

HACKING

101 HACKING GUIDE



ALEX BENJAMIN

Hacking

A 101 Hacking Guide

By Alex Benjamin

Copyright 2015 by TSM Publishing - All rights reserved.

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

In no way is it legal to reproduce, duplicate, or transmit any part of this document in either electronic means or in printed format. Recording of this publication is strictly prohibited and any storage of this document is not allowed unless with written permission from the publisher. All rights reserved.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

Respective authors own all copyrights not held by the publisher.

The information herein is offered for informational purposes solely, and is universal as so. The presentation of the information is without contract or any type of guarantee assurance.

The trademarks that are used are without any consent, and the publication of the trademark is without permission or backing by the trademark owner. All trademarks and brands within this book are for clarifying purposes only and are the owned by the owners themselves, not affiliated with this document.

Contents

Introduction

[Chapter 1: What is Ethical Hacking?](#)

[Chapter 2: Basic Terminology](#)

[Chapter 3: Types of Attacks](#)

[Chapter 4: Types of Tools](#)

[Chapter 5: Hacking Passwords](#)

[Chapter 6: Accessing Ports](#)

[Chapter 7: Penetration Testing](#)

[Chapter 8: Unix](#)

[Chapter 9: Where do I Go from Here?](#)

Conclusion

Introduction

I want to thank you for downloading the book, **Hacking: A 101 Hacking Guide**. This book is for absolute beginners who want to learn about ethical hacking by starting with a solid foundation. Written in a down to earth style, this book contains the key terms and concepts you need coupled with links to online resources that let you build your skills outside the book.

Here is what you will be able to do the end of this book:

- Be able explain the difference between an ethical hacker and a non-ethical hacker, including goals and motivations
- Discuss why ethical and non-ethical hackers use the same tools
- Know the difference between an attack, a threat, and a vulnerability
- Have a solid understanding of the basic terminology you need to study hacking
- Understand the different methods used to crack passwords
- Be familiar with the different types of attacks
- Learn the types of tools used by hackers
- Understand how port scanning works
- Know the steps involved in penetration testing
- Learn why Unix is popular with hackers
- Get some tips on how to keep building your skills

Thank you again for downloading this book. I hope you enjoy it!

Chapter 1: What is Ethical Hacking?

An **ethical hacker** is one that builds, fortifies, secures, and strengthens. To do that, the ethical hacker must get into the mindset of whoever is trying to break into their system. They will thoroughly check their system for weaknesses, and figure out how they can be exploited. Then, they seek to eliminate those weaknesses.

This book is aimed at the **ethical hacker**, not a destructive hacker (also known in some circles as crackers). The purpose of this book is to provide you with a basic understanding of how to start testing your system to make it as safe and impenetrable as possible.

A **white hat hacker** is another word for an ethical hacker, and goes back to the image of the old western movies where the good guy would wear a white hat, and the bad guy would wear a black hat. You can guess what a **black hat hacker** is!

Black hat hackers have many different motivations: some enjoy causing chaos and disruption, others might attack out of revenge or out of sheer malice, still others merely do what they do to show the world that they can, and some may be hired by outside entities and see themselves as merely providing a service, and still others are trying to make a point. They see vulnerabilities as potential points of attack, like unsecured windows on a home, unlocked doors, or faulty alarm systems – that they can use to their own advantage.

White hat hackers are motivated by a concern for security, whether it is for their own system, their company's system, or that of a client. When they see vulnerabilities, they investigate them just as thoroughly – and, better yet, even more thoroughly – as the black hat hackers. However, the goal is not to discover how to use them to their own advantage, but how to secure them.

White hat and black hat hackers will probably use the same tools – just like a locksmith and a professional thief may have the same tools in their bags. It's not the tool that is evil, but how it is being used. A white hat hacker might use a password hacking tool to test how strong a company's authentication is, whereas a black hat hacker may use the exact same tool to gain entrance to a server to steal data.

Data shows that the job market for white hat hackers is good. Companies are quickly learning that it is better to invest in the skills of an ethical hacker before

anything happens than deal with the financial damage, loss of trust, and loss of reputation. According to [Statista.com](http://www.statista.com), the average cost of cybercrime in the US for 2014 was 12.69 million **per company**.

Remember: white hat hackers never intrude where they don't have permission, and never use what they learn about a system for anything but strengthening its defenses.

Online Resources:

How to Get a Job as an Ethical Hacker:

<http://intelligent-defense.softwareadvice.com/how-to-get-an-ethical-hacker-job-0714/>

Occupational Outlook Handbook for Information Security Specialists:

<http://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm>

The Role of White Hat Hackers:

<http://phys.org/news/2015-01-role-white-hat-hackers-cyber.html>

Cost of Cybercrime in the US:

<http://www.statista.com/statistics/193444/financial-damage-caused-by-cyber-attacks-in-the-us/>

Cost of Cybercrime in Selected Countries:

<http://www.statista.com/statistics/293274/average-cyber-crime-costs-to-companies-in-selected-countries/>

Chapter 2: Basic Terminology

When you begin a new subject, the first step is to become familiar with the terminology.

If your system has suffered an **attack**, it means that the security of your system has been violated. A **threat** is something that can affect your system, but hasn't happened yet. A **vulnerability** is an error or weakness that has the potential to compromise your system. It is very important to understand the difference between an actual attack and a vulnerability or threat.

Bugs! No, not the creepy, crawly bugs you can kill with a quick stomp. In hacking, **bugs** refer to errors in a program. The term "bug" came from the old days when computers had physical relays, and a particular mathematical subroutine was giving bad results. The software engineer (legend points to Admiral Grace Murray Hopper) started tracking down the error and found a moth caught in the relay, insulating it so that current couldn't pass through.

In the movies, bad guys often break in through the backdoor. In hacking, **backdoor** refers to sneakily accessing someone's system by bypassing the authentication (think of your locked front door) that is supposed to protect it.

