The not so short introduction to Hacking

An almost complete guide

by mathijssch@gmail.com

Contents

1 Introduction

First of all, thanks for downloading this document and reading it. This document will help you start off in the wonderful world of hacking, by teaching you somewhat advanced knowledge about computers, a few tips and tricks and of course practical information on how to test your newly learned skills in the real world.

When I first started learning about programming there was no Google or Wikipedia, and hacking was basically a synonym for criminal. I had to get my knowledge from dull books in the library, long software manuals and some friends who 'knew a thing about computers. After a few years I hardly knew any people who could teach me about computers, now instead they came to me for help. But where to go from there?

Luckily now we have a vast source of information, called The Internet. Also it helps communication between people with common interests. We can be connected to the internet 24 hours a day, 7 days a week, and if only we had the time and brain-capacity, we could learn anything we want by just Googleing for it. To make it even easier for you, I have put a lot of this information into one document, allowing you to read it from page 1 to the end.

1.1 Is this guide for me?

There are a number of different people who might be reading this. Of course there is the 12-year old boy who knows this girls e-mail address and would like to know the password to it, and is looking for a quick and easy way to do it, without having to do any effort. Sadly, this type of person is found all over most of the hacking-related websites and IRC-channels I know of. For all of you who can identify here: This guide is not meant for you, but if you want to ever be able to do nice stuff with computers, feel free to read on.

The intended audience of this guide is the curious pc-enthusiast who has some basic experience with computers, and wants to know more about the internals of how this wonderful device works, and how to use creativity and knowledge to make them do what you want.

1.2 How can I read this guide?

The guide is divided into chapters, which, if read in sequence, should not be too hard to understand. If you already know a lot about a certain subject, feel free to skip the chapter and read on. Some chapters will require specific knowledge, and the reader will be notified by this in a box on the first page of the chapter, referring to the chapters which should be read first.

The first chapters provide you with a theoretical background, the other chapters are about the actual hacking, putting your newly learnt knowledge to use and showing state-of-the-art techniques.

Have fun reading and learning!

Because some of the information given in this guide can be used for purposes which in some countrys may be criminal, Im unfortunately obliged to include this in my guide: I, the author, am in no way responsible for anything the readers do with the information provided in this guide, nor can I be held responsible for the correctness of this information.

2 What is hacking

In this chapter I will try to give an honest peek into the hacker world, what they did before, what they do know, and they could be doing in the future. Of course this hasn't really got anything to do with the actual technical stuff, so feel free to skip this chapter, but remember you won't be learning all about the culture you are about to take part in.

2.1 Definition of hacker

If you ask a random person on the street what a hacker is, they might recall ever seeing the word in connection to some criminal who 'hacked' some website and stole for example creditcard-data. This is the common image the media sketches of the 'hacker'. The somewhat more informed person might think that a hacker is not really a criminal but somebody with a lot of knowledge about computers and security. Of course this second definition is a lot better then the first one, but I still dont think it catches the essence of what makes one a hacker.

First of all, hacking hasn't necessarily got to do with computers. There have been hackers in the Medieval Ages and maybe even in the Stone Ages. The fact that they used other means to express their skills and knowledge doesn't make them less than any hacker in the modern ages. We are just blessed with the fact that at this moment we are all surrounded by technology, a lot of people even are dependent of it.

But then just what means being a hacker exactly? I don't want to give a single definition that I believe is *the* definition of a hacker, I don't think there is such a thing. But I do believe hacking is about using your creativity and knowledge to overcome limitations, and to approach things with an uncommon view. This is also sometimes called 'Thinking outside the box'. Technology is a very good field to express these skills, whether its cogwheels and wood or hightech quantumcomputers. To be able to be creative with the technology you are handling, you will need to know a lot of its ins and outs, which is what this guide is all about. Of course not all ins and outs are explained because such a book would fill your entire house.

Hacking hasn't got anything to do with breaking the law. It cán be used to break the law though, but then again, your hands can also be used to break the law but should that mean you aren't allowed to use them? Learning about hacking is no crime until you use it for illegal purposes.

There are a lot of different definitions surrounding the word hacker. On many websites about hacking the terms 'whitehat', 'greyhat' and 'blackhat' are used. These names haven't got anything to do with some kind of fashion-preference amongst hackers, but they refer to the nature of the activities hackers involve themselves in. For example, 'whitehat'-hackers strictly follow the law, and can for example be found in a professional context, acting out professions such as Security- or Information Auditor.

