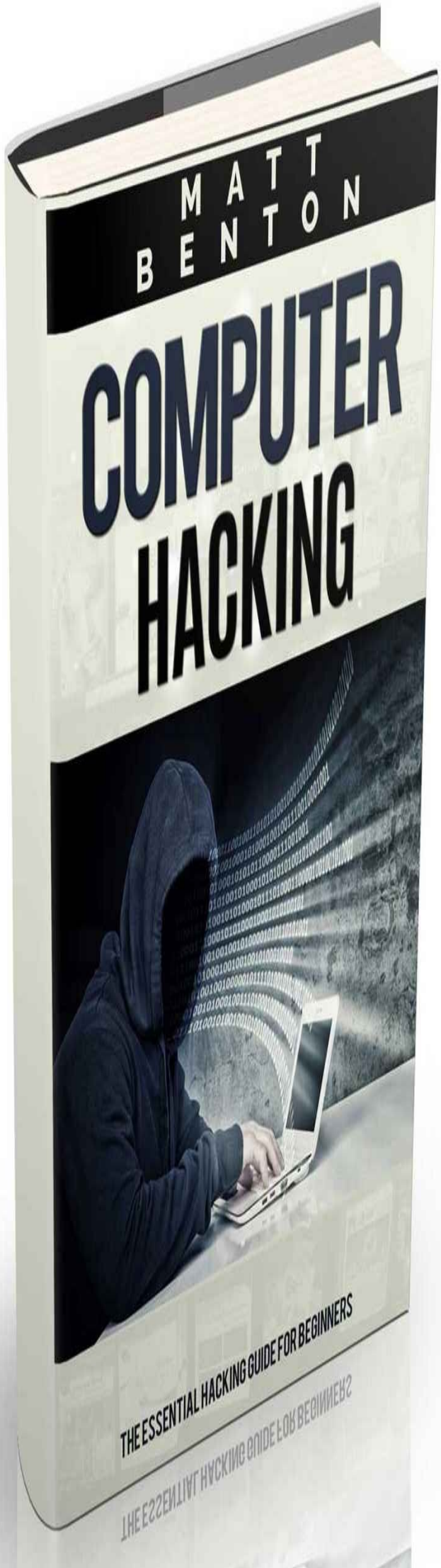
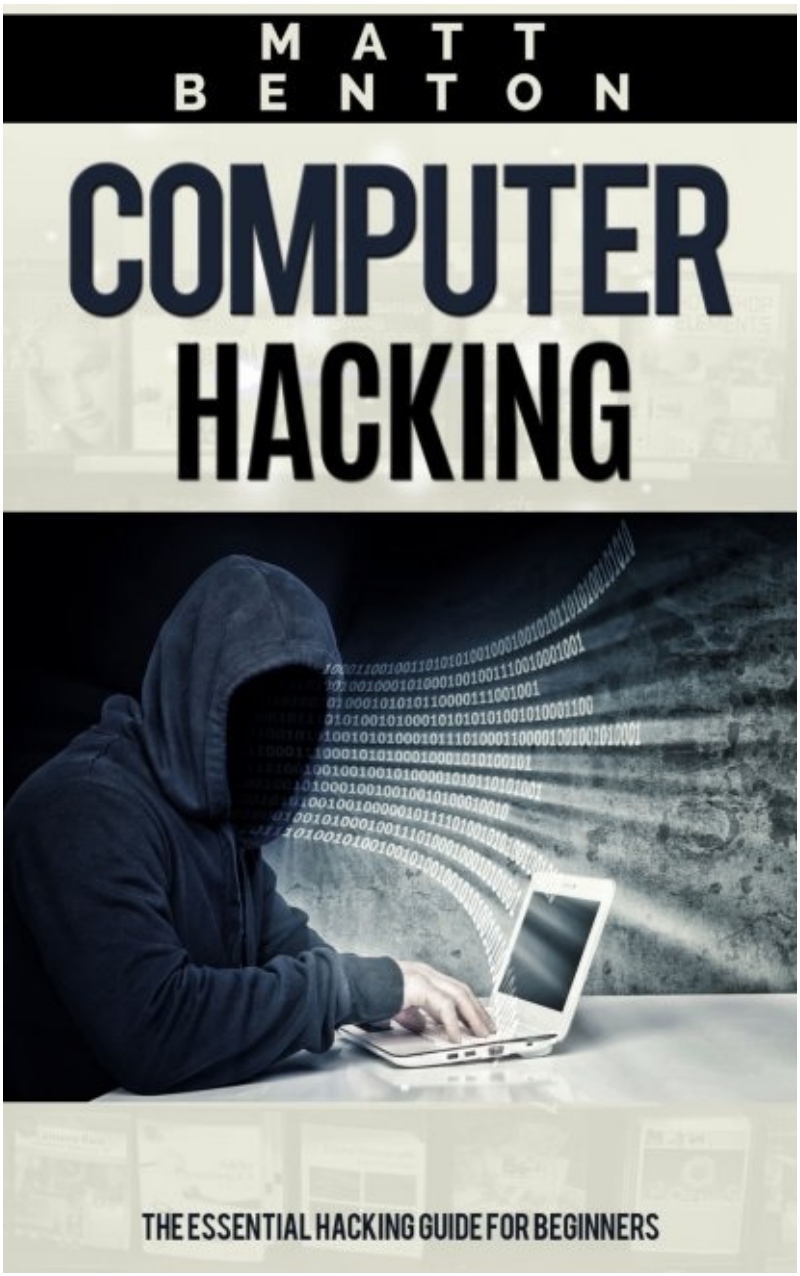

SQL
Beginner's Guide For Coding SQL
Stephen Hoffman


MATT
BENTON
COMPUTER
HACKING
THE ESSENTIAL HACKING GUIDE FOR BEGINNERS

Computer Hacking

The Ultimate Guide to Learn Computer Hacking and SQL (computer programming, hacking, hacking exposed, hacking the system, database programming)



Computer Hacking

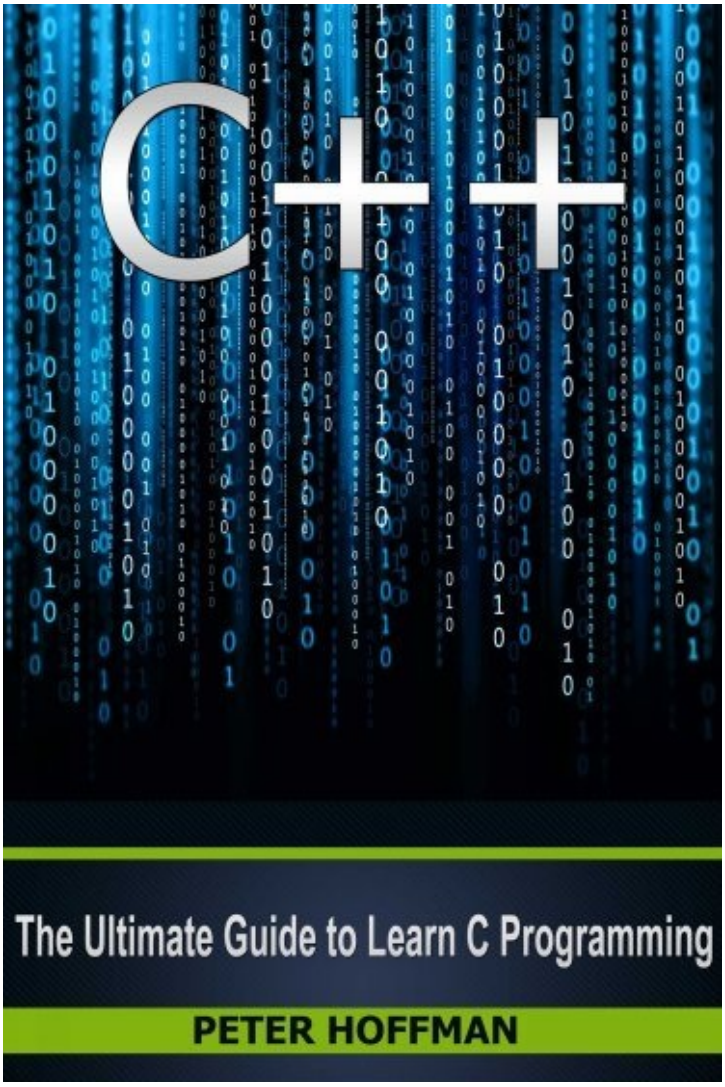
The Essential Hacking Guide for Beginners