You know your computer has cookies, right? **Cookies** are the funny name that someone came up with for text that your browser stores for websites. Let's say you recently purchased some running shoes online. If you go to a new website you have never been too, and you notice it starts advertising running shoes, that may be a good indication that there is a cookie on your computer that recorded what you purchased or searched for, and other websites are accessing it. Cookies can also be what let's you into a website without having to enter your username and password all the time.

Did you know your computer has daemons? It's not possessed, though. **Daemons** in computer-speak refer to services that run on ports. You need these daemons in order for your computer to function properly; so don't call for the Winchesters quite yet.

We know what garbage dumps are, but what about hacking dumps? A **dump** in hacking refers to a collection of information that has been stolen.

If we exploit someone, we take advantage of them, usually through a weakness or vulnerability. In hacking speak, **exploitation** is attacking a system through a

weakness of vulnerability. The word **exploit** is the program used to do it.

When you think of a wall of fire, you should picture something that is almost impossible to get through (well, at least in your street clothes without a vehicle).

A computer **firewall** is a program used to keep unauthorized access to your system. It ' s usually your first line of defense against unauthorized intrusions.

Hactivism is using hacking as a form of activism, and those who participate are called **hactivists**. Their activities can vary widely, from hacking a website and placing their own message on it to accessing an organization ' s emails and releasing them to the public. There is a link at the end of the chapter to an interesting article on the pros and cons of hactivism.

An **IP Address** is a unique identifier for your computer or server as it exists on a network or the web. Knowing the IP address of a computer is a starting point for an attack.

Remote access means to access a computer or server without physically connecting to it, like accessing your office computer from home. For hackers, remote access means controlling the computer or server they have attacked – again, without physically connecting to it.

This should be enough terminology for you to follow the rest of the book, and navigate online resources for beginning hackers. Always remember that if you see a word you don ' t recognize, look it up (or Google it).

Online Resources:

Hactivism – Good or Evil:

<http://www.computerweekly.com/opinion/Hactivism-Good-or-Evil>

Internet Relay Chat: <http://www.irc.org/>

All About Bugs: <https://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/bugs.html>

Chapter 3: Types of Attacks

The next step in preparing yourself to become a white hat hacker is to understand the different type of attacks that are launched against systems. Some of these words you may already be familiar with, but it is important to understand the difference between different types of attacks.

Malware is exactly what it sounds like: *malicious software*. **Malware** is specifically designed to exploit backdoors. It 's sneaky, too: you can download malware at the same time you 're downloading some useful software from the web. That 's why many companies won 't allow their employees to download or install their own software.

A **virus** is malware, and like the common cold, loves to share itself with anything it can. A **worm** is a type of virus that spreads itself, worming its way into other systems by, for example, emailing itself. They often cause the denial of service attacks we talked about earlier as they broadcast themselves to other computers.

Remember the legend of the Trojan horse? The guys brought in this giant, awesome looking wooden horse their enemies left outside their gate. Once they brought it in, when they least expected it, their enemies came pouring out. A **trojan** in hacking speak is a piece of malware that lurks on a computer and will open a backdoor so that a hacker can access it.

How many times in the movies have you seen someone break down a door they can 't unlock? If they had a key, it wouldn 't be necessary. If they were skilled at lock picking, it wouldn 't be necessary, either. We call the act of breaking down a door as brute force. In hacking, **brute force** refers to something similar: it means using a program to generate every possible combination of characters, numbers, and symbols to figure out a password.

A **Denial of Service** (DoS) attack makes a website or server unresponsive. The black hat hacker sends so many requests to the website or server that it gets bogged down and essentially crashes.

Doxing is another disturbing hacking act: putting information about a compromised victim on the web, like passwords, email accounts, etc. It seems to be the hacking equivalent of writing your ex-girlfriend 's name and phone number on the bathroom wall with a message like, " For a good time, call "

A **drive by download** works like this: you land on a webpage, and without clicking a single thing or installing any software, malware is downloaded and installed on your computer. It can also happen via email or messaging, and can attack mobile devices as well. It usually takes advantages of browsers or apps that have a vulnerability that either hasn't been addressed, or the user hasn't downloaded the latest updates for. Websites that host drive by downloads include adult websites and file sharing websites.

Phishing is kind of like fishing. Let's say you are going fishing. On the end of your fishing pole, you attach one of those rubber worms. When you dangle that fake worm in the water, you are counting on at least one fish down there to not be smart enough to tell the difference. You dangle the fake worm and wait ... and then some fish will fall for the bait, and you catch them. In phishing, the hacker dangles something like a fake login form or a fake website and waits for someone who doesn't recognize that it isn't real. When they access it, just like a fish taking the bait, they have just given their information over to a hacker.

Port scanning involves determining which ports on a system are open and what services are running on them. Open ports are vulnerable to attack.

Spam means Spiced Ham in the supermarket, but in the cyber universe it means to harass someone (or something) by sending an onslaught of unwanted messages or requests. A **spammer** is someone who practices this annoying art.

To hackers, **spoofing** refers to pretending to be someone or something else in order to obtain information. One example is email spoofing, for example, where an email is sent out pretending to be from a credit card company and requesting that you follow the link and enter your credit card number to access vital information about your account. The goal is to obtain information from targets. Another type of spoofing is **IP spoofing**, where a computer appears to other to have one IP address, when it actually has a different one.

Spyware is a particular devious piece of software whose entire goal is to send someone a continuous flow of information about their target, without the target being aware. People usually think spyware is limited to computers, but spyware can be on your cell phone, too.

Another type of attack involves taking advantage of a bug in a program. As a simplistic example, let's say program A has this one bug that if a certain variable named STARGATE ever exceeds 400 it will erase everything in your My Documents folder. However, when the developers checked out the bug, they

determined that there is *no way* that STARGATE will ever exceed 400, but they are working on a **patch** to fix problem. A black hat hacker learns of this bug before the patch comes out, and figures out how to convince the program that STARGATE has a value of 501. You can image the rest! That ' s why software is continually checking for updates, fixes, patches, etc.

