target. Not completing steps 1 and 2 will leave you with a significantly smaller target list and attack vector on each target. In other words, you become a one-trick-pony. Although knowing how to use a single tool might be impressive to your friends and family, it is not to the security community and professionals who take their job seriously.

It may also be helpful for newcomers to think of the steps we will cover as a circle. It is very rare to find critical systems exposed directly to the Internet in today's world. In many cases, penetration testers must access and penetrate a series of related targets before they can directly attack the original target. In these cases, each of the steps is often repeated. The process of compromising one machine and then using that machine to compromise another machine is called pivoting. Penetration testers often need to pivot through several computers or networks before reaching their final target.

Let us briefly review each of the four steps that will be covered so you have a solid understanding of them. The first step in any penetration test is "reconnaissance". This phase deals with information gathering about the target. As was mentioned previously, the more information you collect on your target, the more likely you are to succeed in later steps.

Regardless of the information you had to begin with, after completing in-depth reconnaissance you should have a list of target IP addresses that can be scanned. The second step in our methodology can be broken out into two distinct activities. The first activity we conduct is port scanning. Once we have finished with port scanning, we will have a list of open ports and potential service running on each of the targets. The second activity in the scanning phase is vulnerability scanning. Vulnerability scanning is the process of locating and identifying specific weaknesses in the software and services of our targets. With the results from step 2 in hand, we continue to the "exploitation" phase. Once we know exactly what ports are open, what services are running on those ports, and what vulnerabilities are associated with those services, we can begin to attack our target. It is this phase and its tools which provide push-button mass-exploitation that most newcomers associate with "real" hacking. Exploitation can involve lots of different techniques, tools, and code. The ultimate goal of exploitation is to have administrative access (complete control) over the target machine.

Alert! Exploitation can occur locally or remotely. Local exploitation requires the attacker to have physical access to the computer while remote exploitation occurs through networks and systems when the attacker cannot physically touch the target. This book will cover both local and remote attacks. Regardless of whether the attack is local or remote, full administrative access usually remains the definitive goal. Administrative

access allows a hacker to fully and completely control the target machine. New programs can be installed, defensive tools can be disabled, confidential documents can be copied, edited, or deleted, security settings can be changed and much more.

The final phase we will examine is “post exploitation and maintaining access”. Oftentimes, the payloads delivered in the exploitation phase provide us with only temporary access to the system. Because most payloads are not persistent, we need to quickly move into post exploitation in order to create a more permanent backdoor to the system. This process allows our administrative access to survive program closures and even reboots. As an ethical hacker, we must be very careful about the use and implementation of this phase. We will discuss how to complete this step as well as the ethical implications of using backdoor or remote-control software.

Although not included as a formal step in the penetration testing methodology, the final (and arguably the most important) activity of every PT is the report. Regardless of the amount of time and planning you put into conducting the penetration test, the client will often judge your work and effectiveness on the basis of the quality of your report. The final PT report should include all the relevant information uncovered in your test and explain in detail how the test was conducted and what was done during the test. Whenever possible, mitigations and solutions should be presented for the security issues you uncovered. Finally, an executive summary should be included in every PT report. The purpose of this summary is to provide a simple one-to two-page, nontechnical overview of your findings. This report should highlight and briefly summarize the most critical issues your test uncovered. It is vital that this report be readable (and comprehensible) by both technical and nontechnical personnel. It is important not to fill the executive summary with too many technical details; that is the purpose of the detailed report.

Additional Information: The Penetration Testing Execution Standard (PTES) is a fantastic resource if you are looking to find a more in-depth and thorough methodology. The PTES includes both technical guidelines which can be used by security professionals as well as a framework and common language that can be leveraged by the business community

Where Do I Go from Here?

It should be noted that there are several alternatives to Kali or Backtrack. All the examples in this book should work with each of the security auditing distributions discussed below. Blackbuntu is an Ubuntu-based security distro with a very friendly community, great support, and active development. Black box is another great penetration testing distribution based on Ubuntu and includes a sleek, lightweight interface and many preinstalled security tools. Matriux is similar to Backtrack but also includes a Windows binary directory that can be used and accessed directly from a Windows machine. Fedora Security Spin is a collection of security-related tools built off of the Fedora distribution. Katana is a multiboot DVD that gathers a number of different tools and distributions into a single location. Finally, you may want to explore the classic STD distribution as well as Pentoo, NodeZero, and SamuraiWTF. There are many other Linux penetration testing distributions—a simple Google search for “Linux Penetration Testing Distributions” will provide you with a plethora of options. You could also spend some time building and customizing your own Linux distribution by collecting and installing tools as your hacking career progresses.

FINAL THOUGHT

This chapter presented the idea of entrance testing and hacking as a means of securing frameworks. A special "basics just", four-venture procedure including Reconnaissance, Scanning, Exploitation, and Post Exploitation and Maintaining Access was presented. This part also discussed the various roles and characters involved in the hacking scene. The basics of Backtrack Linux, including how to boot up, login, start X, access the terminal, obtain an IP address, and closure the framework, were covered. Kali Linux, a re-envisioned version of Backtrack was also introduced. The creation and utilization of a penetration testing lab were outlined. The specific requirements, allowing you to practice your skills in a safe and sandboxed climate and follow along with the examples in the book, were presented. This section wrapped up by providing additional details on alternatives to Kali or Backtrack Linux which could be investigated by the peruser.