MATT BENTON

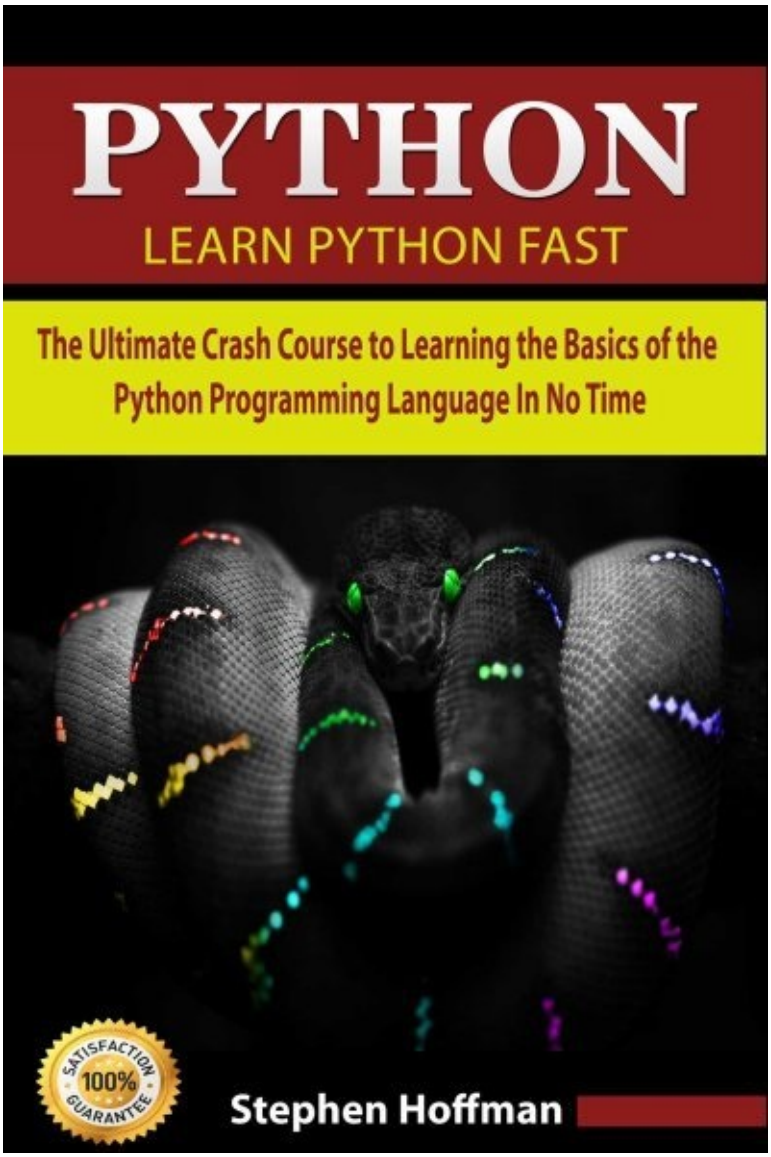
Contents

I think next books will also be interesting for you:

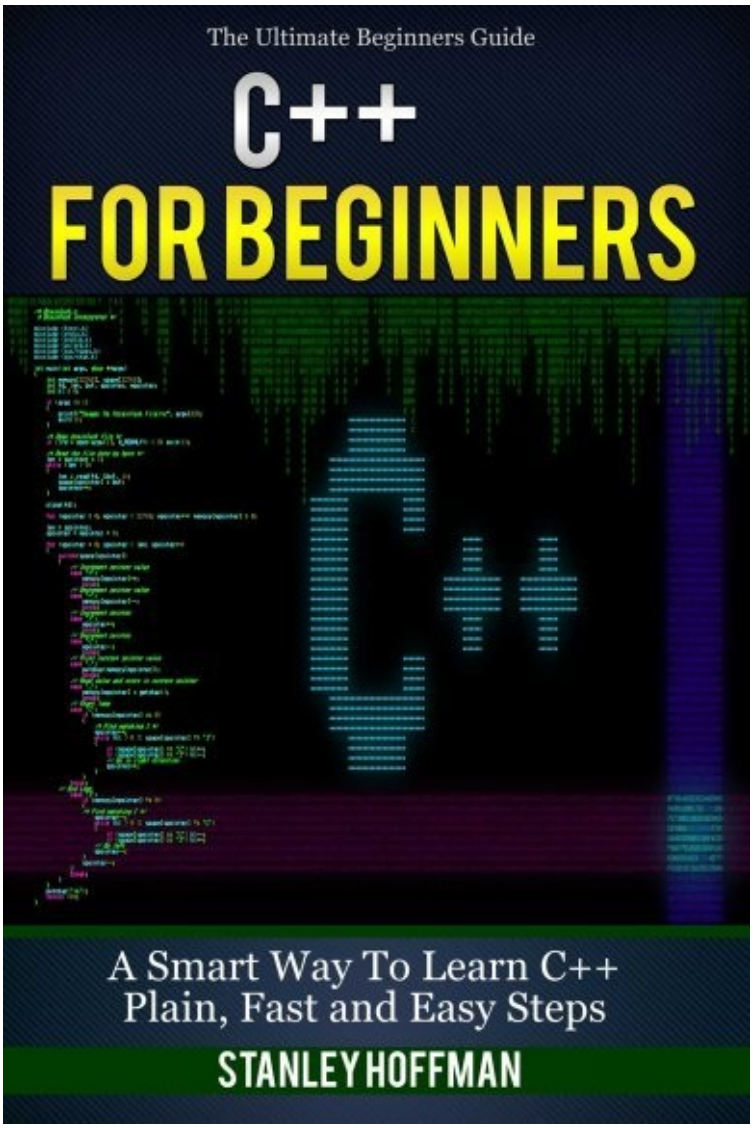
C++



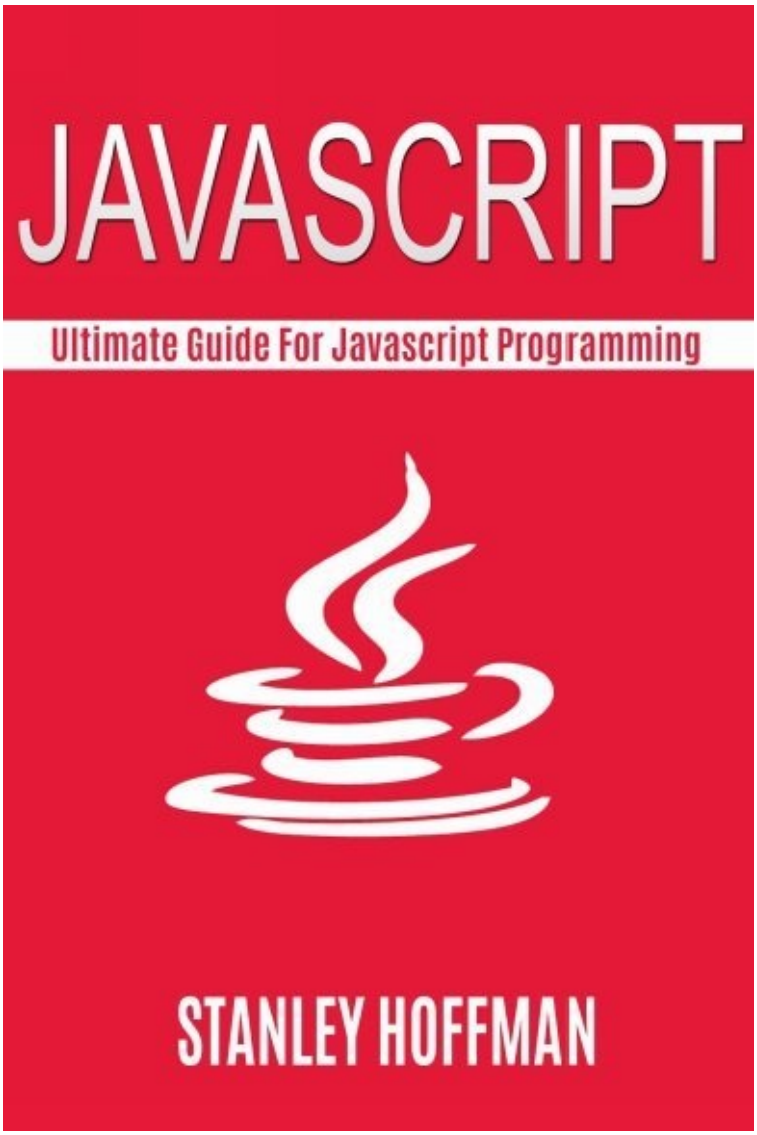
Python


PYTHON
LEARN PYTHON FAST
The Ultimate Crash Course to Learning the Basics of the Python Programming Language In No Time
SATISFACTION 100% GUARANTEE
Stephen Hoffman