Attacks are often classified as **active** or **passive**. A good example of an active attack is denial of service: you can tell when you are being attacked because your computer or server grinds to a halt. For passive attacks, packet sniffing and key loggers are excellent examples: something that could be intercepting your data without you even knowing it. Spyware and port scanning are usually passive attacks, also.

Firewalls and virus protection software are a first line of defense against many attacks, but require regular updating to keep up with new threats that appear. Keep in mind that skilled hackers know how the protection works! Many of the computer security software companies provide up-to-date information about current threats, which is something any hacker should be knowledgeable about. For example, McAfee provides statistics, a world map, and region specific virus information.

In the United States as of June 2014, [Statista.com](http://www.statista.com) reported that the majority of cyber attacks against US companies took the form of viruses, trojans, and worms, followed by malware and botnets.

Online Resources:

Cyberattacks against US Companies:

<http://www.statista.com/statistics/293256/cyber-crime-attacks-experienced-by-us-companies/>

US Adult Victims to Online Attacks:

<http://www.statista.com/statistics/294684/online-adult-cyber-crime-victimization/>

McAfee Virus Information: <http://home.mcafee.com/virusinfo?ctst=1>

Norton Internet Security Information: http://us.norton.com/security_response/

Chapter 4: Types of Tools

In this chapter, we are going to look at some of the tools used by hackers.

Anonymous browsing is used by regular computers and hackers alike. It allows you to surf the web without your browser recording your history. You would be surprised at how much information travels with you on the web. We already talked about cookies, but did you know your IP address could sometimes reveal your actual physical location? That's why hackers use tools to hide their IP addresses, such as JonDo or Tortilla.

A **bot** is derived from the term robot, and refers to a program that hackers use to perform boring, awful, repetitive tasks. A **botnet** refers to a group of systems that have been compromised and are now being used by a hacker to launch other attacks.

IRC stands for **Internet Relay Chat** and is a computer communication protocol that hackers often use to share files and have conversations.

Keylogging is the computer equivalent to tapping phone calls. A **keylogger** records all your keystrokes – and what hackers are usually interested in are the keystrokes that involve typing in your usernames and passwords to the system or different websites. Some companies install keyloggers on employee computers, which is why you should NEVER bad mouth your boss on the computer at work, even if you are typing it in to a personal chat or email account.

Have you ever tried to do something on your computer and it told you that you didn't have the right privileges to do that? On a Windows computer, you probably need administrative access and on a Unix computer you need **root** access. A **root kit** is what hackers use to obtain those high level privileges on systems so they can setup their malware.

The shell of a snail is what they live inside of; the **shell** for a computer is an outer layer program that provides users an interface to interact with it. It's usually a **command line interface** (CLI), where the user types in instructions at a prompt, or a **graphical user interface** (GUI) where the user interacts with icons and controls. In a nutshell (pun intended), it takes the commands you give it and translates them into something the operating system understands. A **shellcode** is a program that gives a hacker access to the shell for the system so they can start running instructions and commands. There are tutorials available online to show

you how to write your own shell code – a link to one is provided at the end of the chapter.

We usually think of black hat hackers as being somewhat introverted, spending their time in a dark room in front of a computer monitor and plying their trade exclusively through typing and clicking. However, there is a method hackers use called **social engineering** where they initiate a conversation with their intended victim in order to learn helpful information. This obviously requires social skills!

A **packet** refers to data that is traveling between systems, much like a packet of mail travels from the source to its destination. A packet could be data from your cell phone to a website, from your computer to the server, etc. A **packet sniffer** is software designed to analyze this data. While a useful tool for network administrators, law enforcement, and the like, it's a powerful force for evil when used by black hat hackers. One example of a packet sniffer is NetworkMiner, and to get a feel for how much information a packet sniffer can get, I recommend you visit their website listed at the end of the chapter. Another is called, aptly enough, Snort.

A **payload** is the program that a hacker runs after successfully gaining access to a system. Keep in mind that most hackers have a purpose for breaking into a system: it may be download files, add themselves as a new user, etc. The payload is what accomplished that purpose.

There are other tools, of course, but this list gives you a basic overview of the tools most often used by hackers. In the online resources below, you will find links to the specific tools discussed in this chapter.

Online Resources:

NetworkMiner:

<http://www.netresec.com/?page=NetworkMiner>

Snort:

<http://www.snort.org>

Shellcoding Tutorial:

<http://www.vividmachines.com/shellcode/shellcode.html>

Social Engineering:

<http://www.social-engineer.org/>

Tortilla:

<http://www.crowdstrike.com/community-tools/>

JonDo:

<https://anonymous-proxy-servers.net/en/jondo.html>

Chapter 5: Hacking Passwords

A common joke that periodically surfaces on the web concerns a set of password requirements and runs something like this: please enter your new password, and remember that it must include both lower case and upper case letters, a number, a symbol, and a single strand of hair from a unicorn. While most passwords don't have requirements quite this bad, companies have a good reason to require strong passwords.

Social Media

One way that hackers obtain passwords is by using a company's social media information to contact employees, by phone or email, with some excuse for which they need the password. Sometimes they will even impersonate a particular individual that works for the IT department. Uninformed employees will often provide that password information, throwing the door wide open for a hacker.

The best way to prevent this from happening is to train employees to contact IT anytime such an information request is received, and never give their password out. Another measure is to remove IT staff information from public forums, such as company websites. If that information is out there, hackers can easily impersonate an IT representative to convince employees to provide them with their password.

Shoulder Surfing

Shoulder surfing is just what it sounds like: looking over someone's shoulder to see what password they are typing in. Sometimes they will watch the eye movements of the person typing in their password to see if they are looking for a reminder, such as family photo, poster, or object.

This can be prevented by asking someone to step back when you are typing in your password, leaning slightly to the side to block their line of sight, or installing a privacy filter on the monitor. Employees also need to be firmly reminded to not base their passwords on visible items in their work area.