On the other side we find the 'blackhat'-hackers. They use their knowledge for illegal activities. We just said a few paragraphs ago that hacking hasn't got anything to do with breaking the law. From now on we will refer to this category of people as 'crackers'. You might have seen the quote 'Hackers build things, crackers break them.' somewhere that illustrates this definition.

Of course, there are always people who don't always break the law, but also use their skills for legal use. This group is called 'greyhat'-hackers. Note that this group is still taking part in criminal activities, so from the law's viewpoint being a grey- or blackhat is practically the same.

Hacker	A person who uses his creativity and knowledge to overcome limitations, often in technological contexts. (See the text above)
Cracker	A hacker who uses his skills for illegal purposes.
Scriptkiddie	Term used to define people who merely use prebuilt tools and scripts (hence the name) to 'hack' into computers and such. These people don't really fit the description of cracker either, because there isn't any creativity nor knowledge involved in blindly using methods designed by others.
Phreaker	A hacker that expresses his skills in the field of telephony networks instead of computers. More about this in the next paragraph; 'A short history about hacking'.

Table 1: Some hacker-related definitions

I already said it before, but i cannot emphasize this too much: do *not* use the skills you will be learning in this guide for illegal purposes. Think about it this way: you might learn a few tricks and try them out on fbi.gov. Some people might make you believe that by doing this, the FBI will want to hire you to become one of their elite computer specialists, because of the skills you demonstrated. This is all but true, the most likely thing that will happen is the feds knocking on your door the next day and you being prosecuted. If you dó want to be hired for your hacking skills, keep it legal and practice on your own computers (See the next chapter about setting up your own box to try out your skills). A good thing to do might be to check your country's law on computer criminality, to make sure you aren't getting yourself into trouble.

2.2 A short history about hacking

Being a hacker also means being part of the hacker culture, whether you like it or not. In this paragraph I will summarize noticable events in the history of hacking. Since this guide focusses on hacking in the software-context, I will limit the timeline to that of computers in general. You could say a person like Leonardo Da Vinci was a hacker as well, but this guide isn't made for teaching history.

1960-1970 – The first hackers
In the ages of the first electronic computers like the ENIAC and PDP11, every computer-programmer could be considered a hacker. During these years there were no integrated development environments, no high level programming languages, just the programmer and the machine. The term hacker wasn't used yet, they just called themselves computer programmers.
Most sources give the credit for first hackers to the group of students at the MIT's Artificial Intelligence lab, who were playing around with software used by a very advanced miniature railroad switching system. They were allowed access to the university's supercomputers, which in these days was a big privilege. This was the first time computers were used for anything besides scienfific or military usage.

1970-1980 – To phreak or not to phreak
With the rise of the telephone system a new technological system presented itself for hackers to try out their skills. When Bell, the largest telephone operator in the United States, switched from human operators to a computer managed phone system, the shit really hit the fan. This system used frequency notes to operate the computers, for example a 2600Hz tone caused the line to open for a new call, without charges. It didn't take very long for the first generation of telephone-hackers to find out this 'feature' and a new movement of hackers hatched, calling themselves phreakers.
Perhaps the best known hack from this period was using the whistle from a box of Cap'n Crunch cereals, which emitted a perfect 2600Hz tone, to make free calls.

1980-1990 – Hacker uprising
Until the personal computer, hacking was limited to phreakers or users of mainframe computers. But when the computer became accessible for 'normal' people, hacking really started to take off. This also meant the interest of Hollywood. In this period a lot of movies about 'hacking' were produced, such as War Games (1983). With this new generation of younger hackers the cult kept growing, and soon groups of hackers started to form. Using BBS's (Bulletin Board System) they could communicate with each other, anonymously, by using self-picked handles. Most boards were not publically accessible to keep the knowledge from the masses, preventing abuse.
Unfortunately, when hacking became more popular, it also caused more crackers to appear and damage the profile of the hacking scene. The media loved it when a 'hacker' was arrested for breaking into a bank computer system or something similar. Also, the first viruses were released in this period, the first one hitting the University of Delaware in 1987.