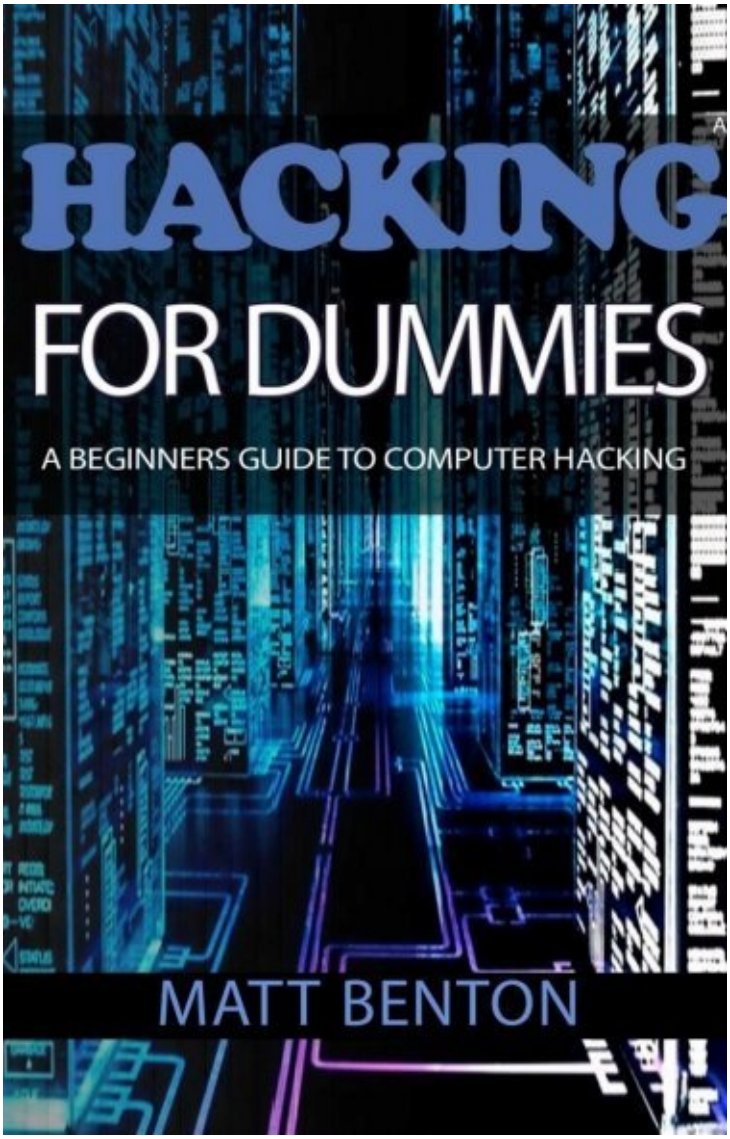
C++


The Ultimate Beginners Guide
C++
FOR BEGINNERS
A Smart Way To Learn C++
Plain, Fast and Easy Steps
STANLEY HOFFMAN

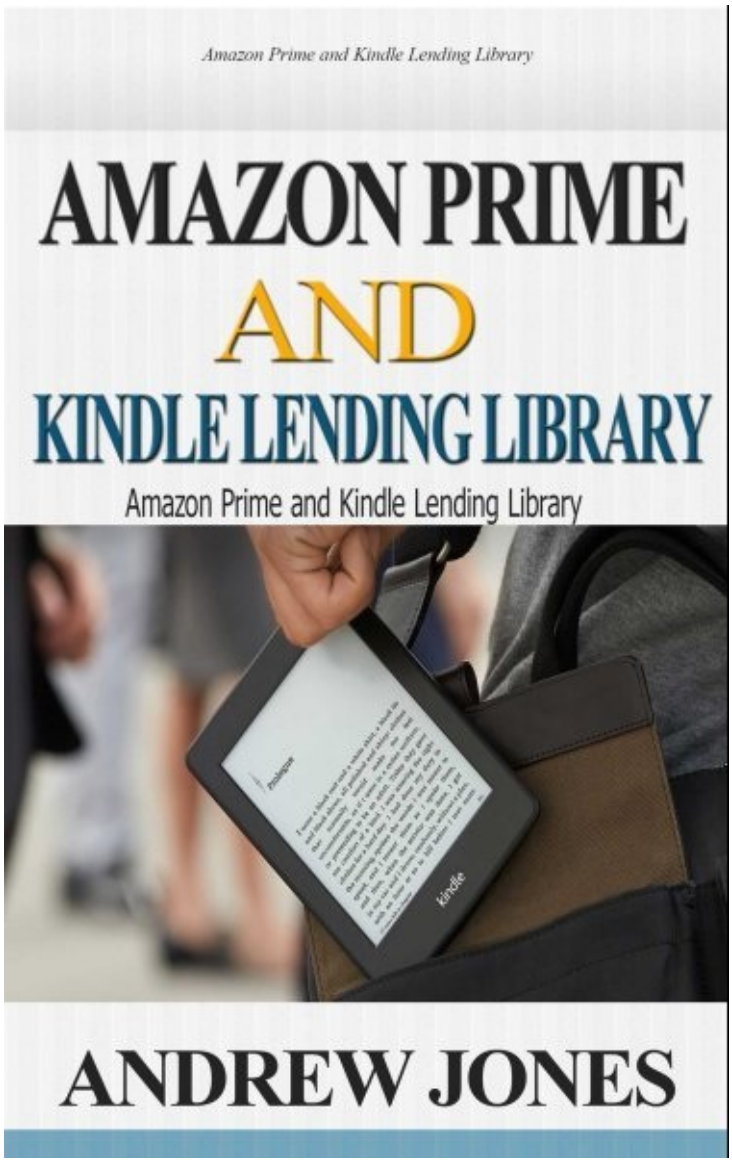
Javascript


JAVASCRIPT
Ultimate Guide For Javascript Programming
STANLEY HOFFMAN

[Hacking for Dummies]



[Amazon Prime and Kindle Lending Library]



Introduction – What is Hacking?

Hacking is the act of gaining unauthorized access to a computer system, and can include viewing or copying data, or even creating new data. Often hacking is understood to be a way of maliciously disrupting a computer system, copying information, or leaving behind a virus that destroys data.

There are many different reasons why hacking takes place, and these reasons range from wanting to disrupt a system due to ideology (so hacking as a means of protesting); wanting to gain profit for example in order to commit credit card fraud; or simply hacking for the sake of enjoyment and amusement.

There is some controversy about the definition of the word ‘hacker’ because those that try to prevent such breaches in security from taking place, or seek to recover lost files, can also be known as hackers. Thus, some people believe that the correct term for malicious system security breaches is in fact ‘cracking’ and that ‘hacking’ is the correct word to use for those who fight against such malicious exploitation of computer weaknesses.

However, in the popular imagination and in general conversation, the word ‘hacker’ is mainly understood to refer to the ‘bad’ method of breaking through computer security. The two processes share many common skills, as regardless of motivation (whether to steal or protect, break in to or save, computer data) the same understanding of computers is required.

Hacking is more than simply a pastime for those who are interested in technology, and more than simply an illegal activity used for personal gain and with malicious intent, although both of these motivations do make up much of hacking activity. In fact, hacking is its own subculture, and members of the community feel very strongly about their ideologies, techniques and social relationships in the computer underworld.

There are many hacking groups and conventions, such as SummerCon, DEF CON, HoHoCon, ShmooCon, BlackHat, Chaos Communication and Hacker Halted, and local hacking communities take their entries into hacking competitions very seriously. Unsurprisingly there are also numerous online groups and forums dedicated to the subject of hacking, and there is certainly a strong community spirit felt by those with similar hacking ideologies.

Furthermore, hackers are often passionate about literary depictions of the hacking

community, and ardently read fictional Cyberpunk and factual hacker magazines.

This book will serve as an introduction to the world of hacking, and will provide insight into some of the key influences, ideologies, groups, concepts, and techniques of hacking.

The first chapter will consider the beginnings of hacking and the influence of the literary genre, Cyberpunk. The second chapter will look at the different types of hackers, and draw a distinction between ethical and unethical hacking. The third chapter will look at the issue of computer security, which is vital to an understanding of hacking.

The final chapter will provide an overview of the various different techniques for hacking, including automated and manual approaches as well as the importance of the cyber confidence trick known as social engineering

Chapter 1 – Hacking and the Influence of Cyberpunk

Michael Bruce Sterling, the American science fiction author, helped establish the popular genre of Cyberpunk. Cyberpunk is a subcategory of science fiction that focuses on the role of technology in a future setting. In this literary and cinematic genre, lower-class citizens are depicted, who have access to, and a great understanding of, advanced technology.

Cyberpunk often explores the role of technology during the breakdown of social order, in which there is an oppressive government restricting and damaging the lives of the general population. Furthermore, artificial intelligence (such as robots or intelligent computers) also plays a significant part in Cyberpunk stories, and the Earth is depicted in the near future in a post-industrial dystopia (the opposite of utopia, and therefore a bleak world characterized by oppression and often social unrest.)

The impact of Cyberpunk in the present-day understanding of hacking is considerable. Science fiction is particularly effective when we can recognize our own world within the fictional representation, and with Cyberpunk we can recognize many of the concerns of the contemporary technological age. Lawrence Person (editor of the science fiction magazine *Nova Express*) describes the typical characters in Cyberpunk:

“Classic cyberpunk characters were marginalized, alienated loners who lived on the edge of society in generally dystropic futures where daily life was impacted by rapid technological change, an ubiquitous data sphere of computerized information, and

invasive modification of the human body."