Keystroke Logging

Remote keystroke logging is a devious method of getting passwords. Basically it records all the keystrokes that are entered, storing them in a log file that can be

accessed later. Note that some antivirus programs will recognize that a keylogger is running, but not all. It is usually recommended that you inspect each computer individually. Also be aware that keyloggers may be installed as malware, which is why many companies no longer allow employees to download and install their own software.

Physical keyloggers are inserted between the keyboard and the computer, making them easy to spot. The most dangerous keyloggers out there are the software keyloggers.

There are quite a few software-based keyloggers out there, but most free keyloggers lack a vital feature: stealth mode, so that users don't know it's running. You might want to check out the free version of REFOG, which is a software that captures keystrokes, clip contents, visited websites, and what programs were run.

Guessing

Another method of figuring out someone's password is simply guessing, based on what they can tell about the person, including items on their desk or in their line of sight, birthdays of family members, names of pets, etc. That is why we are often burdened with what seems like outrageous password requirements: to prevent others from simply guessing our password.

Weak Authentication Requirements

Many older operating systems could bypass the login requirements by pressing Escape, and some newer systems will allow you to login to the physical computer but not the network by pressing a certain key. Phones and tablets without a password are also wide open to such simple attacks. These are known as weak authentication requirements. Passwords that are too simple, or contain words from the dictionary or maybe your username, are also examples of weak authentication.

Password Cracking Software

There are many software tools out there for assistance in cracking passwords, such as Ophcrack or John the Ripper. There are also websites that list default passwords that come with well-known software, and dictionaries of words that can be used in cracking a password. That's why some password requirements insist that you don't use words that can be found in the dictionary!

Online Resources:

Ophcrack: <http://ophcrack.sourceforge.net/>

Ophcrack Walkthrough:

<http://pcsupport.about.com/od/toolsofthetrade/ss/ophcracksbs.htm>

Default Passwords: <https://cirt.net/passwords>

Refog Keylogger: <http://www.refog.com/>

John the Ripper: <http://www.openwall.com/john/>

Chapter 6: Accessing Ports

Ports allow multiple services (remember the term daemon?) to share a single physical connection for communication. The best example would be allowing access to the internet. Ports are associated with IP addresses and have a port number to identify them.

Let ' s look at an example of how these ports work by looking at email. An email server that is sending and receiving email needs two services: one for sending and receiving messages from other servers, and one for allows users to retrieve their own personal email from the server. The first service is called SMTP, which stands for Simple Mail Transfer Protocol. It usually uses Port 25 to watch for requests to either send mail or receive mail. The second service is usually either the Post Office Protocol (POP) or Internet Message Access Protocol (IMAP). Whatever software you use for sending and receiving email uses one of these services to retrieve your email from the server. The POP service commonly uses Port 110.

The only way to attack a service, such as POP, is through the port they are using. You can think of it as a piping system, where the port acts as a valve. If the port is not being used, it is closed and nothing can get through; if the port is open, then it may be vulnerable to attack. If you aren ' t using a port, it should be closed.

We are going to talk about how hackers use ports to gain access to your system, but first let ' s go over some acronyms and definitions.

DNS: Domain Name Server, translates names into IP addresses

FTP: File Transfer Protocol, used to transfer files from one host to another

HTTP: HyperText Transfer Protocol

HTTPS: HTTP over SSL (see definition below)

POP3: Post Office Protocol version 3, used to retrieve email from a mail server

RPC: Remote Procedure Call, allows a program on one computer to run a program on the server

SSH: Secure Shell, used to login to another computer over the network,

move files between computers, and execute commands remotely

SSL: Secure Sockets Layer, uses two keys to encrypt data shared via the internet

SMTP: Simple Mail Transfer Protocol, used to send email messages from one server to another, or from a mail client to a mail server

TCP: Transmission Control Protocol, allows two hosts to make a connection and exchange data

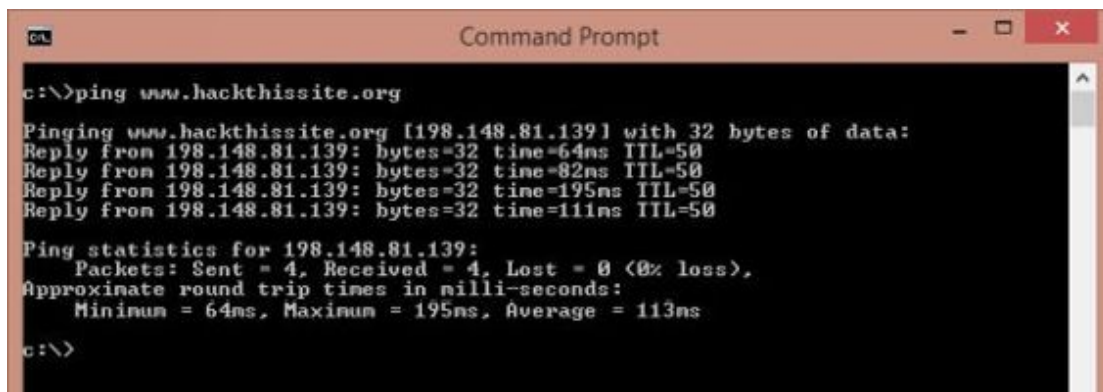
UDP: User Datagram Protocol, primarily used for broadcasting messages over a network

Ports that are commonly hacked include ...

- TCP port 21 - FTP
- TCP port 22 – SSH
- TCP port 23 - telnet
- TCP port 25 - SMTP
- TCP and UDP port 53 - DNS
- TCP port 443 - HTTP and HTTPS
- TCP port 110 - POP3
- TCP and UDP port 135 - Windows RPC
- TCP and UDP ports 137 – 139 - Windows NetBIOS over TCP/IP
- TCP port 1433 and UDP port 1434 – Microsoft SQL Server

Now, how do hackers know if a port is open? The method is called **port scanning**, and it is disturbingly easy, and we are going to look at one of many methods. This example is run on a Windows system, and is so easy beginners can do it. Here is the methodology: obtain the IP address of your target, wait until your target is active, scan the target for open ports, access the system through a vulnerable open port, and hack the username and password.