1990-2000 – The Internet
We will talk about the internet in detail in the chapter about Networks, but in the history of hacking, the emerging of the Internet as a global communication network gave hackers a vast playground to put their skills to the test. The way information could be shared grew enormously, and the concept *scriptkiddie* was born.
Another movie hit the screens in 1995: 'Hackers', in which curiously dressed up teenagers with techno-obsessions get

caught in a cat-and-mouse game with the FBI.

2000-now – Today's hacking
An increase in the use of the Internet, e-mail and telephony meant a natural increase in hacking activity. New fields of interest have opened itself up, such as VOIP, Bluetooth mobile phones, etc. The rest of this guide will focus on modern hacking techniques mostly, so read on if you want to know more.

2.3 Well known hackers

There have been some cult-figures in this rich history of hacking, I have picked out a handfull to describe what they did and what they are doing now, sorted chronologically by birthdate. There are much better and more complete lists out there, I have included it just to have some examples of different styles of being a hacker.

Richard Stallman

(born March 16, 1953)
Regarded as the most influential person in the open source community, Richard Matthew Stallman is the founder of the GNU Project and the Free Software Foundation. Richard was also one of the AI programmers at MIT of which i spoke in the previous paragraph. Today he is probably one of the most active activists in the field of free software, and has received numerous honorary doctorates and professorships.

Eric Steven Raymond

(born December 4, 1957)
Not a hacker in the hollywood kind of way, but a real open source guru and computer specialist in general. Was first active in the hacking scene in the late 70's, and is responsible for many lines of open source software. Within the hacker culture though, he is most known for his adoption of the Jargon File, a glossary of hacker slang.
Furthermore he wrote a lot of documentation and how-to's, mainly for linux programs and distrubutions. In 2003 he wrote the book 'The Art of Unix Programming', and at the moment of writing this guide he is a high-profile representative for the Open Source community.

Kevin Mitnick

(born October 6, 1963)
Probably the most famous hacker out there for the main public, also the first person to serve time in prison for committing computercrimes, five years in total. He started hacking at 12 years old, using social engineering. In high school he picked up phreaking, and soon was notorious in the phreaking scene.
Now, in 2007, Kevin is a professional security consultant with Mitnick Security Consulting, LLC. He also co-authored a few books on computer security and social engineering.

Gary McKinnon

(born 1966, exact date unknown)
A British hacker that is accused of hacking into NASA, the US Army, US Navy, Department of Defense and the US Air Force. Was arrested in 2002 by the British National Hi-Tech Crime Unit, and was later banned from using any computer with internet access. Although he might sound like a real bad-ass blackhat hacker, he later admitted that he was simply using scripts looking for blank/default passwords, which gave him access to the systems listed above. Technically this makes him a scriptkiddie, who just got a lot of media attention.

Theo de Raadt

(born May 19, 1968)
If you have ever used any of the *BSD operating systems then you probably already heard of this guy. Theo de Raadt was one of the founders of the NetBSD project in 1993, and the founder of OpenBSD in 1995. Another great piece of software from the hands of Theo is OpenSSH[1], which we will see again in the chapter about Networks.
OpenBSD is generaly known as the most secure operating system, and best of all, its free! Of course this means a great deal to hackers all over the world.

2.4 Further reading

I could write on and on about how to become a 'real' hacker and what happened inside the hackerscene over the last few decades, but thats outside the scope of this guide. Here are a few resources that you might want to read if you are interested in all this:
`http://www.catb.org/~esr/faqs/hacker-howto.html`
`http://www.paulgraham.com/gba.html`
`http://web.archive.org/web/20010708111438/http://www.claws-and-paws.com/personal/hacking/17steps.shtml`
`http://www.robson.org/gary/writing/becomeahacker.html`

[1] A set of computer programs providing encrypted communication sessions over a computer network using the SSH protocol.

3 Setting up your own testbox

Since just reading information doesn't nearly make you learn as much as hands-on excersise, we will first get ourselves geared up with a working GNU/Linux installation. Most if not all of the following chapters will include practical assignments, to make you comfortable with using the technology you just learned. These assignments wont be step-by-step guides – neither will this chapter be – so they will often require you to do some research of your own. Also, if you run into any trouble with your specific installation, don't contact me personally but use for example Google to find the answers.

If you are already running linux and want to do all the practical excersises on your own pc then just skip this chapter.

3.1 What is GNU/Linux

Some of you might have never used or seen GNU/Linux before, and some of you might have been using it for years. Don't be afraid if this is the first time you will be using it, it doesn't bite (much) and if used responsibly you can't really mess that much up while not knowing what you're doing.