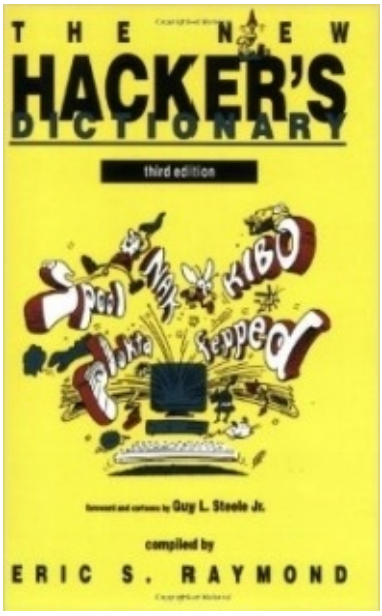
To a contemporary reader, this description of Cyberpunk characters is reminiscent of how hackers are thought of in the popular imagination, and depicted in books and in films. Therefore, the interplay between Cyberpunk characters and how we view real-life hackers is considerable: in many ways our understanding of what a hacker is like is based on how Cyberpunk characters are depicted in fiction. One example of this is how in Cyberpunk the characters often live in filthy conditions, work at night and sleep all day, and do not have any social life beyond chat rooms.

In the present-day imagination when we think of hackers we will often think of a lonely adolescent boy sitting in a darkened room behind a computer screen. In fact, Michael Bruce Sterling, who was one of the first science-fiction writers who dealt with Cyberpunk, has also shown the most interest in understanding the development of hacking.

Sterling has traced the emergence of hacking, and the associated underground computer network, to the Yippies, a counterculture group who were active in the 1960s and published *Technological Assistance Program,* a newsletter that taught its readership techniques for unauthorized access to telephones, known as phreaking.

Many of the individuals who were involved in the phreaking community are also an active part of the underground hacking community, suggesting that the relationship between the two groups.

Chapter 2 – The Different Types of Hackers



The computer hacking underground contains various different subcategories of hackers. This is mainly due to conflicting ideologies, whereby certain groups calls themselves by a specific name, or call others a specific name, in order to emphasize that they do not agree with the ideologies of others.

The generic word ‘hacker’ therefore, although referring to those who have technical knowledge and are able to gain unauthorized access to computer systems, is rather vague and does not distinguish between those who use different methods or believe certain things.

Instead, separate names have emerged in order to distinguish between groups, and to indicate that not all hackers follow the same rules or ideologies. One way in which this

can be seen, as discussed previously, is the distinction between hackers and crackers, as advocated by Eric S. Raymond in *The New Hacker's Dictionary.*

In this book Raymond compiled a glossary of hackers' computer programming jargon, but those from the hacking community feel that this book is too biased by Raymond's own view of hacking as a malicious practice.

Rather than following the dichotomy of hacker/cracker that Raymond suggested, the general hacking community feels that this is too reductive and instead advocate a wider list of name to reflect the spectrum of beliefs and practices of the large hacking community.

One subcategory of hackers is known as 'white hat hackers' and they break through computer security without a malicious motivation. Examples of why this might be done include doing so to test one's own security effectiveness, or when doing work developing computer security software.

These breaches of security can occur whilst performing vulnerability assessments of computer software as part of a contractual agreement, and is therefore legal. In this way, the slang term 'white hat' references an ethical hacker who does so for positive reasons, in order to protect rather than destroy. There are recognized organizations, such as The International Council of Electronic Commerce Consultants, who provide training and certificates for this area of ethical hacking.

On the other hand, there are 'black hat hackers' who breach computer security systems simply to be malicious, or to gain profit. These hackers are the ones who are also sometimes referred to as crackers. This subcategory form the cliché hackers who are often depicted in films and television, and represent the elusive and little-understood computer criminal who the public fears.

These types of hackers violate computer security in order to destroy, change or steal information, or to prevent authorized users from being able to access the system. In this way they can cause disruption, waste time, and cause distress, but they can also steal significant amounts of money or access confidential information.

Generally a black hat hacker will spend time looking for and discovering faults in programs, or weaknesses in computer systems, but rather than alert the public to these problems they exploit them for personal gain or simply for fun. Once they have accessed a computer system, they can consequently make adjustments that prevent somebody with authorized access from using the system and thus the black hat hackers retain control.

Lying somewhere between the two, not quite a white hat hacker and not quite a black hat hacker, is the gray hat hacker. This is somebody who without being asked to searches the Internet for systems with a weakness or security flaw, and will then notify the administrator and offer to rectify the problem for a fee.

In this way they are not as good as a white hat hacker (because they are demanding a fee, and their services were never requested) but they are also not as bad as a black hat hacker because they do not exploit these weaknesses in order to wreak disruption of steal data. Another way in which gray hat hackers might respond to their discovery of a security weakness is to publish their findings online, so that the general public has access to the information.

In this way they are not performing malicious hacking themselves, but they are publishing the information, which leaves their subject at risk of a security breach. This type of hacking is illegal and also considered unethical, whether or not the gray hat hacker has breached security for personal gain, because they have gained unauthorized access to data and have left the system susceptible to hacking by malicious blat hat hacker groups.

As well as these three main classifications for hacking, which differentiate hackers based on their motivation and what they do about the information they discover, there are various other specific types of hacker. There is a social hierarchy amongst hackers, who are recognized based on their skill.

The highest of these statuses is the elite hacker, and sometimes form into elite groups such as the ‘Masters of Deception.’ On the other end of the scale is a script kiddie, who is still learning and has not yet developed their skills with breaching security systems. A script kiddie uses automated tool written by others, and is therefore simply following a code provided by a more skilled, black hat hacker, and not having to work it out themselves. Usually a script kiddie does not really have any knowledge or understanding of the complicated underlying

technological concepts, and simply follows a plan provided by a more experienced hacker.

Even less experience than a script kiddie is a neophyte, who is a completely new hacker who has very little knowledge of computer technologies or the logic and concepts behind hacking. A blue hat refers to somebody who is used by computer security consulting firms but is not actually a part of the company; the blue hat is used to test a system prior to its launch to determine whether it has sufficient security or will be susceptible to hacking.

A hacktivist (a combination of the words ‘hacker’ and ‘activist’) is a hacker who uses their knowledge of technology and their hacking skills in order to broadcast a political, social or religious message. Hacktivism itself has two subcategories: cyber terrorism (where websites are damaged or services cannot be accessed) and freedom of information (making information available to the public that was previously either undisclosed or stored in an encrypted format.)

Groups of hackers working collectively can include organized criminal gangs, and cyber warfare of nation states. The different subcategories of hackers are indicative of the various ideologies, motivations and techniques that are present in the hacking community.

Chapter 3 – Computer Security

Before we can begin to explore the key concepts and techniques of hacking, it is helpful to first understand the basics of computer security. As hacking is the act of breaking through security measures of computer systems, an understanding of these systems is vital to any hacker who hopes to penetrate them. Computer security is applied to computers, smartphones, computer networks (public and private) and the entire Internet in order to protect devices, data and services.

Digital equipment is protected from unauthorized access by computer security, to ensure that data is not stolen, changed or deleted and to maintain the smooth running of systems. In present-day society, where digital culture forever growing, protecting these systems is extremely important and thus the field of computer security is forever growing and developing. Part of computer security is protecting the physical equipment from theft, whereas the other part of computer security is information security, to protect the data itself (and this is where hacking comes into play.)

However, sometimes these two fields overlap because if there is a breach in physical security (e.g. if a laptop is stolen) then it becomes much easier for the individual to succeed in a breach of information security, since they have the piece of equipment and it is therefore easier to access data than it is remotely.

Cyber security encompasses all security measures in place to protect a computer's data, and includes procedures such as awareness training, penetration testing, and the use of passwords to confirm authorization in order to protect data both when it is in transit and when it is simply being stored. The financial cost of being a victim of a computer security breach is considerable and as a consequence there is a lucrative market for anti-virus and computer security protection.

Computer security is a huge field because of our present-day reliance on technology. Almost every industry uses computers to a greater or lesser extent, and therefore the extent and variety of computer security measures is vast. There are some areas, however, where computer security is particularly important because they are especially vulnerable to breaches in security.

One of these is the area of financial systems, because hackers can make a profit by stealing data and consequently accessing funds. Any website that requires somebody to enter their credit card numbers are often targeted because a hacker can immediately transfer money to their own account, or spend the victim's money online.

Even if the hacker themselves do not directly use the person's bank details, they may also sell the information illegally, in order to distance themselves from the crime and attempt to avoid being caught. It is not only online that a person's data can be stolen; in-store card machines and cash points can also be rigged to collect personal information and thus gain access to funds.

People are becoming increasingly aware of this risk when doing online shopping or entering their card details, and therefore various measures are being put in place including using passwords and answering security questions. The aviation industry is another field in which computer security is of the upmost importance, because the consequences of a breach in security can range from the publication of confidential information, to the loss of expensive equipment and human life.

There are various reasons why the aviation industry may become subject to a computer security attack, depending on the motivation for the crime. These motivations include sabotage and espionage in the military aviation industry, and industrial competition and terrorism in the commercial aviation industry. Air traffic control is one of the aviation industries most vulnerable systems, because any attack can be difficult to trace, and are relatively simple because it only requires a spoof message on the radio.

There are those who seek to exploit computer vulnerabilities (either due to thrill seeking, to make a political/social statement or for financial gain) and of course on the other side those who work to uphold computer security again such threats. Somebody with knowledge of technology, and the ability to hack into computer systems, can therefore either become involved in the illegal and unethical form of hacking (otherwise known as cracking), or serve the other side by identifying threats, improving security measures and alerting companies to the vulnerabilities in their systems.

For those whose aim is to protect computer security, there are various countermeasures to guard against damaging hacking, whereby the risk of being vulnerable to a breach of computer security can be minimized or eliminated. These precautions vary in cost and complexity, but can include: intrusion detection systems (to detect threats and also analyze attacks after the event), the use of account controls (passwords and encryption of data); and the installation of firewalls (providing either hardware or software package filtering of certain forms of attack.

Precautions against a computer system being compromised by attack include making steps to prevent attack, ensure any potential attacks are detected, and the ability to respond to an attack to prevent further damage.

However, despite there being a range of countermeasures available, computer systems still

remain vulnerable to attack and it is certainly not uncommon for a computer to have its security compromised. The first reason why attempted security violations still occur is that the police are often unfamiliar with computer technology and as a result do not have either the skill or the inclination to solve the crimes and apprehend the criminals responsible.

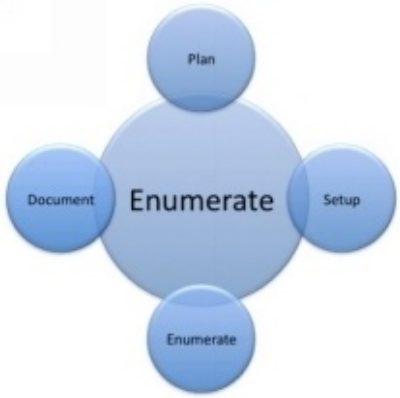
Furthermore, any investigation of such matters requires a search warrant in order for an officer to examine the entire network and this can make the procedure extremely time consuming. Another difficulty in ensuring computer security is that in the age of globalization, in which information can be shared throughout the world using the internet, and technology can spread data extremely easily, identifying and apprehending those responsible is particularly difficult.

The reason for this difficulty is that a hacker might be working from one jurisdiction, while the system they are hacking into is in a different jurisdiction. Furthermore, a hacker can use various techniques (such as a temporary dial-up internet) in order to ensure their anonymity. The third problem is due to the high number of attacks that occur.

Organizations can be subject to many attacks and therefore are unable to pursue every security threat. A computer user would benefit from taking precautionary measures in order to ensure their computer security, as once a breach of security has occurred there is not much that can be done to rectify it.

Chapter 4 – Hacking Techniques

There are various techniques that can be used by hackers in order to gain unauthorized access to a computer system, in order to wreak havoc, steal money or data, or to prevent the system from operating as it is supposed to. The three main methods that are used in order to attack a system that is connected to the Internet are: network enumeration, vulnerability analysis and exploitation.



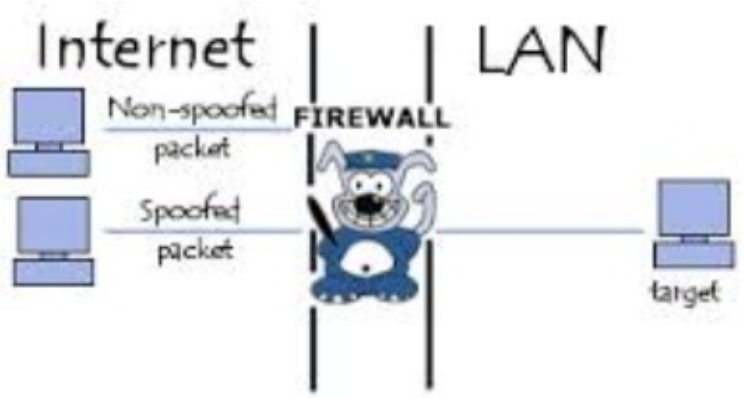
A network enumerator is a program that is used in order to discover the usernames and other information from networked computers. The program discovers any weaknesses in the computer network's security and the findings are reported to a hacker who may then use this information in order to access the network and cause damage (either by stealing data or corrupting the network.)

On the other hand, ethical hackers can use the same process simply to discover any weaknesses in their system in order to tighten security. Another method used is vulnerability analysis, which identifies any points of vulnerability in a system; this information can then be used to either attack the system, or to remove the weakness. Vulnerability analysis can then lead to exploitation, where the hacker uses the vulnerability information in order to breach a computer or system's security.

There are many specific techniques that can be used, but they all employ the main concepts and methods described above. The first more specific example of a hacking technique is a vulnerability scanner, which is a program used to check a network for susceptibility to attack. A port scanner can also be used, which identifies avenues of access to a computer and can establish how to circumnavigate a firewall.

As well as these mechanized devices, hackers can also find these vulnerabilities

themselves, which can be done by manually searching the code of the computer and then testing whether they are right. Brute-force attack is another method by which a hacker can gain unauthorized entry to a computer network, and this involves for example guessing passwords. Password cracking is another hacking technique that uses passwords, but rather than guessing the password, the hacker recovers password information that has been stored in the computer, or transmitted.



A spoofing attack (otherwise known as phishing) is an enemy program, system or website that poses as a trusted one. By falsifying data the hacker is able to masquerade as a trusted system and thus fool a program or user into revealing confidential information such as passwords or bank details. Another hacking technique that is commonly used is a root kit, which is a program that manages to take over the control of an operating system by employing hard to detect methods.

A Trojan horse is yet another technique that is a program which manages to fool systems and users; it works by working in one way while seeming to be doing something else. By using this method a hacker is able to gain unauthorized access to a system and create an access point so that they can re-enter via that established route later on. A computer virus is the most widely recognized form of hacking, as it is the computer threat that most of the public is aware of.

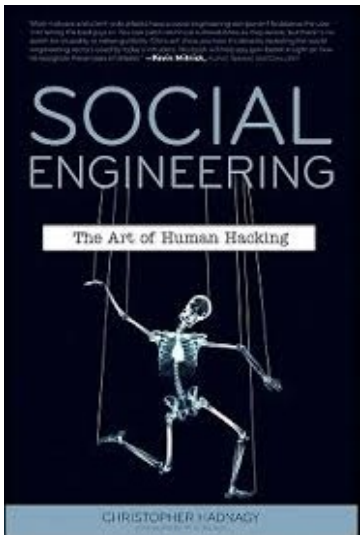
The virus works by self-replicating and implanting itself into documents and code; while some computer viruses are malicious some are merely irritating or harmless. A computer worm is similar in that it is self-replicating, but it is able to enter a computer program without a user inadvertently letting it in, and it does not need to insert itself into present programs.

Finally, a keylogger is a tool that records every keystroke on a given machine, which can later be accessed and viewed by the hacker. This is usually to enable the hacker to access confidential information that has been typed by the victim. In fact, there are some legitimate uses for such a technique, for example some companies use a keylogger in order to detect any dishonesty or fraud committed by an employee.

A large area of computer hacking involves the use of social engineering, whereby in order to circumvent information security a person is manipulated in order to reveal confidential information or to grant access to secure networks. This technique (which includes phishing) is usually only part of a complex routine in a wider fraud scheme, but it is also a dangerous step because human beings are more likely to be won over by a convincing trickster than a machine is.

Social engineering relies on the psychological act of decision-making, and can be thought of as one of the most significant vulnerabilities in a computer security system. There are many different ways in which social engineering can be applied in order to gain unauthorized access to a computer system, and this includes criminals posing as IT technicians who pretend that they are fixing the company computers whilst in fact stealing data.

Another example would be a trickster informing a company that the number of the IT helpdesk has changed, so that when employees phone the number they will willingly disclose their account details thinking that they are talking to somebody who they can trust with the information. These sorts of scenarios come under the category of ‘pretexting’ because making up a believable scenario allows the criminal to access the required information and this leads the victim to disclose the information.



Other professionals that a hacker involved in social engineering could pose as include the police or bank manager, because these are individuals who we believe have the right to be granted any information that they request. Baiting is a subcategory of social engineering because it relies on human psychology in order to work. Baiting is where a victim’s computer security is compromised when an infected disk, device or USB stick is used.

An example of baiting would be for the criminal to post a USB through somebody’s door with a tempting sounding label and simply wait for the curious victim to plug it into their laptop, at which point malware would automatically install and infect their computer. This technique makes the most of the human tendency towards curiosity and greed, because if a label promises erotic images, money or gossip then a victim may find it hard to resist taking a look.

Kevin Mitnick, a once computer criminal who later because a security consultant, has

pointed out that it is much easier and quicker to trick a person into disclosing confidential information than it is to crack into the system using luck, brute force or technical knowledge. Christopher Hadnagy has written a book titled *Social Engineering: The Art of Human Hacking, which* emphasizes the way in which humans are the most vulnerable part of any computer system.

Conclusion

This book has provided an overview of some of the key concepts to do with hacking. We have considered the beginnings of hacking and how it was influenced by the literary tradition of Cyberpunk. It is interesting to note that what was once depicted in science-fiction as an imagined activity in a dystopian society has become real and has gone on to pose a significant threat to computer security and a central concern of information technology experts.

Here we can see the true beginnings of hacking and how what was once a fictional theoretical concept has become a reality with a significant impact on the digital culture. Next we looked at the different types of hackers and noted the distinction between ethical hackers, who perform hacking legally and in order to improve computer security (otherwise known as white hat hackers) and unethical hackers, who use their skills illegally in order to wreak havoc, disrupt services, and steal information and data (otherwise known as black hat hackers.)

With this it is interesting to note how technical skill can be used differently depending on motivation, and how the hacking community is not united by a clear and consistent ideology. Subsequently we looked at computer security in order to better understand the conditions within which hacking takes place. By learning about computer security it is possible to understand the challenges faced by hackers who come up against security measures, as well as the challenges faced by those seeking to maintain the security of their computer systems.

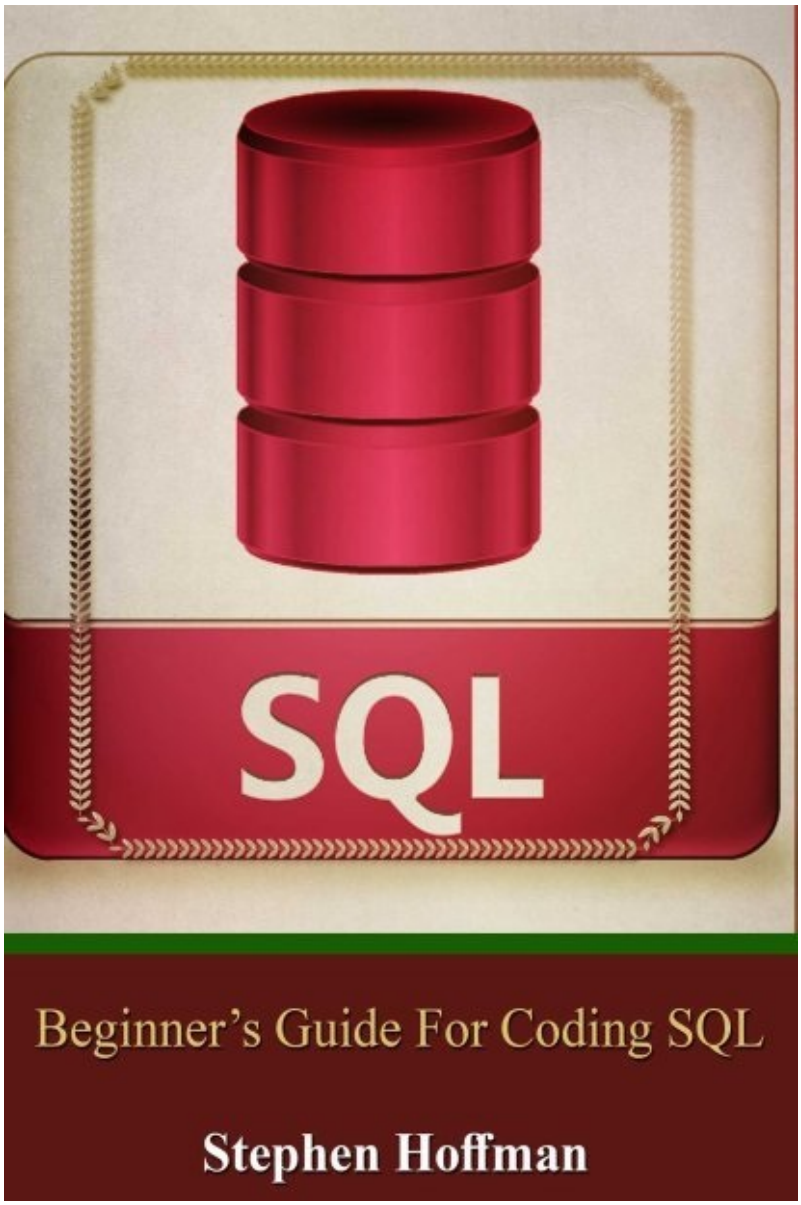
Any introduction to hacking would not be complete without this examination of computer security because the value of maintaining computer security is what motivates ethical hackers, and what unethical hackers are fighting against. Finally we looked at the numerous different hacking techniques that can be utilized, including both automated software that finds and exploits vulnerabilities in computer systems, as well as manual methods for breaking through security measures such as discovering a password through trial and error.

In this chapter we also considered the vast area of hacking that is social engineering, by which a hacker is able to access a secure network by illegally obtaining the information needed. In today's digital culture we are becoming increasingly reliant on technology for everything that we do.

As technology has improved our lives have become easier in many ways, but with this blossoming industry new types of criminals have also been created. A criminal does now not even need to leave the house in order to steal money, but can do so simply by hacking into their victim's computer and accessing confidential data.

Hacking is not always illegal though, and this book has also looked at the ways in which there is an increasing demand for computer experts to become ethical hackers in order to further promote and protect computer security. This introduction to the world of hacking has revealed that hacking is not a simple activity but a huge spectrum of different behaviors that involve a wide range of techniques and motivations.

Moreover, hacking is shown to be an activity that has a strong sense of cult affiliation, in the sense that hackers strongly feel part of the hacking community. As digital culture continues to grow, it seems that both ethical and unethical hacking will become more and more skilled and its impact evermore significant.



Beginner's Guide For Coding SQL

Stephen Hoffman

SQL

Beginner’s Guide for Coding SQL (sql, database programming, computer programming, how to program, sql for dummies)

STEPHEN HOFFMAN

CONTENTS

Introduction

A database serves as a container. It is a storage where programmer scan organize data in a constructive manner. Keep in mind that there is no need to create a database if you are only dealing with few amounts of data. For instance, handling hundreds of Excel spreadsheets require a database while five spreadsheets can be dealt with even without a database. In other words, organizing a threshold amount of data inside a database will save you from complex data management.

A Closer Look On Database

Organizing all data inside a database allows you to easily execute tasks such as querying the data, updating the data, and deleting previous data. Most of all, a database prevents conflicts from having multiple copies of data.

Let us say, for instance, that a company has to manage its overall financial expenses. Basically, there is a specific department to handle this and such department has several people in it. Now if these people have their own copies of the spreadsheets, then it will be somewhat difficult to come up with non-conflicting data considering the possibility of human errors.

A database, on the other hand, will be able to store and organize the data with higher accuracy and lesser conflicts.

Another example to help you understand what a database is, is by looking at your phone’s contact list. Each contact is a data and your smartphone serves as the database. With just a touch of a few buttons, you can easily get the contact you are looking for. You can easily edit the details of each contact as well. This advance data management is way better than listing each contact in a pen-and-paper phone directory.

Types of Databases

The different types of database are as follows:

- Relational Databases
- Object-Oriented Databases
- Document-Based Databases

In this eBook, I will mainly focus on relational databases. Nevertheless, I will not hesitate to include the other two types if deemed necessary.

You might have been introduced to some sort of new database type such as NoSQL. They are considered to be under the category of the document-based databases. These databases are now becoming popular and you might think that it is better to study them instead of learning SQL.

Please do not consider the relational databases to be the things of the past. A lot of databases that we still have today use the relational model. Moreover, SQL is useful when it comes to relational data queries. Besides, the query language of newly popularized databases is still somehow related to SQLsince they were mostly developed after SQL.

In other words, jumping into the new type of databases without understanding SQL will give you tons of challenges along the way. Hence, you have made the right decision of learning SQL first before moving forward to other databases like MongoDB.

Without further ado, let me now bring you to the world of SQL.

Chapter 1 –A Closer Look on Relational Model

The Structured Query Language, or SQL, was originally designed for relational model. It is the language used to define the relationships between multiple data. It is also the database programming language used to query data in a relationalschema.

So when did this all began?

The first relational database was created in the 60s and the programming in this model were primarily defined by mathematics.In fact, this model of database was based on Tuple Relational Calculus and Relational Algebra.

Don't worry, though, there is no need for you to jump into algebra and calculus to handle relational databases. SQL will make relational database programming easier.

Tables, Columns, and Rows

In the relational database, the data is stored in a table. In this model, all tables have names. For instance, you may name a table as "people" while you may name another table as "sales". These names refer to the kind of data stored inside the table. In other words, the table we call "people" contains data of people while the table we call "sales" contains data of sales.

Every table features a set of columns. These columns define the different pieces of data within the table. This what makes the relational model different from other types of databases. You will notice that there are some databases that store data altogether. In a relational model, however, all data is separated by columns and each column has a name. Each column can also restrict the category and size of data that will be added in it. For instance, some columns can only store strings while some columns can only store numbers. The restriction assigned to each column depends on the creator of the database.

Columns are also categorized by being required and not required. A required column means that every row in the table must have a data which corresponds to what the column requires. A not required column, on the other hand, means that we can assert a NULL value for each row under that column. As you go through SQL, you will find that NULL has a lot uses in database programming.

The primary point of using rows is to have retrievable data and containers where you can store data. Why do we store data? Simply because we want to get that data again in the

future and we simply want a place for it to stay for the mean time. Hence, the core idea of a database.

Keys

One of the vital elements of a relational database are keys. We now know that a table has several columns and each column has names. Each column has related properties as well. One of these properties that we can assert in a column is a key.

In a relational database, we always assign a primary key for each row. This special type of key is used to uniquely determine a specific row. This means that the value of that key must remain unique and must not have any duplicates within the table. If, and only if, there is another table which contains that key in our relation model; then, we may merge those two tables together.

When merging two tables, there will always be primary keys that have no duplicates. Upon merging, you will notice that there will be a new column in the table with these keys. These keys are now known as the Foreign keys. They serve as links to the primary keys of the second table, allowing you to conveniently link rows from two separated tables.

Aside from primary keys and foreign keys, we also have the natural keys. Natural keys are global. This means that you cannot change them since they are used all over the globe to identify a single entity. Published books, for instance, always have natural keys which we know as ISBN. Each published book has its own unique ISBN which you need to use as primary keys when handling relational database for books. This is different when handling databases with unnatural keys like emails, people, where you need to invent your own primary keys for each row.

Let us use a database for people and their contact numbers as an example. We cannot consider the name as the primary key. We cannot consider the person’s phone number as the primary key as well. This is because it is possible that there are two or more people in the globe that have the same contact number, first name, or even last name. Hence, we should create the primary key ourselves. This is where the identity column comes in.

The identity column serves as an auto-incrementing column that continues to generate unique keys as we add data in the table. The key combination does not matter since what we only need is a unique key for every identity in our database.

Chapter 2 – Examples of Relational Models

Since I have been using thepeople's database in the previous chapter, I will now give to you a contacts database as a perfect example of a relational model.

A simple database for contacts typically looks like this:

First Name	Last Name	Email Address
John	Doe	johndoe@email.com
Mae	Lee	maelee@email.com

In this table, we have three columns that are looking for specific data. The first column requests for the first name. The second for the last name and the third for the email address of the person.

Our rows, excluding the first row, refers to a single entity. The row for our first entity refers to the person with the name John Doe and his email is johndoe@email.com. The row for our second entity refers to Mae Lee with an email address of maelee@email.com.

This is an example of a database with two entities divided by three columns.

Let us say that John Doe has two email addresses while Mae Lee has three. Some people will end up adding columns to add the additional data. Hence, the table will end up like this.

First Name	Last Name	Email 1	Email 2	Email 3
John	Doe	johndoe@…	johndoe2@…	NULL
Mae	Lee	maelee@…	maelee2@…	maelee3@…

There is a problem here if we use this kind of database design. First, what if the problem we have at hand is to find how many email address each person has? It is quite difficult

and cumbersome to get the answer we need. This is the disadvantage in using a column-base database. The second problem is that we will end up with a lot of NULL values for those people that only have one email address. Third, this design is not practical and you do not need to be a database specialist to realize that.

In order to address these three issues, here is how we can do it.

Key	First Name	Last Name
1	John	Doe
2	Mae	Lee

The first step is to create a table for the entity first. In this table, we assigned the first column for the primary keys. The second step is to create another table where we can store the values for these people's email addresses. The second table will look like this.

Key	Person's Key	Email Address
1	1	johndoe@email.com
2	1	johndoe2@email.com
3	2	maelee@email.com
4	2	maelee2@email.com
5	2	maelee3@email.com

In this table, you will see that there are two columns for the keys. The first column refers to the primary key designated to each email address. The second column refers to the foreign keys from our first table. This is an example of a one-to-many relational model.

In this model, we are able to organize the data in the most practical way possible. It is now easier for us to answer a query on how many emails each person has as well. This method is known as Normalization.

There are different ways to design your database and efficient designs depend on what queries you are trying to answer.

With all that being said, you are now ready to meet SQL.

Chapter 3 – SQL Statements

SQL is a specific database language. We use this language to create statements. These statements are made up of valid words. Some of the words should be defined by SQL while some of the words should be defined by you. Some words you will be asserting in these statements are table names, column names, and variables or data that is inside your tables.

SQL Facts

Before we proceed, I would like to show you some important facts about SQL.

The first thing you need to know about SQL is that it is readable since it is written in the English language. In other words, it is easier to learn SQL more than any other programming languages.

Second is that you can do a lot more using SQL. This is in contrast with most people's impressions that SQL is only used for data queries. This is wrong.

Third is that a valid SQL statement should end with a semi-colon (;). Although some people will tell you that it is no longer needed, I suggest adding it in your queries because it is part of the ANSI standard. Furthermore, it works fine with both old and modern databases. Semi-colons allow you to conveniently assert multiple statements together in what we call a batch.This method works even with modern databases.

The fourth is that SQL is not case-sensitive. You can put your variables in both lower and upper cases. Both ways will work out just fine in the structured query language. One good way to practice this is to assert SQL-specific keywords in uppercases and user-defined keywords in lowercases. This will allow you to efficiently and quickly identify the different keywords within a statement.

SQL Commands

It is important to keep in mind that all SQL statements begin with a command. The command is typically a verb to signify what the database needs to do. One good example of a command used in SQL statements is SELECT.

The SELECT command tells the database to select a set of data. As you can see, it is an English-based word and pretty much obvious of what the system will do. This is how the command looks like when used in an SQL statement.

```
SELECTVALUES
FROMTABLENAME;
```

The example above is a generic example of an SQL statement and I will talk about it piece by piece later on.

Take note that the set of values or variables that you need to assert in a statement depends upon what command you are trying to execute.All these variables come in the statement right after the command.

Naming Values

As what I have already said, the SQL statement is made up of words that are within SQL specification and user-created names. This means that naming values such as keys, table names, columns, and etc. are significant in SQL.

There is no absolute rule in naming things in SQL. People have their own ways to name variables, but I suggest you follow the method I use in this book for the sake of consistency. Here are two naming rules that I will be implementing in this book:

- All table names must be in singular form. (e.g. person, email, phone, user, etc.)

- Column names should never be repeated in the database.

There are some people who prefer to use plural names for their table names. This is not wrong. Nevertheless, I am more comfortable when using singular form table names. As for the column names, there will always be a time when two or more columns from separated tables have the same name. Hence, what I do is to uniquely name each column for disambiguation purposes. For instance, instead of using ID as the column name for primary keys in both the user table and email table; I would rather use the names user_ID and email_ID for each column.

You should also take note that names in SQL are scoped. This means that the database itself must have a name. The tables within the database must have names too. The columns inside every table must be named as well. When calling for the table, its name must be scoped through the database name with the use of period (.) to separate the names. Hence, this is how they are going to look like when asserted in an SQL statement:

- Database.Table
- Table.Column

Creating Data in SQL

I have no intention to teach you how to create a database from scratch. As beginners, your tasks should only circle around adding or updating values to an already existing database. However, I have a feeling that some of you still have that eagerness to learn SQL in order to create the entire application including the database itself. And so, I will introduce that part of SQL, but just enough to help you jumpstart.

The SQL keyword to create a new database is the statement CREATE DATABASE and this falls under ANSI specification. Let us say you want to create a new database for contact numbers of people and you want to name this new database as Contacts. Here is how you are going to do it.

```
CREATE DATABASE Contact;
```

In order for you to work on that database, you need to use it first. The keyword we need here is the USE DATABASE command. This is how we are going to execute it.

```
USE DATABASEContact;
```

If you want to create a table inside the Contact database, then what you need to execute is the CREATE TABLE command and here is you do it.

```
CREATE TABLE Contacts.User (…);
```

After that, you can now put all column names, types, and definitions.

Data Types

Data types are important because it follows column restrictions. Each column has a rule on which type of data you can store in it.

Here are some of the major data types and their short descriptions.

Data Type	Description
CHARACTER	Used to define the number of characters allowed in the column. No more, no less.
CHARACTER VARYING	Used to define the maximum number of characters allowed in the column. Can be less but not more than the specific value.
BINARY	Used to dedicate the column for hexadecimal data.
SMALLINT	Used to store numerical data ranging from -32, 768 to 32, 767.
INTEGER	Used to store numerical data ranging from -2, 147, 483, 648 to 2, 147, 483, 647.
BIGINT	Used to store numerical data ranging from -9, 223, 372, 036, 854, 775, 808 to 9, 223, 372, 036, 854, 775, 807.

BOOLEAN	Used to store either TRUE or FALSE values.
DATE	Used to store DAY, MONTH, and YEAR in the following format: YYYY-MM-DD.
TIME	Used to store SECOND, MINUTE, and HOUR in the following format: HH:MM:SS.
TIMESTAMP	Used to store both TIME and DATE.

There are other data types supported that you can use for more complex numbers, for instance. There is also specific data type for UNICODE characters against regular characters. Now these are just a few of the flavors that you can use in database programming.

CHAPTER 4 – Coding in MySQL

I will teach you how to code SQL database using MySQL. Why MySQL? First is because it is an open-source platform and you can get it for free. Second is because it is compatible with all major operating systems: Linux, OSX, and Windows. Third is because it features ANSI mode which enforces ANSI specifications to your database. Not all RDMs have this kind of feature.

In this chapter, I will teach you how to efficiently use the SELECT command. This will allow you to effectively use this command to create interesting and efficient data queries.

SELECT Command Anatomy

You should have learned by now that we have the SQL keyword SELECT which is a command to get specific data from our database. After the SELECT keyword, what should follow is the list of items we want to have from our database. We call it the “select list”. If the select list only contains variables, then you can simply write the statement in this manner:

```
SELECT ‘[Variable]’,’[Variable]’;
```

Most of the time, the select list contains the names of the columns. When it does, it is important to define which table can we find these columns. This is where the FROM command comes in.

For instance, our select list contains User (column A) and Email (column B) and we can find these columns inside the table named as Contact. In order to define which table to look for, we should include the FROM command by doing this.

Format:

```
SELECT [COLUMN_NAME], [COLUMN_NAME] FROM [TABLE_NAME];
```

Example:

```
SELECTUser,EmailFROMContact;
```

But, what if I have a lot of columns in the table and I want to select them all, you ask? Well, this is where the power of the asterisk (*) comes in. This is known as the wildcard symbol in SQL statement. To select all columns inside a table, you need to execute a

statement in this format.

```
SELECT* FROM [TABLE_NAME];
```

As you can see, the wildcard symbol comes after the SELECT command. Now this is useful when selecting all the columns. If you do not need to select all the columns, then using the wildcard symbol may become dangerous and inefficient.

There are different ways to quickly assert multiple columns in an SQL statement depending upon the platform that you are using. I personally recommend that you avoid using the wildcard command since the inefficient use of this syntax may also break applications.

FROM Command

Selecting a single table is fairly easy to write an SQL statement. As you can see from previous examples, it only follows a simple format.

```
FROM [TABLE_NAME]
```

In FROM clause, you can shorten the name of your table by creating an alias for the table name. Let us say that your table name is “telecommunication” and you are looking for the list of owner names that can be found under the “owner” column of our table. Typing it with the use of the generic format will give you this.

```
SELECTtelecommunication.ownerFROMtelecommunication;
```

Why do we need to type the table name before the column name in our format above? This is because we must qualify the column by using the table name. Furthermore, this will save you from name collisions in the future if there will be columns with similar names. I suggest you make this a habit.

Going back to the topic, doing so is too much of inconvenience. We can shorten the table name by assigning an alias in the FROM clause. Here’s the format.

```
SELECT [TABLE_NAME_ALIAS].[COLUMN_NAME] FROM [TABLE_NAME] [TABLE_NAME_ALIAS];
```

Using the owner column and telecommunication table, our new statement will look like this.

```
SELECTt.ownerFROMtelecommunication t;
```

As you can see, I assigned the letter “t” to be the alias of the table name. This is easier to write than repeating the complete name over and over again. Keep in mind that it is not necessary to use the first character of your table name as an alias. You can assign any alias to the table name and it can also be a combination of characters. Just make sure that you are consistent with the alias you use.

Limiting The Results

Whenever you execute a SELECT query, the database will give you all the rows under the column you specified. This means that if your table has 20, 000 rows, then it will return 20, 000 values to answer your statement.

There are two ways to address this issue. One is to assert additional clause after the FROM clause. The other is by adding the DISTINCT qualifier before the select list. This shown in this format.

```
SELECT DISTINCT [SELECT LIST] FROM [TABLE_NAME];
```

This format will only give you distinct values in return instead of providing you with all the values under the specified column. This is the perfect way to have all the unique values under that column. For example, you have a table for your contacts and some entities were asserted in several rows because they own multiple phone numbers.

Relying on the generic format will give you duplicated results. This is not efficient if you only want to know who are your contacts, right? Using the DISTINCT qualifier will prevent you from receiving multiple rows for the same entity.

Here is our telecommunication table which has the column for owners and another column for the telecom company.

Key	Owner	Company
1	Jean Grey	AT&T
2	Luke Skywalker	Comcast
3	Mae Lee	AT&T
4	Stan Lee	AT&T

Let us say we want to find the list of telecom companies in our database. Using the generic format will give us three results for AT&T which is unnecessary. With the DISTINCT qualifier, however, we will get only two results – one for Comcast and another for AT&T which is what we really need.

Aside from DISTINCT qualifier, you can add the WHERE clause after the FROM clause to limit the results as well. The WHERE clause serves as a search engine with higher precision. This is done by using the following format.

```
SELECT [SELECT LIST] FROM [TABLE_NAME] WHERE [EXPRESSION];
```

To give you a simple example, let us use another table for people's first and last names. Let us name this table "person".

Key	First	Last
1	Lee	Mortis
2	Mae	Lee

3	John	Doe
4	Stan	Lee
5	Bruce	Lee

Let us say imagine that this is our database for our customers and we would like to know how many customers do we have that have Lee as their family name. In order to do that, we need to the execute the WHERE clause command.

```
SELECTp.lastFROMperson p WHEREp.last = ‘Lee’;
```

This will address the database to look specifically under the last name column and merely focus on rows with Lee value under that column. It will ignore the rows with no Lee value under that column. In other words, you will only get the rows of your customers that have Lee as their last names.

The secret in understanding the WHERE clause is by taking note of the different operators you can use for WHERE expressions. These operators are similar to mathematical operators. This means that it is easy to remember them. In contrast with mathematics, the operators we use here also works for non-numerical data such as the equal (=) sign we have used in our WHERE clause above. Here is the list of the simple operators that you can use for the WHERE clause.

Operators	Name & Description
=	Equal to – gives TRUE value if data on both sides are equal to one another.
<>	Not equal to – gives TRUE value if data on both sides are not equal to one another.
>	Greater than – gives TRUE value if data on the left side is greater than the data on the right side.
<	Less than – gives TRUE value if data on the left side is less than the data on the right side.

>=	Greater than or equal to – gives TRUE value if the data on the left is either greater than or equal to the data on the right.
<=	Less than or equal to – gives TRUE value if the data on the left is either less than or equal to the data on the right.

These are the operators that will give you Boolean values when handling WHERE clauses for your SQL queries.

Conclusion

At this point, you are now familiar how relational databases work and how efficient SQL in querying data from databases that utilize relational models.

It is important to take note of how to designate restrictions to columns and how to call them specifically using SELECT statements. You have also learned that SQL statements contain both SQL-defined keywords and user-defined keywords. These keywords are not case sensitive, but organizing and writing the statements in a visual-friendly manner is highly recommended. Doing so will not only give you convenience when viewing your SQL coding but other people as well. This is a best practice if you are working with a team.

It is also significant to qualify column names using table names when writing SELECT statements. This will prevent future issues when there are different tables with similar column names. You just learned how to shorten the table names using an alias.

You can also specify data extraction of the SELECT statement by using clauses such as FROM. Using the FROM clause allows you to specify which table to look at whenever your SQL statement looks for columns. Aside from that, FROM clause also allows you to assign an alias for the table name.

You also learned how to limit the number of answered values with the use of DISTINCT qualifier and the WHERE clause. Keep in mind that you have to avoid using the wildcard symbol since it is an inefficient way to extract data from your database. The WHERE clause, on the other hand, is a powerful command that you need to master as a beginner. Familiarize the different operators and try them out on an already built database.

I am aware that this book is basic, but it is enough to prepare you in moving forward to advanced studies in the structured query language. As a beginner, there is no need for you to rush immediately on how to create your own database. It is important to learn which database model does SQL functions first. By understanding the relational model, you will easily grasp the concept of SQL together with the statements that you can do using this language.

If you want to continue learning about SQL, then I suggest that you take advantage of the book’s sequel where you will be introduced to deeper clauses of SELECT statements. The second part also tackles about JOINS, handling data, and table creation.

Thank you for reading. I hope you enjoy it. Ask you to leave your honest feedback.