To get the IP address, use the command ping. For example, in a Windows environment you can open the command prompt and type in the command ping followed by the URL of the site.



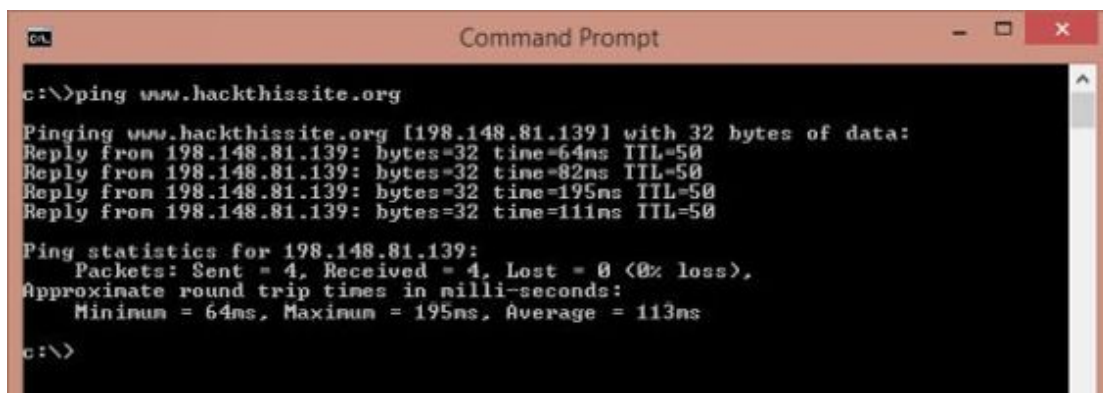
```
c:\>ping www.hackthissite.org

Pinging www.hackthissite.org [198.148.81.139] with 32 bytes of data:
Reply from 198.148.81.139: bytes=32 time=64ms TTL=50
Reply from 198.148.81.139: bytes=32 time=82ms TTL=50
Reply from 198.148.81.139: bytes=32 time=195ms TTL=50
Reply from 198.148.81.139: bytes=32 time=111ms TTL=50

Ping statistics for 198.148.81.139:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 64ms, Maximum = 195ms, Average = 113ms

c:\>
```

This was entered at the command prompt: ping www.hackthissite.org



```
c:\>ping www.hackthissite.org

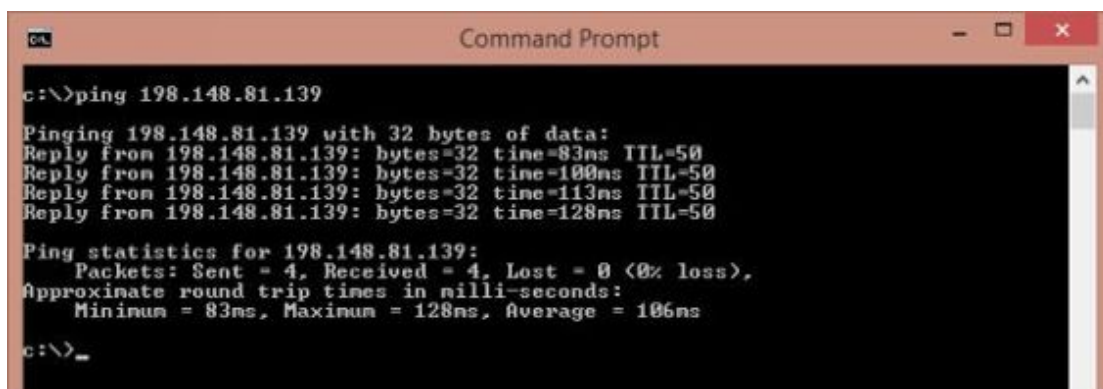
Pinging www.hackthissite.org [198.148.81.139] with 32 bytes of data:
Reply from 198.148.81.139: bytes=32 time=64ms TTL=50
Reply from 198.148.81.139: bytes=32 time=82ms TTL=50
Reply from 198.148.81.139: bytes=32 time=195ms TTL=50
Reply from 198.148.81.139: bytes=32 time=111ms TTL=50

Ping statistics for 198.148.81.139:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 64ms, Maximum = 195ms, Average = 113ms

c:\>
```

Based on what we see, the website www.hackthissite.org has the IP address 198.148.81.139.