One of the main features of GNU/Linux is that it's opensource software. This means that anybody can view the source and make changes where he/she sees fit. This also means that (for most of the distributions) there is no single company responsible for solving bugs and exploits, we will come back to this later on in this guide.

Another reason to use GNU/Linux for hacking is its highly modular and configurable nature. You could build your own distribution, completely suited to your needs, leaving packages out which otherwise just sit there and occupy valuable bytes on your harddrive. Also, this feature means it can be modified to run on practically any piece of hardware, whether its your old 286 machine or the latest high-end workstation.

3.2 Physical or virtual?

The easiest way to get yourself geared up with a working GNU/Linux workstation is by installing VMWare Server on your current operating system, and installing a common distribution on a virtual machine. Now this is only a recommendation, of course, as a hacker you should be able to make your own choices. The other way is installing it on a physical computer, which of course requires you to have a spare computer that can run linux.

A really nice feature of having a virtual machine as opposed to a physical machine, is the ability to make a complete snapshot of the system in its current state, which can be loaded at any time later on. Imagine you made a snapshot just after installing your new GNU/Linux system, and while installing additional software, you accidentally delete some important files and your system stops working. You could of course boot it up with a diagnostic CD and start troubleshooting, but with the ease of one click you can reset the virtual machine to the last snapshot and its working again!

VMWare Server runs on most 32- and 64-bit operating systems such as Windows, GNU/Linux and Solaris. This probably means you can run it on your current desktop computer. Because VMWare Server is free of charge, unfortunately not opensource but still, you can download it at `http://www.vmware.com/products/server`. You might get the impression I work at VMWare and want to advertise for this great product, but thats just because I use it myself and am quite happy with it. Feel free to look around for other virtualisation software products, such as Xen[2] or KVM[3].

3.3 Which distribution?

GNU/Linux has many different implementations, called distributions. Some are aimed at beginning users, some for the powerusers who want control over every aspect of their operating system. I personally am a Debian user, and will therefore recommend this distribution for use troughout this guide. Debian is a distribution which is fairly easy to use, and is known for its good stability and security. Also, since we will only be using the console, there is no need for any particular desktop manager such as Gnome, KDE, XFce, etc.

[2] A free software virtual machine monitor that runs on GNU/Linux and other Unix-like operatins sytems. More info on `http://www.xen.org`

[3] A Linux kernal virtualisation infrastructure, more info on `http://kvm.qumranet.com`

If for some reason you don't want to install Debian GNU/Linux, feel free to look around for other distributions such as SuSE, RedHat, Gentoo, and many many others.

3.4 Installation

I assume by now you have either have a powered on physical computer, or a virtual machine set up with the default settings on which you will be installing your chosen distribution.

4 Security

Brainstorm: what is security, authentication, authorisation, Types of attacks (MITM, etc), Security by obscurity, Social engineering

5 Networking and Protocols

Brainstorm: Networking basics, Ports, DHCP, DNS , DNSSec, TCP/IP, IPSec, Routing, Firewalls, Proxy

6 Programming

Brainstorm: HTML, Javascript, PHP, Perl, JAVA, C++, C, Assembly

7 Encryption

Brainstorm: Binary / Hex / ASCII, MD5, AES, DES / 3DES

8 Software

Brainstorm: Virusses & Trojans, Exploits, Backdoors

9 Hacking Tools

Brainstorm: NMAP, Nikto, MetaSploit

10 Hacking a local computer

Brainstorm: BIOS, Windows, Linux

11 Hacking a remote computer

Brainstorm: Footprinting, Scanning & Enumeration, Denial Of Service

12 Hacking other stuff

Brainstorm: PBX / Dialup hacking, Wireless Networks, VPN, IRC, Email

13 Defense

Brainstorm: Securing your own pc, Securing a (web)server, Securing a network

14 Other useful topics

(Maybe these can fit in other categories)
Brainstorm: Telnet & SSH, SSL, Port Knocking, One Time Password systems, Shellcode, Session Hijacking, Phishing, Reverse Engineering, Buffer Overflows

15 Keeping up to date

Brainstorm: Newsgroups, Mailinglists, Websites

Appendix A: Lists

List of Tables

List of Figures

References

`http://en.wikipedia.org/wiki/Hacker`

16 Index