To determine if they are online, ping the IP address. If the IP address responds, then it is online. Here is the command: ping 198.48.81.139



```
c:\>ping 198.148.81.139

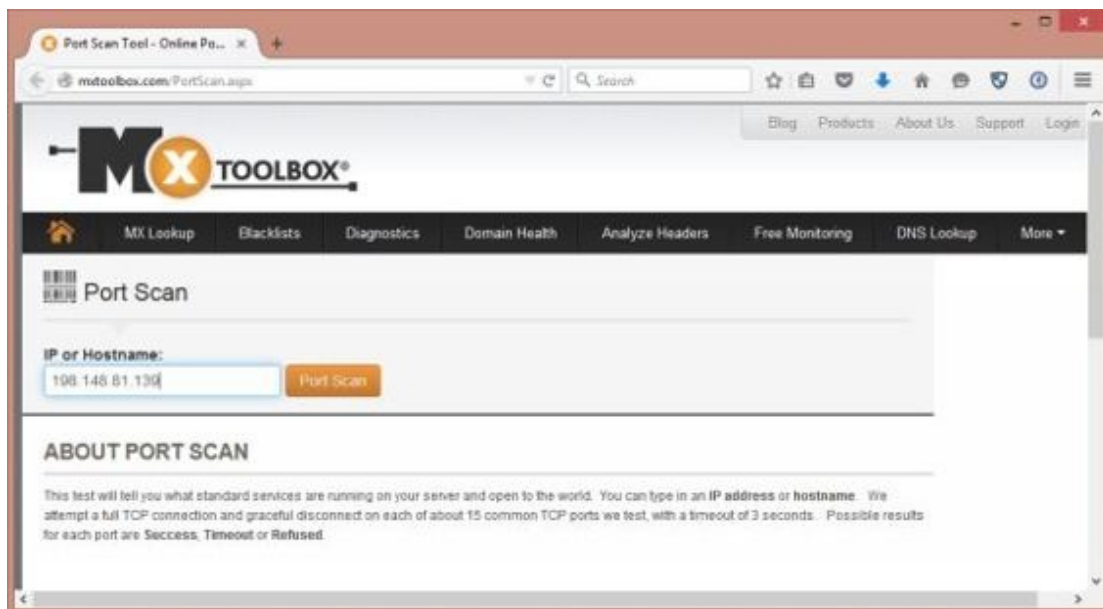
Pinging 198.148.81.139 with 32 bytes of data:
Reply from 198.148.81.139: bytes=32 time=83ms TTL=50
Reply from 198.148.81.139: bytes=32 time=100ms TTL=50
Reply from 198.148.81.139: bytes=32 time=113ms TTL=50
Reply from 198.148.81.139: bytes=32 time=128ms TTL=50

Ping statistics for 198.148.81.139:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 83ms, Maximum = 128ms, Average = 106ms

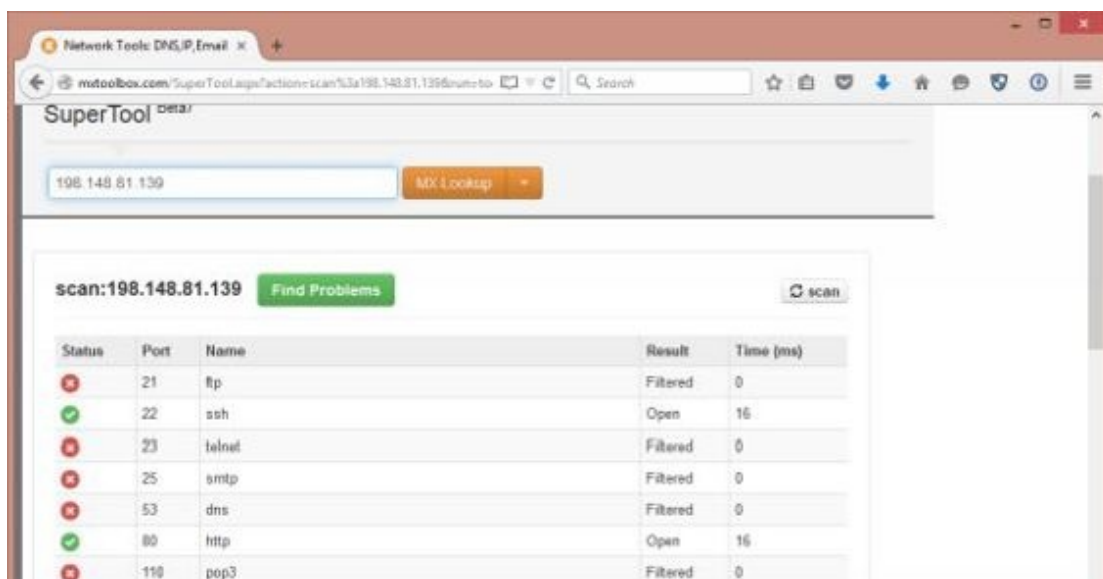
c:\>_
```

The next task is to scan the ports. If you are working with Unix, you can write a script to accomplish this. If you are not adept at programming and/or using a Windows system, there is software available that will do the port scans. For demonstration purposes, this example will use a free online port scanner at <http://mxtoolbox.com/PortScan.aspx>

You start by typing in the IP address, then click Port Scan.



Here is a sample of the results:



You will notice that ports 21 and 80 are open. These are the SSH and HTTP ports.

The next job is to access the open ports. In a Windows environment, you will need to use the command telnet. You may have to install it as a Windows component from the Control Panel. In newer versions of Windows you will need to go to Programs and Features ... Turn Windows Features On or Off and then check the boxes next to Telnet Server and Telnet Client.

Type in this command at the command prompt: telnet 198.148.81.139 22

You will notice that you type in the IP address followed by the number of the port you are wanting to access. Normally you will be asked to provide a username and password, which is another challenge.

You have just learned the basics of running a port scan ... but what do we use that information for? On to the next chapter!

Online Resources:

Online Port Scanning Tool: <http://mxtoolbox.com/PortScan.aspx>

What ' s My IP Address: <http://whatismyipaddress.com/>

Chapter 7: Penetration Testing

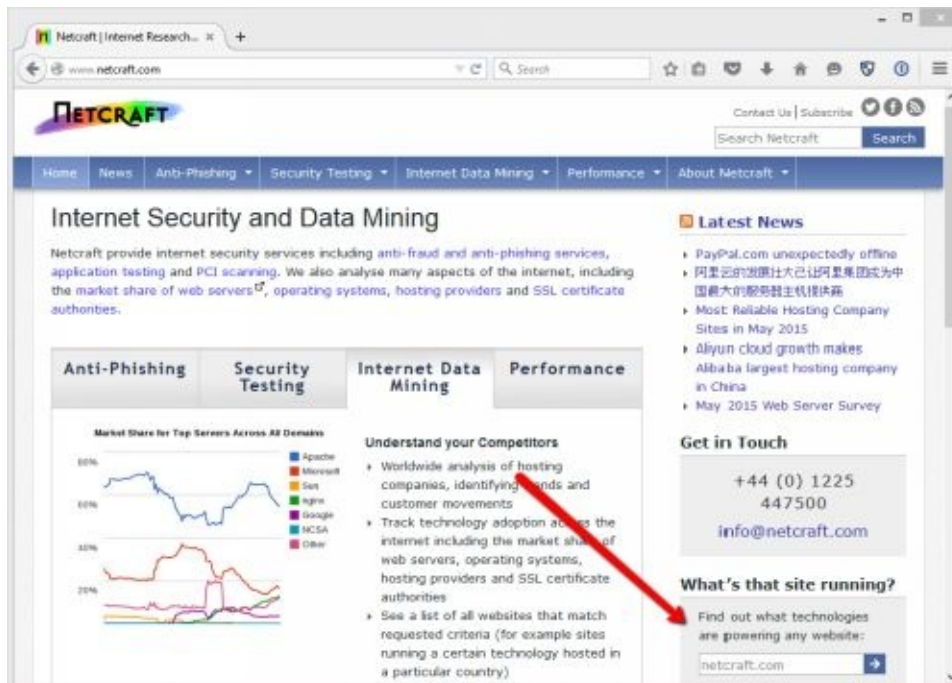
Penetration testing, also known as pen testing or PT, is legally hacking into a system to determine its vulnerabilities, and is part of the white hat hacker world. However, it usually goes beyond just determining the vulnerabilities to demonstrating how they can be exploited. Sometimes this last step is necessary to convince users that the danger is real and must be addressed in a timely manner.

Remember earlier when we discussed the difference between vulnerability and a threat? Pen testing looks for vulnerabilities in the system – unintentional loopholes that leave the system open to attack. Think of it as being similar to hiding a key to your house under a rock in your flowerbed. Everything is fine ... until someone finds it that shouldn't have access to your house. Vulnerabilities work the same way: everything is fine, until a less than ethical hacker decides to use those vulnerabilities to stage an attack.

There are many different ways to approach pen testing. In this book, we are going to look at **Zero-Entry Pen Testing**, which consists of four phases: reconnaissance, scanning, exploitation, and maintaining access.

The **reconnaissance** stage involves gathering information about your target, and the most important result of this step is a list of IP addresses – but that is not all. Many people don't realize that this is the most important step in pen testing: finding out everything you can about your target. Care is required in this stage to make sure that the target isn't alerted to the fact you are prowling around. A skilled black hat hacker doesn't reveal their presence, and neither should a white hat hacker.

To study your target's website without gathering attention, you might want to make a copy of their entire webpage – that way you are only accessing it one time, which shouldn't arouse suspicion. HTTrack is a commonly used tool for this. Basically, it makes a copy of the entire webpage, allowing you to carefully mine the HTML code for valuable information and clues. Another well-known tool for gathering additional information is called Harvester: it searches the web for employee names, email addresses, subdomains, etc. Yet another tool you can put in your toolbox is the website WhoIs. NetCraft has an interesting tool available on their webpage. You can see it in the figure below.



Just as an example, I am going to type in snopes.com (a well-known website for debunking hoaxes and investigating internet rumors).

As you can see, www.snopes.com is running Linux.

Search Web by Domain

Explore 1,656,301 web sites visited by users of the Netcraft Toolbar

19th June 2015

Search:

example: site contains .netcraft.com

Results for snopes.com

Found 5 sites

Site	Site Report	First seen	Netblock	OS
1. www.snopes.com		february 1997	amazon.com, inc.	linux
2. snopes.com		april 2000	amazon.com, inc.	unknown
3. now.snopes.com		april 2015	activehost corporation	linux
4. message.snopes.com		march 2007	hosting services, inc.	linux
5. msgboard.snopes.com		april 2005	thepanet.com internet service, inc.	freebsd

COPYRIGHT © NETCRAFT LTD 2015. ALL RIGHTS RESERVED.

Once you have finished the reconnaissance stage and have the IP addresses, they

feed directly into the **scanning** stage, which starts with scanning the ports at the given IP addresses. Once open ports and the services running on them have been identified, then vulnerability scanning takes place. This was discussed in the previous chapter on ports, however we will take some time to discuss a few of the tools available. The first is the Angry IP Scanner, which works on just about any platform you need and can export the IP scan data to a variety of file formats. NMap is another very powerful scanner, which comes with most modern Linux system, but is available for Windows also.

Once you know what ports are open and what their vulnerabilities are, you begin the **exploitation** stage. The end goal of this step is to obtain administrative access over your target. This can happen remotely (from a different physical location) or locally.

Now, what exactly does **administrative access** mean to a hacker? It means the or she can take down the remaining defenses, install and run their own code, corrupt or delete files, make copies of files, and more.

After the hacker has administrative access, his or her payload (the program that gives the access to the command line) is deployed. Image you are a thief, and you found an unsecured window in your targets home. The payload is the tool you use to get the window open just enough to get in, but to carry anything out you need a bigger opening. This leads to the phase of **maintaining access**, where the hacker would modify security settings, set themselves up as a user, etc. to keep that access open long enough to accomplish their task.

As a white hat ethical hacker, the only time you do penetration testing is to reveal the weaknesses in the system so it can be strengthened, not to take advantage of it. As part of strengthening the system, the ethical hacker will create a detailed report of how they gained access, a discussion of weaknesses discovered, and recommendations/solutions for eliminating those weaknesses.

Online Resources:

HTTrack: <https://www.httrack.com/>

Harvester: <http://www.edge-security.com/theharvester.php>

WhoIs: <https://whois.net/>

Netcraft: <http://www.netcraft.com>

Angry IP Scanner: <http://angryip.org/>

Nmap: <https://nmap.org/>

Nmap examples: <http://www.tecmint.com/nmap-command-examples/>

Chapter 8: Unix

Now that you have a good grasp of the concepts and methods behind white hat hacking, you can start building your hacking skills. If you aren't already familiar with Unix, it's time to learn it. Unlike Windows, Unix is an open source operating system, which means that you can actually look at and modify all the code that was written to create it.

Imagine you are a mechanic, or at least a budding mechanic, and you purchase a car. You start hearing some strange noises from the engine, so you decide to pop the hood to take a look. However, when you try to pop the hood you find it welded shut! You can't even change the oil without going through the car manufacturer to have it done. Fortunately, cars aren't like that – but some operating systems are!

Unix is more like a car: you can pop the hood, look around at the source code, find out how it works, make changes to see what happens, and more.

Unix comes in so many different flavors: Unix, Linux, Kali, Fedora, FreeBSD, Ubuntu ... the list keeps growing. Just like ice cream, hackers have their own favorite flavor of Unix. However, Kali seems to be quite popular because of its support for penetration testing.

Unix operating systems are usually free, too, and have exhaustive documentation available on the web. Since Unix is open source, many of the tools developed for it are also open source and free.

There is still another reason why you should learn Unix: some of the best hacking tools are open source and originally written for Unix. Once you learn how to use them, and become familiar with Unix, you can modify them or start developing your own tools.

When you download Unix, you will probably be downloading it as an .ISO file, which you can burn to a CD or copy to a flash drive. This is the full image you need for installation.

If you don't have a computer or laptop that you can dedicate to just running Unix, there are other options. One is VirtualBox, that lets you run Unix through a virtual machine on your computer, or just from the CD drive or flash drive (you can't make any permanent changes to settings, Unix source code, etc. this way, but you can experiment).

Another cool option just for learning how to use Unix is one of the online Unix simulators. Coding Ground offers a Unix shell simulator, as well as just about any other kind of online simulator you need to learn a programming language.

As a hacker, you will spend a good deal of your time working with the Unix **CLI**, or **Command Line Interface**. With a command line interface you type in your commands and instructions, as opposed to working with a **GUI**, or **Graphical User Interface**.

Using the CLI, you can do everything you do with a GUI, like copying files from one directory to another, or searching through files and placing copies of only the files that have the word “ chapter ” in them into a new directory.

Once you get the hang of Unix, its time to learn how to write shell scripts. As is typical in Unix, there are a variety of shells out there, with bash (Bourne Again Shell) being popular among hackers. A shell script is similar to a program that includes operating system commands, and hackers use them not just to develop hacking tools, but also to automate boring, repetitive tasks that require interfacing with the operating system.

There are a tremendous number of sites with tutorials and examples for writing shell scripts, and if you already know how to program, then you will find it quite easy to work with.

Be sure to check out the online resources for links to some of the popular flavors of Unix, as well as excellent tools to help you learn how to use Unix effectively.

Online Resources:

Unix: <http://www.unix.org/>

Linux: <https://www.linux.com/>

<https://www.kali.org/>

Fedora: <https://getfedora.org/>

FreeBSD: <https://www.freebsd.org/where.html>

Ubuntu: <http://www.ubuntu.com/>

VirtualBox: <https://www.virtualbox.org/wiki/Downloads>

Coding Ground: <http://www.tutorialspoint.com/codingground.htm>

JSLinux: <http://bellard.org/jslinux/>

How to Write Bash Programs:

<http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>

Chapter 9: Where to Go from Here

The next step on your journey is to learn a programming language. If you already know how to program, then let ' s be more specific: definitely learn Python. Python is popular among the hacker community, and like Unix it is free and open source.

The main page for Python include downloads, documentations, tutorials – everything a beginner needs to get started, and everything a programmer needs to hit the ground running with it. If you have Unix, you have Python and thus don ' t even need to download it. Coding Ground has a few different releases of Python available to work with online, as well as tutorials, reference materials, etc.

Other useful languages are the old standbys like C/C++, Java, and Perl – among others. The more languages you are familiar with, the more knowledgeable and flexible you will be. Always, always be ready to at least get your feet wet with promising new languages that come out.

Did you know there is a website for hackers that lets you test out your skills? It ' s called Hack This Site, and has tutorials, missions, and a discussion board. This is a great way to test out your skills while minimizing your chances of getting into trouble by hacking the wrong server.

One of the interesting things about the hacker community is their open-source mindset: they will share tips, tools, scripts, etc. with others who are interested in the same thing. That is another reason why you are encouraged to register for Hack This Site.

If you are going to start communicating with other hackers, just be aware that, like many fields of interest, hackers can spot a noobie (or newbie, or simply a person who does not have extensive skill or knowledge on a subject) and many do not have any patience with noobies that ask questions that a simple Google search can answer. Another tip for getting the most out of hacker discussion boards is to never pretend to know more than you do, and be humble. You can ' t learn anything when you already think you know it all.

Before long, you are going to be able to not just use hacking tools intelligently, but begin developing your own. You should start by trying to understand the algorithm, or sequence of steps, that the tool follows. Once you have a firm grasp on that, dig into the source code to find out exactly how they got the

computer to execute those steps. Think of it as dissecting a program!

When you see a command you don't recognize, look it up. Don't stop until you understand exactly what that line of code is doing. Then move on to the next line.

The first time you do this, it will take forever and you will get frustrated. You will probably not get it done in one sitting, but hang in there. After your first program dissection, you will have gained a tremendous amount of information. The next program you dissect won't take nearly as long ... and before you know it, you will recognize most of the commands and options being used in any program you dissect!

Now, for a warning: remembering how Winnie the Pooh loved honey, and always had a honeypot nearby? Well, there are honeypots used to catch hackers. Some companies setup something on their system that hackers can access – as a trap! Maybe they have been made aware of recent intrusions, or they are just on the lookout. Regardless of the motivation, the goal is to find out who is messing around with the system.

There is another type of honeypot that even white hackers need to be careful about: government webpages that have been set up specifically to lure hackers in. There are other types of honeypots – music piracy, child pornography, etc. – but our concern here is what this means to hackers. Don't hack where you don't have legal permission, unless you want to suffer the consequence.

Online Resources:

Python: <https://www.python.org/>

Coding Ground: http://www.tutorialspoint.com/execute_python3_online.php

Hack this Site: <https://www.hackthissite.org/>

Recent Article on Honeypots:

<http://thehackernews.com/2015/02/pirate-bay-fbi-conspiracy.html>

Conclusion

Thank you again for downloading this book!

I trust this book has helped you learn the basics of ethical hacking, and that you enjoyed learning it, too. You now have a good foundation to build on, and I wish you the very best!

Finally, if you enjoyed this book, then I'd like to ask you for a favor. Would you be kind enough to leave a review for this book on Amazon? It'd be greatly appreciated!

Check out my other books on Amazon:

