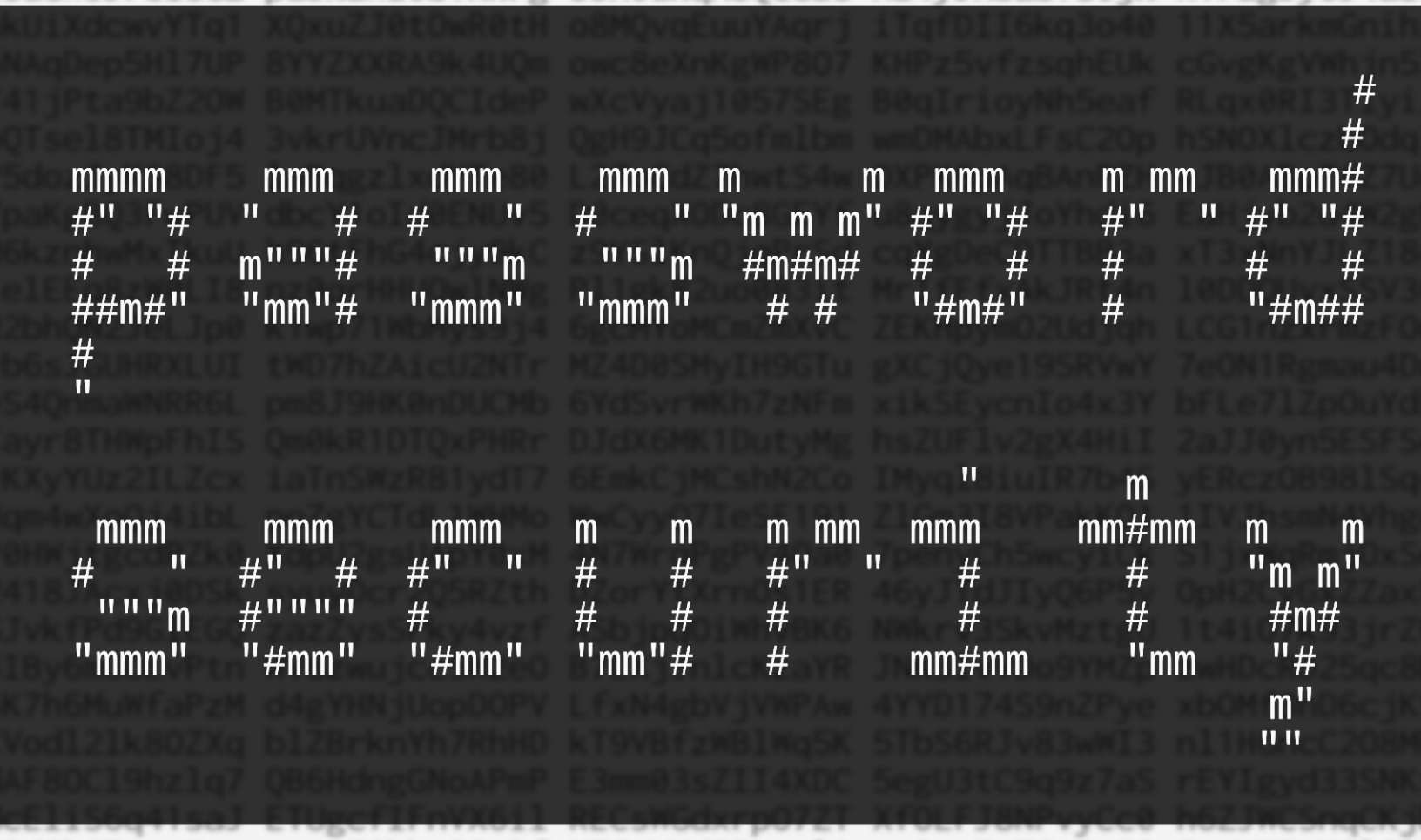
Every Sentient Being's Guide to



by Chris Spackman

This page intentionally left blank

Every Sentient Being's Guide* to Password Security

Chris Spackman

May 2021

Contents

List of Tables

Chapter 1

Introduction & Quick Overview

Preface

Let me begin by explaining the * in the title. Obviously, this book is not actually *everyone's* guide to password security. Many people who do not use computers will find nothing useful in this book. Likewise, most infants would probably not use this book for anything like it's intended use. People who cannot read English would also not get much from this book. You get the idea.

My intended meaning of "Every Sentient Being" is that this guide is accessible to anyone with an interest in the subject (*password security* in the current case). You do not need any special training or experience to understand and use the information here. Also, I do not want to insult my readers by calling them "stupid", "idiots", or "dummies". Uninformed on this topic they might (might!) be, but reading this book should help with that. "The Uninformed Person's Guide to Password Security" might have been acceptable, but it still focuses too much on the negative. "Every Sentient Being's Guide" is the best way, I think, to make it clear to everyone that this book will be accessible and useful.

If you have comments, suggestions, corrections, updates, etc. please contact me at chris@ChrisSpackman.com.

Who This Book Is For

This book is for anyone who is interesting in learning to make strong passwords and protecting themselves online, but who maybe is not a tech enthusiast.

Quick Overview

This book is about passwords; what they are, how they work, and especially, how to make and remember strong passwords. Despite the noise occasionally made about newer, supposedly better, ways of authenticating a user on a computer (or web site, or wherever), passwords will probably continue to be the most common way for most users to log in, and to protect their data, for many years. So a bit of knowledge about passwords should be useful to you for many years to come.

First, there is a chapter for people who just want to quickly and easily learn what to do to have better passwords online. I give some things to do, but very little explanation of why those things are important. That is what the rest of the book is about. Even if you plan on reading this book "cover-to-cover", you might want to first read Chapter 2 on page 11 and make any changes to your own current password practices before then reading the rest of the book.

After Chapter 2, "Too Long; Don't Want to Read", we will briefly look at some of the terms that you should have a basic understanding of before getting into too much detail. Do not worry if you do not fully understand the meanings of the words in the vocabulary section. Actually, I would be surprised if you got a full understanding of any unfamiliar words from just the vocabulary section. My hope is that after you have read through that section, you will be better able to understand the content when we get to it in the other parts of the book.

We will also briefly talk about some of those "other systems" and why they are not always better than, or good replacements for, passwords —

in other words, why understanding strong passwords is still useful in your daily life.

Before getting too deep into what passwords are, how they work, how we can use them securely, and the such, we will look at some concrete ways to make strong passwords. For many readers, Chapters 2 and 5 may be the only ones they read, which is fine, of course. If you are in a hurry, or do not really care all *that* much about the other stuff, and just want to know how to make strong passwords **right now**, skip to chapter 5, on page 25. (Of course, if you don't even care about how to make strong passwords, but just want to know what to do to have more secure passwords, look at Chapter 2 on page 11.

The rest of the book after Chapter 5 will go into more detail into how passwords work and how we can use them securely — making a strong password is good, but only one part of the battle for digital security. A strong password will not help you, for example, if you cannot remember it, or if, in order to "remember" it, you end up writing it down on a piece of paper taped to your computer screen.

Too Many Passwords

Unfortunately, many people *do* end up writing passwords down and taping them to their screens, or hiding them under their keyboards. We need to remember passwords for the many web sites we visit and usually also the computers we log in to. Ideally, all those passwords should be different, but realistically, that many different passwords is too much for a normal human to remember. To deal with this, people tend to use simple passwords and to *reuse* passwords (that is, use the same password on several sites). Both are understandable compromises, but both are bad security.

Personally, I have to remember log in information for five different computers, several banking, mortgage, and other financial sites, a handful of music and art sites, web sites related to my day job, forums that I am a member of, social media sites, and so on and so on. You probably have a similar list of sites you need passwords for.

To keep track of all my passwords and site log in information, I use several tools, including *password manager* software (more on that later). Right now, my main password wallet has over 100 entries, almost all with user names, passwords, and security questions, plus the answers to those questions. Imagine trying to remember all of that just in your head! No

wonder so many people decide to use weak passwords like `password123`! I am sure I would use something similar if I did not know about password manager software. How else is the human brain going to keep that many isolated, basically meaningless phrases in memory? (Answer: it probably will not be able to.)

So, it really is no surprise that many people do not make the effort to use better passwords. Nor is it surprising that they might use the same password (strong or otherwise) for several web sites. Even for people who understand why passwords are important and what makes a password strong can feel overwhelmed by the number of passwords they need to deal with.

Don't worry, though. One of the goals of this book is to show you several methods for creating strong passwords that you can still remember, if you need to, and that the bad guys cannot easily guess.

After you have finished this book, you will know why simple passwords and password reuse are bad security, and you will know several ways to practice good security. If all goes well, you will be able to:

- explain to your friends and family, in general terms, what passwords are, how they work, and why they are important. Whether your friends and family will care is another matter.
- decide how strong a password needs to be in any given situation.
- create strong passwords that are appropriate to the type of login.
- use software to securely store passwords, and other security data, for your computer logins, financial web sites, and other accounts.

A Little About Computer Security

Passwords are one part of computer security. Security is ***always*** a trade-off between security and other factors such as convenience and costs. Let me say that again:

Security is ***always*** a trade-off.

Physical safes, for example, are rated by how long it takes a professional safe cracker to break into them, or how long a safe will protect the contents from a fire or other disaster. There is no such thing as a "perfect" safe that will keep all professionals out or will protect the contents from all possible disasters. Obviously, you can spend more money to get a better

safe, and some very expensive safes will protect their contents from *almost* everything. However, those safes have huge price tags. For a bank, that extra security might be worth the cost. For Joe User at home, his comic book collection is probably not worth the expense and inconvenience of the best safes.

The situation is similar with computer security. Although there are "almost unbreakable" passwords, such passwords are very long and impossible for most people to remember. Another trade-off is that some web sites may not allow a password long enough to be "almost unbreakable". (In this case, the web site made the trade-off, not you, but you are affected by their choice.)

I use the phrase "almost unbreakable" for two reasons. First, because even though a strong password is unbreakable in theory, in reality, a bad guy could always get lucky and recover a strong password in much less time than it "should" have taken. Second, a strong password can be bypassed in many ways, some not under our (the user's) control. If the bad guy can get access to your account, the effect is the same as if the password had been broken. Basically, I don't want to say "unbreakable" and have people think that a strong password will automagically protect them from everything.

Even if you use an "almost unbreakable" password, the bad guys can break it by luck or, much more likely, by getting the password some other way — for example, by tricking you into giving it to them (this is called *social engineering*), or by capturing what you type when you visit the site in question (using *keyloggers* or *Person-In-The-Middle (PITM) attacks* — more on those later). So, a strong password is just one part of the security equation.

All else being equal, strong passwords are still much, much better than weak ones. If nothing else, strong passwords take longer for the bad guys to break — perhaps giving you time to change your password before the bad guy can break, and then use, your soon-to-be-previous password.

"But", you might ask, "if the bad guys can break even a strong password, why should I inconvenience myself by using one?" Fair question. Think about the deterrent effect of a home alarm system — just having it (or even just putting a sign in your yard saying that you have an alarm) might make a thief go to another, less well-protected, house.

How would this help in the digital world? Pretend, for example, that the bad guys attacked your bank and got the database of the users' en-

crypted passwords. They don't have your password, but they have the encrypted version (the *hash* — more on them later). The bad guys have to somehow convert that hash into your actual password before they can log in as you and send all your money to some country with loose banking regulations.

The bad guys are going to do some simple and quick attacks on all of the passwords hashes they stole. After they *recover* (aka *find* or *break*, meaning they now know the password) the easy passwords, they will probably start using them — after all, eventually the bank will realize that someone has been in their database. Better for the bad guys to start using the passwords that they can get easily, as soon as they can. As a bonus for them, someone who uses `password123` as their on-line *banking* password is probably using that same password on other sites as well.

So now, if I were a bad guy, I would try that user name and password on as many banking and financial sites as I can. If Joe User really did use the same password on several sites, not only can the bad guys take all of his money from the bank they originally hacked, but they have a good chance of finding a few more of his financial accounts and stealing all of that money also. Poor Joe User can say good-bye to *all* of his money.

Of course (and this is where strong passwords enter the picture), once the bad guys start draining people's accounts, someone will notice and law enforcement will get involved. The bank that was attacked and had their database stolen will warn all of their users, and possibly even reset all passwords, just to be safe. At this point, any passwords that the bad guys have not already recovered and used are worthless to them. So, if your strong password can withstand their attempts to break it long enough, you might have time to find out and change your password. Joe User, with his weak password, is probably not so lucky.

Think of a strong password as one of those better safes — it keeps the bad guys out long enough for the police to get there and stop the bad guys from getting away with anything from the safe. Using strong passwords is like protecting your accounts with a bank vault. Weak passwords are like protecting your accounts with a cardboard box.

Chapter 2

Too Long; Don't Want to Read

If you don't care about understanding passwords and just want to know how to have strong passwords and be safe(r) on the Internet, here are the steps you should start with:

1. Use a password manager. See chapter 9 on page 56

2. Use the password manager to create passwords as long and as strong as each web site will allow. Ideally, use passwords that are at least 20 characters long. Longer is stronger. Mix of character types (upper case, lower case, special characters) is stronger.

3. Use Two-Factor Authentication (2FA) wherever you can. 2FA is also sometimes known as "Multi-Factor Authentication" or MFA.

    - SMS messages (text messages sent to your phone) are not secure, but are better than no 2FA.
    - Authenticator apps (such as from LastPass or Google) are more secure than SMS messages because they are only susceptible to PITM (Person-In-The-Middle) attacks at setup. Every individual SMS message is, in theory, susceptible to PITM attack.
    - Physical cards / keys, such as from Google and Yubico (the YubiKey), are similar to authenticator apps — very strong, and bonus for depending on *something you have* and not just *something you know*.

- BUT, both SMS and authenticator apps are susceptible to *social engineering*. **NEVER** believe someone who asks for your SMS code or authenticator number outside of the actual web site you are trying to log in to. Bad guys will try to get the code from you so they can log in as you from somewhere else. (I've not heard of a way for bad guys to trick you when you use a YubiKey or similar, but we shouldn't assume it is impossible. At best, it just isn't widely attacked right now. That may change.

4. For machines that you have to log into without access to your password manager (like, pretty much every machine log in), use a unique ***passphrase***. A passphrase is something like: *Use The Force Luke* or *Correct Horse Battery Staple*. Not passwords, but pass*phrases*. These are not as random as truly secure passwords, but they are long enough to be secure enough, and are easy for humans to remember.

   You could use these for web sites, but a password manager with actually random, secure passwords is better there. Passphrases could be useful for software that you log into locally (not on the web), though, just because they are easier to remember and usually easier to type.

   - BUT, don't use well-know phrases (both of the examples above are bad passphrases! — search for "XKCD correct horse battery staple" (without quotes) if you aren't familiar with the second reference) and don't re-use your passphrases. Diceware (http://world.std.com/~reinhold/diceware.html) is a great off-line way of creating a secure passphrase.

5. ***It is okay to write down your passwords and passphrases.*** This may sounds surprising, but there are rules. Mostly this applies to machines log ins (where you don't have access to your password manager. But, if you need to for some (good!) reason, you could write down a regular password or passphrase. But, you should follow these rules:

   - Put the paper someplace safe. Don't leave it under the keyboard or on the monitor. Your wallet is a good place. So is an actual safe. If someone steals your wallet, they probably don't have access to your computer, so no worries.

- If you are keeping it NOT in a safe, don't include information about user name or what machine or which web site the password is for. A password by itself could be to any of hundreds of web sites. If your paper says `gmail: username@gmail.com ; Ahb9sohd`, then even the dumbest thief is going to try that username and password combo on Gmail. If it just says `Ahb9sohd`, they probably aren't going to waste any time trying to figure it out. Not when they could be buying things with your credit cards instead.

6. Also, if it isn't obvious, use a strong (long & unique) passphrase for your password manager. It is also okay to use a secure password (long & random) for your password manager and write that password down — the rules above still apply.

    - You will be able to remember and type in even the most difficult password after several times. Use the paper until you have the password memorized, then put the paper someplace safer (it should always be someplace safe) as a backup.

7. Do ***NOT*** lose your password manager password or it is game over. Don't use any password manager that has a way to access your wallet without the password. A password hint — sure, but probably only helpful for a meaningful passphrase — but otherwise, alternate ways in is just another way for the bad guys to get in. The exception is "one time passwords" that you can set up ahead of time. You are expected to print them out and keep them safe. These are legit methods that your password manager may provide. "Password reset" for a password manager is ***NOT*** a legit method — it means that the password manager company can access all your data!

8. If you have family / anyone who will need access to your accounts if you die or are incapacitated, you should definitely either use an escrow service (LassPass and other password managers include emergency access for paid accounts) or keep a printed / handwritten copy of at least the password manager password in a safe place ***and make sure the appropriate people know how to access it***.

9. Don't use actual information (i.e. correct answers) for security questions. If a web site wants to use security questions as backup for forgotten passwords or as a sort of 2FA, that is fine — just make up random data to use as the "answers" to the questions.

    - Favorite food? `aDui8thi`.
    - Mother's maiden name? `Er9Ona1N`.
    - As much as possible, avoid any site that actually checks, or requires meaningful information. I once made an account at a site that asked for an old phone number as a "security" question. I put in gibberish; they complained that it wasn't a 10-digit phone number. That is stupid security (or perhaps they are doing something else with the data?). They probably have many users whose previous phone number is 867–5309.

10. Finally, if you have work computers or log in to several different computers at home and away, consider having a local password manager wallet for financial, medical, and other sensitive log ins, and an online wallet for day-to-day web usage. Of course, ***ALL*** your password manager wallets should have different, strong, unique, passwords.

    Many people are hesitant to trust an online password manager (1Pass, LastPass, BitWarden, etc.) with their banking or other sensitive / valuable login information. Fair enough — as Rachel Tobac says, we should be "politely paranoid". So, not trusting even our password manager providers is totally acceptable. Other people aren't worried, usually saying they have to trust the provider anyhow, so why draw an arbitrary line? Whichever side you come down on, or wherever you draw your line, it can be a good idea to have a local password manager as well as an online one. (If you work from home most of the time, and rarely travel, you may not even need an online manager.) KeePassXC is a good cross-platform local password manager. You should do what works best for you, your risk tolerance, and your situation.

That is it. There is more of course — thus the rest of this book — but if you do the above steps, you should be a lot more secure online. If nothing else, please use a password manager and secure passwords and please do ***NOT*** re-use passwords.

Chapter 3

Vocabulary

One best practice in education is "pre-teaching" the vocabulary that students will need to understand the content that they will learn. Pre-teaching just means that the teacher should introduce and explain important vocabulary words *before* students encounter those words in the content that they are learning. For example, an English teacher might be sure that all the students understand what "theme" and "setting" are before talking about the theme and setting of the book they are reading. A math teacher will be sure that students know all the relevant words that refer to "addition" in word problems ("plus", "added to", "more than", "sum", etc.) before doing those word problems.

This chapter briefly pre-teaches the vocabulary we will need. As I mentioned earlier, please don't think that you have to totally understand all the vocabulary we look at in this chapter.

You will have an easier time understanding both the words and the content when you encounter these vocabulary words again later in the book.

All of the explanations are relevant to our purposes in this book. There may be other definitions, but if they don't relate to passwords or computer security, we won't worry about them.

biometrics — the use of some biological information about a person (your fingerprint or your face, for example) to log that person into a computer or device. Because it is biology, "close" matches might work. Biometrics are not as clearly "matches / doesn't match" as passwords. In theory, at least, this leaves room for attacks.

brute force — an attack that tries every possible password until it finds the correct one. Brute force attacks are the base-line; to be taken seriously, other attacks must be faster than brute force.

character — any printable letter, number, symbol, or even white (blank) space, usually just the ones on your keyboard, but depending on the system and the language, "characters" may include letters from other alphabets or Asian characters.

dictionary — a collection of words, phrases, etc. that people can use to check the security of passwords or to attack passwords. They do not have any definitions, just words or phrases. They do not have to be "just" regular dictionaries. Security "dictionaries" may contain all the words from several regular dictionaries from several languages plus known common passwords. Most likely, the dictionary is just a regular text file on a computer. It could also be a database; for example, if a web site wanted to check the passwords of new users, it might have a dictionary database to check against. In theory, the site would then reject weak or already well-known passwords.

dictionary attack — an attack that uses words or phrases from a dictionary to try to find a password or log into a site. It is similar to a brute force attack, but against most people's passwords the dictionary speeds up the process because the attack will start with the most likely passwords. Even "strong" passwords can be recovered quickly if they are made with well-known patterns, like substituting numbers for letters, as with `Pa$$w0rdl2E`.

facial recognition — a type of biometric that uses a camera or cameras to confirm the identity of the user by comparing information about the face with information previously stored about the user's face. More advanced facial recognition uses more than just what we think of as a picture; it could also use infrared and 3D information, for example. Cheap facial recognition can be fooled by pictures. Some current systems can be fooled by twins.

hash — an alphanumeric string (a bunch of letters and numbers) that is mathematically unique to the text or file used to generate it. In other words, hashing software does complex math on a password and gives you the hash. Hashes have several very important features:

- The hash should be unique to the password. (If it is not unique, that is called a "collision" but it is *very* rare and we don't need to worry about hash collisions here.)
- The hash **cannot** be used to find the password. Hashes are "one way". Password ⇒ hash is easy (for a computer). Hash ⇒ password is very, very, very (like close to impossible) hard.
- The hashes gives no information about the password. If just one character is different in the password, the hash should be much different. This matters because it means that the bad guys cannot use a hash result to see if they are getting close to finding your password. For example:[1]
  - The hash for "password" (without the quotes) is: `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8`
  - The hash for "Password" (without the quotes) is: `8be3c943b1609fffbfc51aad666d0a04adf83c9d`

  Notice how the one changed character — "p" into "P" — resulted in a very different hash. This is both normal and good.

keylogger — software or hardware that literally "logs" (keeps a record of) everything you type on your keyboard. The goal, of course, is to find your user names and passwords in that log.

malware — basically "bad software". General name for any bad or malicious software. Malware includes, for example, viruses, trojans (as in the Trojan Horse), worms, ransomware, and many others.

passphrase — a password that is a longer phrase or sentence, usually in regular English (or other language). Passphrases are memorable, so they get their strength from length rather than from randomness.

password — a string of characters used to authenticate a user on a computer system.

person-in-the-middle attack (PITM) — an attack that involves someone literally being between you and whoever you are communicating

[1] I made these hashes with the very old SHA-1 hash function. I use it only because hashes from newer functions are longer than a line of text.

with. By listening to the communication, or worse, being involved in setting up the security of the communication, the bad guys can get information they want — such as, for example, your credit card information or the password to a web site. (This used to be known as, and you will still sometimes see, "man-in-the-middle" attack.)

recover — there are several ways to talk about finding out passwords, including "attack", "break", "find", "recover", and others. These usually refer to trying to find a password when you have the hash or similar information about the password. *Social engineering* tries to get the password from you, in ways that don't involve recovering or attacking it.

salt — random characters added to a password before hashing. Salt makes each password unique and also increases the strength of a password. Normal users don't have any control over salting. The web sites that store your passwords either use salt or they don't. All should use it.

- For example, the (SHA-1) hash for the salt "abcd" added to "password" (without any quotes) is:
  `87cecca7930c300285188bcd123eca6618920750`
- The (SHA-1) hash for the salt "abcd" added to "Password" (without any quotes) is:
  `6ecac58a2306787e89df14a5484682a872e76de3`

Again, these two are very different from each other, and they are also very different from the hashes of the original passwords without the salt.[2]

social engineering — trying to get information or access by tricking people. This is often the easier way to get a password because it avoids the technological defenses (like strong, salted passwords). "Phishing" emails that tell you that you need to log in and change your password are a common social engineering technique. Of course, the link in the email leads to their (fake) site, not the real site you think you are going to. "Phishing" is just one social engineering technique, but it is probably the most common.

[2]If you are a serious geek and okay using the command line, you can make strong passwords just by hashing two words together like this.

Chapter 4

Other Methods of Logging In

Before we talk more about strong passwords and how to make them, lets look a little at some other ways of protecting access to your devices and accounts. An alternative that is popular (especially on smart phones) is the use of biometrics for logging in.

Biometrics

Biometrics is the use of some part of you — your face, your eye, your fingerprint, etc — as a way of logging into your computer, other device, or web site. Are biometrics really not as "good" as passwords? Will they replace passwords soon? If not, why not? Are they not as secure as passwords? More secure? What drawbacks do they have?

Good questions. In MS Windows 10, Microsoft introduced to the general public a new way of logging into your computer. Basically, the computer looks at your face and if it is you, it logs you in. Microsoft calls this feature "Hello". This is known as *facial recognition* and you may have seen it used in TV shows and movies, usually to find a suspect in a crowd of people. Apple introduced a similar system in their iPhone X.

Another, older, biometric system is the fingerprint scanner, which uses your fingerprint to unlock your phone. The technology behind both facial recognition and fingerprint scanners has been around for a while and these days it works pretty well, but not perfectly. That is, these systems usually let you in when you use them and they usually do not let others in, when someone tries to access your device.

Aside from the fact that they only "usually" work, my biggest issue with biometrics is that the systems in use today (by regular people) have not been around long enough for us to understand their security well enough to trust. Passwords in computer systems have been in use long enough that security researchers know a lot about how the software that uses passwords works and what the common ways to attack or weaken password systems are. This is not true for general purpose biometric systems today. The obvious weaknesses are pretty obvious: Will a picture of me unlock my account? (That has been done.) Can the fingerprint scanner tell the difference between a plastic duplicate and my real finger? (Some can be fooled.) What about my real finger, but cut off of my hand — that is, stolen from me? (That can work, and has been done.)

Those are just a few of the most obvious examples, and to be fair, companies are making progress on those and similar issues. Facial recognition systems are getting more and more complicated, meaning that fooling them is getting more and more difficult. Microsoft, Apple, and other consumer-oriented companies are trying to make the technology more secure against some types of attacks, which is, of course, a good thing.

However, this consumer technology has not been around long enough for researchers to have found and dealt with all of the possible weaknesses. Pictures and fake fingerprints are just the most obvious ones — elementary school level attacks that almost anyone can try. What about the software that takes the picture that logs you in? The software that stores the data, to compare with the picture? The software that does the comparing? Those are all (mostly) still security unknowns. As far as I can find, there has been very little research done on the security of these parts of those systems. (I think it is reasonable to believe that internally Microsoft, Apple, and Google are probably doing some research in these areas.) There might already be several problems with the systems that no one has found yet.

One final issue with biometrics is that you cannot change them. To my knowledge, this has not yet been a problem. However, in theory if the biometric information that is stored (the biometric "hash") about you were leaked or was stolen, you would have no way to change it. Best case, you would just have to stop using that form of biometric to log in. One good thing about passwords is that they are not tied to you — you can change them whenever you like and the previous one is immediately worthless.

To be honest, I'm not sure that the above is as big an issue as many people think, but it is also not an issue that we should just ignore. There are many ways I can input a password even if I break both of my hands, or lose an arm, or some other tragedy affects me. How many ways are there to input my face if I injure it in an accident? How can I input my fingerprint if I lose my hand? Again, not something most people need to worry about, but these things already have solutions for passwords. The solution for most of the non-password systems? Falling back on passwords! No fingerprint? Put in your password. Facial recognition not working? Put in your password. So, explain to me again how these systems replace passwords (and don't just add a layer on top of them)?

Password systems are not perfect, of course, but the biggest issue is the actual passwords that people use (which is why we are talking about password security here). The system itself has been pretty well tested over the years. The software that takes care of accepting, checking, and updating passwords is about as secure as professional security people can make it. Logging in with passwords — whether in to your computer or a web site — is a well-understood, well-researched problem with (hopefully) very few surprises left. The same is not true for biometric systems. If someone gets your stored biometric information, will they be able to use it to impersonate you? We don't know yet.

Another reason for not using biometrics is legal and may seem to some people to be approaching "tinfoil hat" levels of paranoia. Now, please understand that I am not a lawyer and this is **not** legal advice. Having said that, my understanding is that, legally, in the U.S.A., you cannot be forced to tell the police your password. Passwords are "things you know". Thanks to the 5th Amendment, the government cannot require you to tell them "things you know".[1]

The same is not true of "things you have". The government can require you to turn over keys, usb drives, phones, and any other thing that they can convince a judge they need to make a case against you. This probably includes biometrics. How hard would it be for a police officer to just hold your phone up in front of you, and log you in against your will? My under-

[1] Although, in December 2016, a state supreme court ruled that a suspect could be required to tell police the password to his phone, so that they could unlock it and search it for evidence. This ruling, however, goes against a USA Supreme Court ruling that specifically said suspects could not be required to do this. So, until the laws get sorted out, you may be in a gray area.

standing (again, I am not a lawyer) is that it would be totally legal for an officer to do that. Worst case, the officer might have to get a warrant first. Certainly, the 5th Amendment would **not** protect you.

Once the police unlocked your device, searching it is no different than searching your briefcase, purse, backpack, or whatever. Actually, there is one difference: the police can make a perfect copy of everything on your device, keep that copy forever, and perhaps even share your data with others.

Digital law is still not entirely decided, and laws could change — and I'm only talking about the U.S.A. here. Laws are no doubt different if you are not in the U.S.A. In the meantime, if you have evidence of criminal behavior on your devices, or if you just like your privacy, you might prefer to use passwords and not biometrics. Or, at least talk with an actual lawyer before deciding on biometrics. Of course, please understand that governments have many other ways to get into your device, even if you use a password. So, don't think that a password is going to prevent access. It may just prevent you being forced to unlock your phone. Remember how I am not a lawyer, and this is not legal advice? Good.

A perhaps more realistic example of why you might not want to use biometrics (or why you might want to also understand strong passwords) is allowing someone else to log in as you. What if you forgot that critical document on your main work computer and you need a coworker to log in and email it to you? Or you are washing dishes when a friend calls and you need your significant other to answer your phone for you? If you are only using facial recognition on your computer or fingerprint authentication on your phone, you may be out of luck because they will not be able to log in as you. You may even have a password set up as an alternate log in method, for just these sorts of situations. In which case, everything we are talking about in this book is still relevant.

Finally, biometrics would be overkill for most web sites because it would require every computer or device you might use to have that software and hardware. Cameras might work for facial recognition or maybe even fingerprint recognition, but not every computer has a camera that would work for this and many just do not have a camera at all. Fingerprint scanners would be better, but even fewer (non-smartphone) devices have those. What happens when you want to log in but the computer or device you are using does not support the biometric you are using? You might just be out of luck. My guess, though, is that there will be a pass-

word fallback for those situations. So, a strong password would still be very important. Even if biometric or other methods work well and become popular, passwords will still be with us for a long time.

SQRL

Mr. Steve Gibson created a new way to log into web sites. His system is called *SQRL* (pronounced "squirrel"). SQRL stands for "Secure, Quick, Reliable, Login". SQRL was officially released in 2019. SQRL fixes the biggest issue with logging into web sites — the fact that the web site has to keep a database of the username and (hopefully) **hashed** passwords. Every time you log in, the web site re-hashes the password you sent, checks the hash against the hash for your username in the database, and if it is you, the site lets you in. But, that database is one of the first thing hackers often take when they break into a site. As we talked about earlier, the hackers can take the hashes and try to recover the passwords. Then, with the user names, they can try to log in to that site and any others with the recovered passwords.

SQRL (as I understand it) does not depend on any web site to keep any information. Because the web site does not keep any log in information, there is no log in information for hackers to steal. This makes you safer because it is one less way for hackers to try to get your password. (As I understand it, the SQRL private log in information is basically on **your** computer, so anything the web site loses control of is the public part, so it doesn't matter to your security if it gets stolen. If I am understanding SQRL correctly, that could mean that **your computer** could become a more valuable hacking target. But, probably not, because if the bad guys are already into your computer, they already have access to your passwords (via the keyboard and keyboard loggers), so SQRL isn't really the issue at that point.)

SQRL takes care of securely logging you into any sites set up to use SQRL. You don't need to use a user name and password. But, guess how you prove to SQRL that you are really you? That is right — you use a password! You still need a strong password to protect your SQRL identity. Your SQRL identity is stored on your computer, and protected locally, so in theory at least, it is still safer than using a password with a web site. In reality, almost everything I mentioned for biometrics it true for SQRL also.

That is, it is a new system and security researchers have not had time to take it apart and find any weaknesses or novel ways to attack it.

As I mentioned above, one worry I have with SQRL is that it may make individual users more tempting targets than they are now. Now, hackers are more likely to go after big web sites, to try to get the user name and password (and other) information. This is just because that is where the information is. But, if the web sites don't have that info anymore — if with SQRL, it is entirely on the users' computers, then maybe the hackers will decide to go after the users instead. I think it is unlikely, but in theory, if a hacker can get your SQRL identity, it sounds like he will be able to log in as you to any site where you use SQRL. Again, though, we just have to wait and see — new systems may be better, but they have to prove themselves first. (And I freely admit I may be misunderstanding the details of how SQRL works in this respect.)

In the end though, because systems that use biometrics (and also SQRL) also need passwords, we still need strong passwords.

Let's make some.

Chapter 5

Quick Start

The goal of this chapter is to show you several methods for creating strong passwords, some of them for passwords that you can remember — and that the bad guys cannot easily guess — and some for passwords that you probably wouldn't be able to remember. If all goes well, after reading this chapter, you will be able to:

- create strong passwords that are appropriate to the type of login.
- use software to securely store passwords for your computer, web, and other accounts.

What is a Strong Password?

A strong password:

- is hard for a human to guess (even if the bad guys know you)
- is not in any dictionaries (even of other languages)
- is not in lists of passwords
- **hashes** to something that is not in any **hash** dictionaries
- has lots of **randomness** (meaning, basically, that knowing the first couple of characters will not help the bad guy figure out the next few characters).

- is long enough to withstand a contemporary **brute force** attack. Four- or 6-character passwords are right out. Use at least a 10-character password. For future-proofing, use a 20-to-30-character password.

- includes **characters** of all types — upper, lower, numbers, special characters, and white space if allowed

Basically, the more unusual, the less like a word from real human language, and the more random a password, the stronger it is.

Okay, so how do we make one of those?

Making Strong Passwords

There are 4 basic ways to make a strong password:

1. software,
2. paper methods: charts, or dice and a word list
3. a system to change something else into a strong password,
4. a pass**phrase** instead of a pass**word**.

Software has the advantage of being fast, easy, and ***much, much better***. For most people, for use on most web sites, I highly recommend using software to generate a long, random password. Create a password that is as long as the web site will accept, up to about 30 characters. Then, use a password manager to save those long, random passwords. I say that they are much, much better because they are. Humans are horrible at creating "random" numbers and strings of letters. You might think it is random, but if you thought it up in your head, it probably isn't. Computers are not perfect (and there are lots of places the software programmers could have screwed up), but in general the state of the art is much, much better than you at creating "random" passwords.

Stop. Please read the above paragraph one more time. Really, seriously, most people can stop reading right now (assuming you also have already read Chapter 2). Install LastPass, 1Pass, Bitwarden, KeepassXC, PasswordSafe, or other password manager, and use that to create and store your super-strong passwords. See Chapter 9 on page 56 for more information on password managers. Actually, you should still read the rest of this

chapter anyway, because you will want to use strong passwords for your computer login and for your password manager. Obviously, most people should not rely on a password manager to remember a computer login password, because they will need to be logged into the computer before they can open the password manager. Also, you need a strong password to protect access to the password manager.

What do you do to come up with a strong password to protect your log ins and your password wallet, without being able to just save a complex, strong password in the wallet? You have several options:

Charts are not random, but make it easy to carry your password system with you, while not having to worry about someone stealing your passwords.

Dice and a word list are very likely to be random, and they should give us a fairly secure password. But, it takes more time and obviously requires dice and a word list.

Other pen-and-paper type systems transform something into a password. We can remember something easy and then recreate the password when we need it. So, we can get strong passwords that we don't have to remember. The downside is that we have to recreate it (using our system) every time we need the password. However, it still might be a good system to use if you have to log in to different computers, or to web sites from different computers, and will not always have access to your password manager.

Pass*phrases* are just very long pass*words* made out of real English (or other language, or a mix of languages) sentences. They get their security by being very long, but because they are meaningful, they are easy to remember. However, you need to make them unique, because the bad guys will try famous sayings, passages, proverbs, song lyrics, and the such.

Strong Passwords: Software

Software can easily create strong passwords for you. I am not going to recommend any specific software here because what is good as I write this may not be good or even available when you read it. If you search your

app store or the web for "password generation software" (with the quotes) plus the name of your operating system (Apple, MS Windows, Linux, Android, iOS, etc.), you should find several. Be careful to only download from sources that you trust, and likewise only put data in (or use data from) web sites that you trust. Actually, I would never use a password I got from a web site, even if I trusted the web site, just because it is on the Internet. I prefer creating passwords and passphrases locally. Your risk tolerance may be different.

After you find some software, I suggest searching Wikipedia for the name of the company or group offering the software, or search for the software itself. Wikipedia can give you enough information so you can decide if the software is secure or acceptable for your needs. Wikipedia also usually has pages comparing different sorts of software. For example, take a look at Wikipedia's List of Password Managers. That page is about password managers, but it is still a good place to start because most password managers can make strong passwords for you. Here are some things to consider when looking for good password software.

Such software should:

- install on your computer or device ("local") — avoid password generators on web sites
- have options for as many types of characters as possible (upper-case, lower-case, numbers, special characters, and white space)
- have options to require at least X number of each character type
- tell you how strong it is (usually in bits of randomness — for more important sites, you want at least 80 bits (about 14 random characters) for a "strong" password)
- Bonus: have an option for "entropy collection" to increase randomness. This is not a requirement, but if the software has this option, you should use it. "Entropy collection" could involve you moving the mouse at random or typing randomly on the keyboard (or both) until the software has enough random data.

Of course, once you have a 25-character, random password, you will probably want to use a password manager so that you don't have to remember it. Writing it down on actual paper may also be an option.

Once again, let me state that for most people, using good password generation software and a password manager is probably the best option. Just make sure that you do not forget the password to your password manager!

Strong Passwords: Paper Methods

As long as you use a good system, pen-and-paper password generation is actually one of the best ways to make secure passwords. The reason is simple: it is basically impossible for a bad guy to influence the end result — your new password.

Every now and again, security researchers discover a flaw in the software that makes random numbers for security software. Sometimes the flaw looks like an unintentional mistake by the people who make the software. Other times, the flaw looks like something a bad guy would want to put into security software, to make it weaker. When that happens — especially with closed-source, proprietary software — it is better to assume that it was the bad guys that put it there. They would put the error there to weaken the system and thus they would be able to more easily break the passwords or other encryption that used this system. Paper-and-pencil methods are immune to this sort of attack.

This section discusses two main paper methods for making secure passwords. The first relies on a card or chart, from which you will create your password. The second uses dice to create truly random passwords or passphrases from a list of characters and words.

Cards and Charts

One type of paper system that might do what you need is the PasswordCard[1] and other similar chart-based systems. The PasswordCard is a handy wallet-sized card with a chart of numbers and letters. By remembering colors and symbols, you can recreate strong passwords whenever you need them. You decide on your own system for reading off the card, so if someone gets yours, they do not automatically get all your passwords.

A similar system is to have a chart with numbers, letters, and symbols, with letters and numbers around the outsides. Using a keyword, you look up the letter or letters in the chart for each combination of letters in the

[1]PasswordCard is at: https://www.passwordcard.org/en

keyword. A very brief version, that changes a key number into a password, is shown in table 5.1.

	1	2	3	4	5	6	7	8	9	0
1	a	b	c	d	e	f	g	h	i	j
2	k	l	m	n	o	p	q	r	s	t
3	u	v	w	x	y	z	0	1	2	3
4	4	5	6	7	8	9	A	B	C	D
5	E	F	G	H	I	J	K	L	M	O
6	P	Q	R	S	T	U	V	W	X	Y
7	Z	!	]	{	;	:	’	"	<	>
8	@	#	$	%	^	& *	(	)	=	?
9	[	{	_	-	.	+	aa	ae	ax	ch
0	dg	th	xx	ct	il	yy	zz	gt	z!	@(

Table 5.1: Example Chart for Making Passwords

That table is just an example. A better one would have more columns and rows — perhaps 26 or even 36 each way — and could hold many more characters, including more combinations of letters (like we see in the last two rows in the table). This table can convert a number into a password, but the number has to be roughly twice as long as the password, which is a pretty significant drawback. A full table would be much more useful. All you need to do is decide on rows or columns first and look up the character at that “address”. So, with this table, 12-07-1941 would became `bzzi4`, which is a pretty weak password because it is so short. Still, it is easy to make the charts for these systems, and to add letters to the rows and column headers. The best reason to use these systems is that you can carry the card on you without worrying about anyone being able to figure out your passwords.

Dice and A Word List

One of the best pen-and-paper ways to create passwords is Diceware, available at http://world.std.com/~reinhold/diceware.html. Technically Dice-

ware makes something between a password and a **passphrase**. Diceware passwords can be strong because physical dice are used to find words at random from the Diceware word list. As long as you use common sense when generating a password, the resulting password should be strong enough for most uses.

The Diceware list is just a document with a column of numbers and another column with the words, something like this:

Table 5.2: Brief Example of the Diceware List

Results of 5 Die Rolls	Word
11111	A
11112	A'asia
11113	AA's
11114	AARP
11115	AARP's
11116	AAgr
11121	AAgr's
(. . .)	(. . .)
66656	zymurgies
66661	zymurgy
66662	zymurgy's
66663	zythum
66664	zyzzyva
66665	zyzzyvas
66666	zzz

By "use common sense", I mean do not just blindly accept the words that you get from Diceware. If you end up with something that is too short (less than about 20 characters) or is a common phrase (perhaps "no pain no gain") then redo or add another word or two. To be as secure as possible, be careful not to add any order — don't get 4 words and then put

them into alphabetical order, for example. Best to take them in the order you generate them.

Pretend that we roll five dice four times, to get four words from the Diceware list. Let's say we get the four words `correct`, `horse`, `battery` and `staple`. If we want, we could add some upper case letters, and end up with: `CorrectHorseBatteryStaple`. This is not hard to remember and is unlikely to be in a hacker's dictionary (actually, this one is in everyone's dictionaries!). It is not a great password, but it is strong and more importantly, it is easy to remember. Sometimes that is what you need — perhaps as your computer log in, for example.

In general, using words in a password is a bad idea. But, when you put several *unrelated* words together, you can make acceptable passwords. Especially when you choose the words randomly with something like Diceware.

A password made up of several random words with a few numbers or special characters is best for situations where you have to type the password without access to your password manager — such as logging in to your computer or logging into your password manager itself. In these cases, being able to remember the password is more important than extra randomness. Also, in both of these cases, you are logging in *locally* and not over the Internet. Thus, there is usually less risk of the password being exposed to anyone. Trading a little strength for easier to remember is a fair trade-off, in this case.

Paper methods have two main benefits. First, they are offline, so you do not have to worry about someone influencing, or spying on, your password. Second, you can customize both systems as you like, by making a new card or rearranging or modifying your Diceware list.

Strong Passwords: Systems to Transform

You can also use old-fashioned, pen-and-paper cryptography-like systems to create stronger passwords. These systems are more work, but also can result in much stronger passwords than you get just by putting words together. Also, they can start with easy-to-remember words or phrases, making them good systems for use with passwords you need when you don't have access to your password manager.

Mix Together

Star with two or three (or more) words that you will use for several sites or computer log ins. Do not type them in one after the other though. Instead, mix them together — for example, `newyork` and `hello` are your very easy to remember words. Mix them by typing in `newyork` and then go back and type `hello` into `newyork`, skipping one or two letters after each letter — `nheewylolrok`. Be sure to have a set system that you will use for every site — for example, always skip two letters. Do not try to skip one letter on some sites and two on other sites.

Although the above is not a bad password, it is still not a great password because it only contains lowercase letters. Capitalize a couple of letters and add some punctuation — `%NheewylolroK!` — and now we have a pretty strong password that would take several years to break with brute force. It is hard to remember that password, but easy to remember the system you used to make the password. But make sure that you remember your system. In this case it might be:

1. start with `%`
2. type `NewyorK` (first and last letter capitalized)
3. go back and type in `hello`, skipping one letter after each letter in `hello`
4. add `!` at the end

The biggest problem with the mixing system is that you are using the same password on every site. This is a very big problem! Do *not* use the same password on several sites. To avoid this, add an identifier for the web site or login that they are using to the password. This adds some *salt* to a regular system such as the one above and also help make it a bit more complex without making it much harder to remember.

For example, maybe you use the mixing system above with the two base words `e1ephan7` (replacing two letters with numbers) and `McQueen`. For your GMail password, maybe you add `GMail` at the beginning to make the password unique. You might end up with: `%GMaileM1ceQpuheaenn7!`. That is a very, very strong password. If you need to, you could type it in "from memory" without needing to write the password down anywhere. But, it still might be best to put it into a password manager. (Being able to recreate it from your system, though, will help when you don't have access to your password manager.)

Columnar Transposition

This system is for the crypto-geeks but can still make very strong passwords. Columnar transposition is a bit like the mixing system above, but more complicated. You will probably need to actually use paper to figure out your password, at least until you get used to it and can remember it.

Start with a long-ish password or a passphrase. You will probably want at least 20 characters, so this system is good if you need a very secure password but don't want to keep it in a password manager. Also, you probably don't want this for a password you have to type in all the time. Although the more you type it in, the better you will be able to remember it, so it is really up to you.

You have a passphrase, now choose a number. Something around 4 to 8 or so is usually good. Too big and your columns will be short and annoying. For our example here, lets go with 6. Next, write your passphrase into graph paper or other grid with six letters (the number you chose) per line. If you decide to use "This is my pass word!" as your phrase, you will end up with this:

T	h	i	s		i
s		m	y		p
a	s	s		w	o
r	d	!			

Table 5.3: Columnar Transposition Example

Now, take the letters off of the grid by column. In our example, we will get:[2] `Tsarh␣sd␣ims!sy␣␣␣wipo`. You can see how we just wrapped around at the bottom of each column and didn't worry when the columns got shorter (after the column ending with "!"). Our not very good passphrase `This␣is␣my␣pass␣word!` got turned into a much stronger password. We would have a hard time remembering the result but we can remember the system and the passphrase very easily, so we can get the password any time we need it.

[2] I put in ␣ to show the spaces.

ProTip: If you don't like the three spaces in a row that we got in our example, then use underscores (_) in place of spaces. This is also an easy way to add some special characters to your passphrase — although spaces and other white space also add complexity, so either is a good addition to a password. Unfortunately, many sites (in my experience) do not allow white space. Many others limit which special characters you can use. Oh, well. Security is a trade-off, but in this case, the sites made the trade-off for us.

ProTip: You can also do the transposition in a text editor and then copy and paste it into the log in field. Technically this is unsafe because other software can read the clipboard. But if you have malware reading your clipboard, it is also probably recording everything you type, and you are in big trouble anyway. To be safe, though, if you do cut-and-paste, copy something else as soon as you don't need that password anymore, so that the most recent entry in your clipboard is not your password. Also, close the text editor without saving the document.

There are many other old-style transformations you can do on passphrases. If there is one you like, go ahead and use it. There are also some ways that are sort of in-between methods. One way is to just take the first letter or two of each word of a passphrase. So "This is my pass word!" might become `ThIsMyPaWo!` if you take the first two letters of each word, capitalize the first one of each, and then add a `!` at the end. This is shorter, easier to remember, and faster and easier to recreate from the passphrase. Usually, it is also weaker.

If you do use this "first letter or two" system, I recommend a slightly longer passphrase and a less obvious system of adding capitals and other characters. Something like capitalizing hard consonant sounds and adding a comma after 3rd, 5th, 7th, and 11th letters would probably be a good-enough system.

Whatever your system, you should use the same system (but not the same password!) on every web site for which you use this system. It is okay to use computer-generated passwords on some sites, Diceware or card / chart passwords or passphrases on some sites, and these old-fashioned systems on other sites. But, if you make three different systems of the same type, it is much easier to get them confused and to forget which system you used where.

Strong Passwords: Passphrases

An easy way to make your password more secure by making it longer. Unfortunately, this will not work if the web site or computer administrator limits password length for some reason.

We have already talked a bit about passphrases. Now, we will look at them in a little more depth.

The difference between a passphrase and a password, is mostly two things: first, a passphrase in likely longer than most passwords, and second, a passphrase is more likely to be a meaningful sentence or phrase.

A passphrase could be a sentence. Or a paragraph. Security is a trade-off, so if you really, really need to be safe, and don't want to write anything down anywhere, a pass-paragraph might make sense. More likely, a passphrase will be fine.

For example, instead of the password `19Dog91Pug,` use the passphrase `I bought my first dog, a pug, in 1991.` That passphrase has 38 characters, and includes numbers, upper and lower case letters, and even commas, periods, and spaces! Of course, it might be a bit much to type in 10 times a day, but once or twice a week for an important computer or web site, it might be a great passphrase. The loss of randomness from using human language phrases is more than made up for by length and character variety. A 38-character random password with similar character variety would be much, much stronger than our example passphrase, but, at 38 characters, the difference is not going to matter that much.

Passphrases can be just about anything that the software or web page you are using will allow. But, as I mentioned earlier, do *NOT* use famous quotes, passages from a book or poem, etc., or anything else that a lot of people in your culture would probably recognize. Things such as `Use the force, Luke` and `The path of the righteous man is beset on all sides by the inequities of the selfish and the tyranny of evil men` are *bad* passphrases. Yes, that second one is very long and has lots of variety. *But*, it is well-known, from two sources at least. Passphrases like those two examples are in hackers' tool kits and are not safe to use. Basically, if it is from a movie; Shakespeare or other famous author, a commercial, or a celebrity, it is probably not a safe passphrase.

Conclusion

In this chapter, we learned several methods for making strong passwords. There are, of course, many more systems, but these should meet the needs of most people. Check out some of the links at the end of the book if you are interested in learning more about ways of making passwords.

Let me say again however: please get a password manager and use it. Let it make strong passwords for you. If you are worried about forgetting the password to your password manager, then write it down. Yes, you can write down your passwords. Just don't keep the paper with the password someplace obvious. (See Chapter 10 for more information on writing down your passwords.)

Chapter 6

Passwords: What They Are & How They Work

If you are reading this book, you already have an idea of what a password is. An important thing about passwords that you might not know, however, is that when you use passwords to identify yourself, it is called *authenticating*, or *authentication*. By proving that you know your password, you get access to the stuff that is yours on the computer, web site, or to other stuff that you have been allowed access to (perhaps on a shared network drive).

Notice that the computer doesn't know or care if the person who entered the password and is now accessing your data is really you. Maybe you gave your coworker your password so they could get a critical document off your computer and email it to you for an important meeting. Maybe you left your password on a note on your monitor and an unscrupulous coworker logged in as you in order to go through your files. Your computer doesn't care. The correct password was entered; access is allowed — and anything that coworker does while logged in as you gets logged as you. If your coworker does something evil, you will have to prove to your boss and maybe the to the police that it wasn't actually you who did it. Whatever evil thing "it" was.

There are other ways to authenticate people. A password is an example of "something you know" and anyone who knows the password is allowed access. Another way is "something you have." Keys and cards are the usual examples of this, but fingerprints and other biometrics are also examples of "something you have" where the things are a bit harder to copy. Keys and key cards are obviously easy to lose or have stolen. Passwords can

be forgotten, revealed, or guessed, but are not as easy to "steal" in the traditional sense. They can also be changed more easily than keys and key cards, making passwords easier to use and more cost-effective for logging in to computers and web sites. This is another reason that passwords will probably be with us for a while yet.

Security is a Trade-Off

Realistically, who cares about cost effective? Who cares about easy to use? Well, okay, we all care about easy to use. And cost effective, when we are paying. As we have seen a few times already, security is always a trade-off. You want to be totally secure in your daily life? Put on a suit of armor, hide in your basement, and never go outside. You will be almost as safe as humanly possible. At least you will be, until the bank takes away your house, because you didn't pay your mortgage.

You have to decide on the amount of security you need for a given situation. You also need to consider the amount of time, effort, and money that you are willing to invest in security. Governments tend to invest a lot of time, money, and effort in securing their presidents, kings, dictators, whoever, because even one mistake can have catastrophic consequences. I am not willing to invest quite that amount of effort, time, or money in protecting my social media account. It just isn't worth it for me. (And yes, I only have one social media account.)

More concretely, if you spend $100 a year on a safe-deposit box to protect a comic book worth $25, you are obviously wasting your money. The same holds true for passwords and computer security. If you spend several hours a week changing passwords to all the web sites you visit (banks, social networking sites, etc), you are probably losing money and wasting time. Probably, you don't want to be doing that.

ProTip: Ignore sites that tell you you should change your password every six months or so. That is no longer a best practice — that advice comes from the computer stone ages, before anyone did research on passwords. Today, the best practice is to make a strong password and use it for as long as you can. Usually, you **should** only need to change a (unique, strong) password if it was compromised — perhaps the web site was hacked or you discovered malware on your computer. In both of those cases, it is possible that the bad guys got, or could get, your password,

so you should change it. You should not have to change a password just because it has been six months since you last changed it.

Passwords, luckily, are cheap. Free actually. And strong ones are just as free as weak ones. Using strong passwords is one of the most cost-effective steps you can take to secure your on-line life. Before we get into the details of strong passwords, let's look at how passwords work.

How Passwords Work

You turn on your computer and log in. Or maybe you don't log in — maybe the machine does that for you automatically. Either way, you have a user name and a password to control access to your account. How does the machine know the person trying to log into the computer is you? Obviously, it looks at your password, right? Well, sort of, but not exactly.

The log in screen is actually a program, aka software. All this program does is present a nice picture and have a place for the user to put in a user name and a password. What the log in program does next is hand off the information it was given to *another* program and ask *"Is this the correct password for this user?"* That other program looks at the list of passwords for users on the computer and tells the log in program if the password is correct or not. If it is, the log in program starts a few other programs, which then take over from it and you are able to get to work. If the password is not correct, the log in program gives you an error, and you (probably) get to try again.

The problem, you might have noticed, comes in when that program looks at the list of passwords for all the users on the system. If anyone can look at that list, everyone can see everyone else's passwords! "Where is the security in that?" you might ask.

You would be right to ask, because if everyone could see the list of users and passwords, there would not be any security at all. That is why the smart people who design operating systems (the software that run computers, talks to hardware, and does things like manage who can see what on the computer) set up the system so that only certain programs can see the list of passwords. However, the real security comes from the way the passwords are stored in that file.

Hashes

This might sound weird at first, but that list of all the users and passwords on the computer usually *does not* include the passwords. "Say what?" you might be saying. As we saw above, people and programs need access to that list, so including the passwords themselves would not be secure. Instead of storing the passwords, the file contains *hashes* of the passwords. Your password might be `hello123` but the hash listed for your password might be `4233137d1c510f2e55ba5cb220b864b11033f156`. The hash is created from your password by doing a lot of difficult math stuff to it.

Creating hashes is easy if you are a computer, but even for a computer, it is *almost impossible* to take a hash and reverse it to recreate someone's password. So with `hello123` I can get `4233137d1c510f2e55ba5cb220b864b11033f156` easily, no problem, whenever I need to. Well, I cannot, but my computer can.

However, not even governments, with lots of money, super computers, and geniuses, can take `4233137d1c510f2e55ba5cb220b864b11033f156` and reverse it, easily or quickly, into `hello123`. The fact that it is almost impossible to reverse a hash makes it safe to store them in a file on the computer.

So, when you log in, the log in program does not send `hello123` to be checked at all. Instead, it hashes the `hello123` and asks the password-checking program "Is the password hash for user 'Chris' equal to `4233137d1c510f2e55ba5cb220b864b11033f156`?" If it is, Chris gets to log in. If it is not, he gets to check that the CAPS lock key is not on and try again. Either way, the log in program did not send the actual password to any other program. Only the log in program saw my password. This limits the chances of *malware* capturing the password.

The same sort of process happens when you log in to web sites. The process is a bit more complicated, because all the information has to go over the Internet, but the process is basically the same. In the real world it took you a lot longer to read that explanation than it takes your computer or a web site to do all of that and log you in.

There is a lot more detail to hashes and log in security. For example, there are several ways of doing the math for creating hashes and some are more secure than others. Another example is what if two users have the same password? If Joe and Steve both have the same hash listed in the password file, wouldn't they be able to figure out that they have the same

password? How do those operating system geniuses deal with that?! (Short answer: they thought of that. They dealt with it.)

We are not going to get into *all* of that, but hashes really are very important for password security. We will look at some of it in the next chapter because attacking those hashes is one of the ways that bad guys try to recover passwords.

Chapter 7

Types of Attacks

In this chapter we are going to look at some of the ways the bad guys can try to attack your passwords.

As we have talked about, passwords as we know them today are not the easiest thing for people to remember. Worse, the stronger the password, the harder it is, in general, for a typical person to remember. Our memory is usually the weakest link in the password chain. Unfortunately, the easier a password is to remember, the easier it is for a bad guy to guess and the weaker it is. For example, `gravytrain` is a pretty bad password and `thr*)1!qqJ75XAn` is a very strong password.

Weak passwords are weak because they are easy to *recover*. Passwords are recovered through various attacks, including *brute force* attacks, *dictionary* attacks, *hash* attacks, and others. Usually bad guys will choose attacks based on what sort of information they have, and how big a hurry they are in.

Brute Force

"Brute force" is really not a sophisticated attack. It is almost always the most time-consuming attack. In fact, if a bad guy, or a security researcher, finds a new attack, it needs to be faster than brute force to be considered worthwhile. Brute force is what you are doing if you try every combination on a 3-number luggage or briefcase lock (`000`, `001`, `002`, ... `998`, `999`). Eventually you will get to the correct sequence, but it may take you a long time to get there. The difference with computers and passwords is that

a computer can easily try several thousand passwords every second. The bad guys might start with `aaaa` and try every possible combination of letters and numbers up through `999999999` (for example). That would take a long time, but if your password is `A11ofU5` (that is, if your password is less than 10 characters and only letters and numbers), the bad guys will find it. They might find it in 10 minutes, or they may not find it for 5 years, but they will find it.

Any decent web site is going to prevent a brute-force attack by limiting the number of times you can try to login with an incorrect password. Maybe after three incorrect attempts, the account is locked and you get an email with directions to unlock it.[1] Or maybe the site only allows three login attempts per minute — that would slow down the attack so much that it would probably not be worth the attacker's time. At three attempts per minute, the attacker could only try 4,320 attacks per day. That sounds like a lot, but a six-letter password with only lowercase letters, has 26^6 (26 to the sixth power, or $26x26x26x26x26x26$, or about 308 million) possible combinations between `aaaaaa` and `zzzzzz`. A typical desktop computer could easily try hundreds or thousands of passwords per minute if web sites did not prevent this sort of brute force attack. (Note: Even though it may sound strong here, a six-letter password with only lowercase letters is very, very weak, and you should not use one that short if you have a choice.)

Brute force attacks are pretty easily detected, if anyone is looking. Brute force is sort of the computer equivalent of a burglar standing at your front door with a thousand keys, trying each one until he finds one that opens your door or he runs out of keys. After a couple of minutes, someone in the neighborhood is going to notice, right? After an hour or two, hopefully someone has called the police!

Back in the digital world, brute force is the least likely direct attack. However, we need to understand it because all other attacks are measured against the brute force attack. *All* the other attacks we will talk about are much, much more dangerous than brute force attacks. The first attack we will look at is the *dictionary attack*.

[1] At the school I teach at, and probably every school on the planet, students sometimes maliciously lock their friends out of their (the friend's) phone. They do this by purposely putting the wrong password into the phone's lock screen until the phone decides to block all access. Usually this is a rage- or depression- inducing prank.

Dictionary Attacks

Dictionary attacks are similar to brute force attacks but a bit more intelligent. The biggest difference is that dictionary attacks use a list of words (thus the name) and especially words or phrases that are often used as passwords. Like a brute force attack, a dictionary attack will try each possible password one at a time. Because the dictionary attack uses common passwords instead of just every possible combination of letters, numbers, and special characters, the dictionary attack is more likely to be successful against weak passwords *in a reasonable amount of time*. For example, a common, but complex, password that would take years to brute force might be successfully attacked with a dictionary in only a couple of hours, or even minutes.

Dictionaries of passwords are available on the Internet from both bad guy suppliers and security researchers. Most include common letter substitutions, such as `!` or `1` for `l` or `I` (those are lowercase 'L' and uppercase 'i', respectively), `3` for `E`, `7` for `T`, and so on. `Isle` is a horrible password. Making it into `1s!3` would not be all that much better because if the bad guys even bothered trying 4-letter passwords, they would certainly try all those substitutions. (Not that it matters: a four letter password could be broken by a brute force attack, on a regular home computer, in a minute or two at most.)

This leads to an important point: a password may be very strong against a brute-force attack but still very weak against a dictionary attack. If you are using software or a web site to check the strength of your password, *be sure you understand the results it gives you*. If the software or site is just reporting the strength against brute force, that does not automatically mean you have a strong password.

For example, `Supercal1fragi1isticexp1alid0ci0us` will likely be reported as a very, very strong password. This should be a strong password: it is long (over 30 characters) and has numbers and upper- and lower-case letters. **But**, because the word is from a famous movie, there is a good chance that it is in a dictionary somewhere — and if it wasn't before, it probably is now that it has appeared in a book like this one. That makes the password not as good as it would be otherwise, even with the number-letter substitutions. Not all password strength meters check this, so be careful using them. Luckily, humans can check this very easily. If you can recognize the password as a word or phrase, that password (even with

number–letter substitutions) is probably in a password dictionary somewhere and is probably *NOT* a strong password.

This does not automatically mean that words in passwords (aka passphrases) are a bad idea. We just have to be careful not to use combinations of words, or short phrases, that are likely to be in a dictionary. As we talked about earlier, putting four medium-length, unrelated words together (such as `XmasKickAugustWaterfa11`) can be a strong passphrase. `UseTheForceLuke`, though, should be considered a very weak passphrase.

Hash Dictionary Attacks

Pretend your evil coworker, Alice, is trying to get into your account so that she can steal your report and send it to the boss as her work. Maybe she comes in late one night, after most everyone has left. She sits at your desk and tries to log in, but she needs your password. How is she going to log in without it? Well, she might try just guessing it — a brute force attack. Depending on how well she knows you (or how much research she has done on you), she might try likely passwords first (a *dictionary attack* — where the 'dictionary' is customized to you).

But wait a minute, you might be saying, *she cannot sit at my desk all night trying different passwords until she gets lucky. Someone would notice eventually.* You might be right. Alice probably knows that, too, and planned for it. She will not use one of those time-consuming attacks, if she can avoid it. She will try something else, something more dangerous.

Unfortunately for us, Alice doesn't have to try to watch you type your password — although that is a great way to get one if you have the chance — or risk sitting in front of your computer for hours trying to brute force your password. Alice can use the **hash dictionary attack** (also called a **reverse dictionary** attack) against our password. *If she can get access* to the list of users and their password hashes on your computer, Alice can compare them to a list of hashes for common passwords.

We talked about hashes earlier (pg 6). If the bad guys can get a list of password hashes (or even better, a list of user names and password hashes), they will try a hash dictionary attack. Hashes are easy to create, so hash dictionaries are easy to create. Once created, they can be used over and over. Hash dictionary attacks are sort of brute-force attacks that are all set and ready to go ahead of time, speeding up the attack considerably.

Alice can spend weeks, months, or even years creating a list of hashes for many, many, many, possible passwords. More likely, she downloads one from the Internet, or buys one from a person of questionable morals. We said earlier that it is impossible to *reverse* a hash to get the associated password. That is true. But, there is nothing to stop her from creating or buying a list of passwords and their associated hashes. All she has to do, if she can get access to your hash, is look up that hash in her list and see what password created it.

Checking for your password hash in her list of hashes would not take long at all — not long, as in maybe a minute, if Alice was unlucky and if she had a truly huge list of hashes. With this attack, it doesn't matter that hashes cannot be reversed. The dictionaries the attackers use link each possible password with it's hash. So, if she finds your hash, `4233137d1c510f2e55ba5cb220b864b11033f156`, in her list of hashes, she can see that `password123` is the password associated with that hash. Now, Alice can log in as you anytime she wants, with no delay at all. Trust me when I tell you that the hash for `password123` is in ***every*** list of hashes on the Internet. So are the hashes for *every* word in a typical English dictionary. As are `qwerty` (the first six letters across on a typical keyboard), `123456`, and *every other weak password* that is in use by people not as security-conscious as you and me.

Of course, the bad guys cannot use a hash dictionary attack to get any passwords unless they get that file with the list of user names and password hashes. You and I cannot control that — we have to hope that the software and web sites we use store our data securely. Every so often, however, a company makes news because hackers managed to get into the company's systems and take all sorts of data. Sometimes the company notices right away. Other times, not so soon. But we cannot control this. What we can do is use strong passwords, so that *our* passwords do not get "found" and used before the company notices the breach and warns us.[2]

[2]Troy Hunt is a security researcher. He has a web site "have I been pwned" (https://haveibeenpwned.com) where you can check to see if your email address was exposed by a company that got hacked. Usually, if the hackers got the email addresses, they also got the hashes of the passwords. If your email address was released in a hack, then assume that your password for that site is also compromised. ("pwn" — pronounced like "owned" with a "p" in the beginning — means to dominate or conquer something. It started in online games but now is used more widely on the Internet.)

This situation — web site operators losing the file of user names and hashes, and the bad guys trying to find the passwords that go with those hashes — is exactly the problem that Gibson's SQRL is intended to solve.

Side Note: Passwords in Plain Text

If you ever hear of a company getting hacked, and they stored their passwords in **plain text** or **clear text**, stop dealing with that company immediately. Storing passwords in plain or clear text means that they did not store hashes of the passwords, but they actually stored the passwords as regular text — just the way that you would type it in. Which means that anyone who can get access to the file of passwords automatically gets **all** the user names and passwords!

A company that does this either has no understanding of security or no desire to do it correctly. Either way, if they have anything at all to do with your money, run away from them as fast as you can, after you close your account and delete any stored credit card or other financial information. Seriously and no joke. There is no reason at all, ever, for **any** site to store passwords in plain text.

Kind of, sort of related to the above: sites **never** need to ask you for your password. This is not because they can already know your password — they should not (if they store the hash and not the password, that is). Rather, this is because they do not need your password. They cannot use the password, but they can delete it or reset it. Modern operating systems and the software that runs on top of them are able to allow some people to access other accounts without a the password for that account.[3] These lucky, hopefully highly-ethical people can also reset passwords, lock accounts, and so on. They do not need your password to do this. This has been true since the beginning of multi-user computer systems decades ago. When you click on the "forgot my password" link on a web site, basically the site software just resets your password and sends you a special log in link to reset it again yourself. Or, they may send you a temporary log in password — same thing, they just reset your password to that temporary password. So, point of this digression is: ignore any email or text that asks you for your password. The legit site does **NOT** need it.

[3]The special user will probably be required to confirm their own password however, to prove they are who they are supposed to be, before they can change or delete someone else's password.

Technical Details

This attack by looking up the hash of your password in a table of hashes can be defended against in two ways. The first, using **salt**, is not something you and I (typical users) have control over. The people who make the software we use and the web sites we visit either use salt or they do not. "Salt" basically involves adding a known but random value to your password when you create it. Meaning, they add, for example, `a9tl` to the front or end of your password before hashing it. The web site or your computer stores the salt and automatically adds it to your password each time you log in, before hashing, to check if the password is correct.

The salt is important but does not have to be secret. In fact, there is a good chance that if the bad guys have the file with user names and password hashes, they probably have the salts as well. Because of the magical(-seeming) properties of hashes, without the password, the salt is useless. Just as there is no way to figure out the password from the hash, there is no way to use the salt and the hash together to find the password.

Using salt to create password hashes means that the reverse dictionaries that the bad guys have are either worthless, or have to be much, much bigger. Bigger, because instead of having the hashes for `password`, the dictionary has to have hashes for `(all possible salt values)` + `password`. In the real world, if the salt is big enough, the hackers cannot (currently) win. Even if the salt is smaller, it still makes it much more difficult for the bad guys to recover passwords. Today, there is absolutely no reason for a web site not to use salt when storing passwords.

As a side benefit, using a random salt for each user also ensures that two people with the same password will not have the same hash. Maybe Joe and Steve, just accidentally, have the same password. Because of the salt, no one who sees the password hash file will ever know that Joe and Steve are both using the same password. Problem solved.

Actually, an important feature of hashes, and thus salted hashes as well, is that similar passwords (salted or not) do *not* have similar hashes. If even *just one character* in the password is different, the hash will most likely be very different. Seeing two or more hashes that look alike does not mean that the passwords that made the hashes are similar. Hashes don't reveal any information about the passwords they came from.

Like I said above, salt is not really under your control. The second method is under our control (and much easier to understand): use strong

passwords. You knew I was going to say that, didn't you? Before we look again at strong passwords, though, lets look at a few other ways that bad guys can attack your passwords.

Other Attacks

These attacks are often not affected by the strength of the password and are thus much more dangerous. One type of attack was mentioned earlier and is called **social engineering**. This is when you are tricked into giving your password away. Sometimes it might be a link in a legitimate-looking email that takes you to a fake site that looks like the real site. Thinking that you are on the real site, you log in and the bad guys now have your user name and password. No time consuming brute-force or dictionary attacks needed. The danger of malicious links of this sort is one reason why security experts recommend never clicking on links from unsolicited emails. (I recommend going even further and not using HTML email, just plain text. But, every year I am more and more in the minority in holding that opinion.) To protect against this, always check where links actually go, not just what is displayed in the email. HTML can show one thing but link to another. Also, use your own bookmarks for important sites (aka sites that involve your money) that you go to — such as banks and shopping sites. Don't click on the link in the email; instead, open a new tab in your browser, and go to the site from your bookmarks.

Keyloggers

Another danger is keylogging software. If this sort of software is on your computer, you already have a serious security issue that needs to be addressed immediately. keyloggers do exactly what the name says — they record ("log") every key you press and usually send that data to the bad guys. Finding your password in all the data is not always easy, but it is not exactly hard either. Certainly finding a password in the data from a keylogger is much, much easier than trying to brute force your password or trying to get into a secure site and steal a list of hashes.

To add to the nighmare that is software keylogging, there are also physical keyloggers that can be plugged into your keyboard cord, and then into your computer. No software needs to be installed on your computer, but

someone has to physically come and install — and remove — the logger. These are also much more likely to be discovered. They are useful, however, because you do not need to install software on the computer (which, ironically, might require knowing a password) and because they do not require any technical expertise. Anyone who can plug in a USB cord can plug in a hardware keylogger. Such as, for example, spouses, parents of teen-age children, and unscrupulous coworkers.

Further Reading

Defending against social engineering, keyloggers, and other malware is a bit off topic for this book, though. You should certainly learn more about them, if you want to be as safe as possible on-line. Start with the section on "Computer Security Beyond Passwords" on page 60, in Chapter 10. Then, check the links in the back for some good sites for learning more about computer security.

Chapter 8

Strong Passwords, Part II

Now that we have a clearer idea of how bad guys attack passwords, we can look again at what we can do to avoid or slow down some of those attacks. First, we need to talk a little about **randomness** because it is an important part of making strong passwords.

Randomness

We have not discussed **randomness** in depth yet. We all probably have a simple idea of what constitutes "randomness" but often, our understanding is not entirely accurate. For example, some people complain (or are at least confused) if their music player plays songs from one artist more than it plays songs from other artists. This seems like that artist is being favored and thus play back is not actually "random". That certainly could be the case, but more likely is that we are not correctly understanding randomness.

Random simply means that *previous results do not give any sort of information about what the next results will be.* For a truly random music player, if I hear a song from Singer X now, the next song should be equally likely to be from Singer X again. You could get several songs from the same singer or group in a row. That outcome would be just as likely as getting several from different groups or a couple from one group and a couple from other groups. Every outcome is equally likely — the song we just heard has no influence on what song we hear next. (Actually, some music players now include options for these sorts of 'random' — true ran-

dom, random favoring artist or album, random favoring frequently or infrequently played songs, etc. Only the first is actually 'random'.)

Think about the English alphabet and English spelling for a moment. For starters, not every letter is equally used. "E" is much more common in English than is "X". More words start with "S" than start with "Q". If you see "q" as the start of a word (or really anywhere in a word), what letter would you guess comes next? The answer, as you probably realized, is "u" because "qu" is a very common combination in English; "qz", however, does not happen at all in normal, everyday, English.

Language is *not* random. As with the "qu" above, there is a lot of information in language and this makes it not random. This is another reason why words are usually not great for passwords. In English, a "q" gives us some information about what letters might come next. The same thing is true if you see "governm" – you can easily see that "ent" or "ental" or the such will come next. "Xj9b" will almost certainly **not** come next. This means there is not much randomness here.

This is true in sentences as well. Grammar is a sort of information and information is not random. If you see "yesterday" in a sentence, you can guess that the verb is likely to be past tense. If you see "eats" you can guess that the subject will be third-person singular (that is, "he", "she", or "it", or appropriate noun such as "dog").

For us normal computer users, this is not really that important, as long as we realize that it means individual words, or even sentences, are going to make bad passwords or passphrases. We need to understand that when it comes to passwords, random is good, but words (in any language) are not random.

As we talked about before, however, four or five medium-length words **randomly chosen** can make a strong-enough password. Such a password will not be as strong as a truly random password of the same length, but it will be much easier to remember. Sometimes being able to quickly and easily remember a password is a more important consideration. Remember: security is a trade-off. Sometimes losing a little security to get a more easily remembered password is an acceptable trade-off. For example, when you log in to your main computer, you do not have access to your password manager (because you need to be logged in before you can access it). A four-random-word password might make an acceptable login password. I would not use a four-random-word password for my banking site, however.

Back to Strong Passwords

We can now expand the list we started earlier, of characteristics of strong passwords. A strong password:

- is hard for a human to guess (even if the bad guys know you)
- is not in any dictionaries (even of other languages)
- in not in lists of passwords (or other "hacker" dictionaries)
- hashes to something that is not in any hash dictionaries (that is, it is close to unique)
- has lots of randomness (knowing the first couple of characters will not help the bad guy figure out the next few characters).
- is long enough to withstand a contemporary brute-force attack (don't use 4- or 6-character passwords. Use at least a 10-character password, or a 20-to-30-character password if you want to future-proof it).

Basically, the more unusual, the less like a real human language, and the more random a password, the stronger it is. A strong password is less likely to be guessed or to succumb quickly to a brute-force attack or any dictionary attack. Of course, these passwords are hard to remember, so use a password manager.

These requirements are why some companies and web sites have silly password rules like *more than 8 characters, of which at least 2 must be numbers and 1 or more must be special characters.* I call these rules silly because although in general such guidelines do make sense from a security viewpoint, in reality such rules almost guarantee that users will decide on a fairly simple, easy to remember password. Worse, they may use it everywhere that has similar rules. Using an easy to remember password is bad enough, but using it for more than one log in or web site is even worse.

For example, faced with the rules above, many people might use something like `1Password!`. Technically, that is not a horrible password. In theory, it would take about 50 years for a desktop computer to brute-force that password. It meets all the requirements and is easy to remember. That password meets the letter of the law, but not the intent. Anything

with `password` in it is likely going to be very quickly discovered in a dictionary or hash attack.

Let's review what makes a password hard to guess.

Hard to Guess

A hard-to-guess password is:

- not a word
- not something that someone could guess if they knew you (like your birthday or a pet's name)
- almost always longer (at least 8 characters, but 12 or more is best)
- as random as possible
- includes characters of all types — upper, lower, numbers, special characters, and white space if allowed. Actually, the longer the password, the less character types are required:
  - A 6-character password needs all the character types it can get (and is still too weak, today).
  - A 30-character password (that is NOT a passphrase) could be all lowercase letters, really, because it is long enough that it is plenty strong enough without the extra randomness from the extra character types. (For now — this could change in a few years, due to either faster computers or better "recovery" tools, or something else that we just can't predict today.)
  - BUT, this is only true for an actual random password. A 30-character passphrase in English has lots of information in it, and thus is not ***as secure***.[1] So, it would benefit from extra character types.

I hope this seems pretty anti-climactic by this point. A strong password is all the things that a weak, easily-guessed password is not.

Now that we understand all (okay, not all) about passwords, let's look briefly at these password managers that I keep talking about.

[1] To be honest, a 30-character passphrase that is NOT from a book, movie, song, religious text, etc is pretty strong. It is not as strong as a 30-character random password.

Chapter 9

Password Managers

Please use a password manager. The best and easiest way to increase your online security is to use a password manager and let it make strong passwords for sites you log in to.

Some people do not like the idea of online password managers, so they are separated out into online and offline managers.

Offline Managers

Offline managers run on your computer and do not need to access the Internet. This is probably safer, but it is much less convenient. Security is always a trade-off, remember?

If you mostly only use one computer, you may find an offline password manager works best for you. Just be sure to make multiple backups of your password wallet! If it is only on one machine and something happens to that machine, you could be in a difficult situation.

KeePassXC runs on MS Windows, Mac OSX, and Linux. It is based on KeePassX, which was based on the original KeePass. It is Free / Libre / Open Source Software — so should be fairly trustworthy — and has been around for a long time. At the time I am writing this, I am using KeePassXC as my local password manager.

Password Safe is perhaps the original standalone password manager, dating back to 2002. It was originally written by Bruce Schneier

himself, but has been developed and maintained by other people for many years.

Your Browser has a tool to save passwords. Personally, I don't recommend using it, because I find standalone password manager software to be easier to use. Note that I am *not* talking about add-ons such as a LastPass or Bitwarden extension; I am talking about the manager built in by Mozilla or Google or Apple, etc. Using the browser built-in locks you into that browser. If you only use one browser, then maybe no problem. If you use multiple browsers (use one only for online banking, for example) then you may end up with different log in passwords scattered across browsers, with no way to get them all into one place.

Your OS may have a tool to save passwords. I believe that Mac OSX has a sort of password manager built in. I'm not sure about MS Windows. KDE on Linux has KWallet, which I used for many years back in the days before KDE 4. (KWallet is still around — I just don't use KDE any more.)

Point is, you can also check if your operating system includes a tool for saving passwords. Personally, I would not suggest using them, even if they have it, because it is unlikely to be cross-platform or cross-machine. That is, your wallet may end up tied to just one machine. With KeePassXC or Password Safe, you can copy the wallet file to another computer and use it there, easily. It may not be as easy with an OS-dependent wallet.

Online Managers

Online password managers might have software you can install to your computer, but usually you use them in your browser. To do this, you need to install a browser extension or add-on. They should store your data online only as an encrypted file. That means that the company cannot see your passwords or other data. (Of course, you usually have to just take the company's word for that.)

The convenience of online password managers is that they can sync your data across all of your computers, as well as your smartphones. (See

my worries above about getting locked into one browser or one machine.) Also, because they are added on to your browser, they can easily fill in your user name, password, and other information on web sites.

If you need to log in to several sites from different computers or devices, you may want an online password manager.

1Password — I've not used it, but 1Password has been around for years and is generally regarded as safe. I'm not an expert, but they also seem to have avoided a lot of the community drama that LastPass seems to create for itself every few years.

Bitwarden — I've recently switched to Bitwarden. It is Open Source (always a good thing for security software), including the server (unusual and a very good thing, but normal computer users don't need to worry about it). It has free and paid tiers. As I write this in 2021, the paid tier starts at $10 a year.

LastPass — I used LastPass for many years and liked it. The original company was purchased by a larger corporation a few years ago, which caused some concern, but they maintained their quality and security. Recent changes, however, to pricing and features have once again created concerns in the community, leading some people to switch away. They have free and paid tiers.

LastPass is generally easier to use than Bitwarden (I've not used 1Password), so I would recommend people less familiar with tech to try 1Password and LastPass to see which they like better. Slightly tech-savvy people and up, I would recommend use Bitwarden.

That said, there are more password managers out there. Do your homework, try a few out, and find the one that you are most comfortable with, and use it. Maybe start with looking at Wikipedia's list of password managers. Also, check news sites (such as The Register and Ars Technica) for any recent news about any password manager you are thinking of using.

Now, let's finish up by looking as a few other issues around passwords, such as how to protect them (aside from keeping them in a password manager, of course!), what to do about security questions, and computer security in general.

Chapter 10

Tips, Tricks, & Closing Notes

We talked a bit about using passphrases and transformations because sometimes you just cannot write down your password / passphrase securely. Keeping your computer login password in a password manager that you cannot access until you are logged into your computer would be counterproductive, to say the least. For other passwords though, especially to important sites such as banking or other financial sites, you should use a password manager to securely store long, strong, passwords and passphrases. (There is a reason I sound like a broken record about this. For those of you who remember "records", that is.) Password managers also allow you to store notes with the passphrases and user names — such as the answers to your security questions for each site.

ProTip: Speaking of security questions — lie. No one says that you have to use real data for these. The answer does not even have to match the questions. Street address where I grew up? `TTZ1lwhrqsyt$`. City where my parents met? `73AxelPuff`. (Not my actual answers, if you were wondering.) If you use a password manager, you can keep your "answers" safe and not have to worry about trying to remember them.

Protecting Passwords

Okay, so now you know what strong passwords are, and how to make them. Next — how do you protect those passwords? What good is a strong password if everyone has access to it?

First consider who are you protecting your password from. If you are worried about hackers getting into your computer and stealing your banking passwords, then you might consider writing them on paper and storing that paper in a safe (or just hidden somewhere) in your house. If you do not log in very often, and are not worried about people in your house maybe finding the list and using it, keeping passwords on paper might be a totally okay thing for you to do.

For online sites you go to a lot, but maybe aren't that important financially? It is probably a good idea to keep them in a password manager.

For your local log in? At work, it is ***not okay*** to store those passwords on a post-it note on your monitor. At home, it might be. However, it is probably always going to be okay to keep that password on a piece of paper in your wallet (or purse), assuming you don't leave it unattended near your computer. But, don't put any user names or web site details on that piece of paper.

Do not use a single password for more than one login or web site. If you absolutely have to reuse a password, at least use the password for different types of access: physical log in at a computer and for a web site. An overseas criminal trying to get your PayPal password is probably never going to log in to your home computer. So using the same password for a web site and a computer account login is less bad (*but still bad!*) than using the same password for two web sites. But please, please, never use any password for more than one web site. Certainly never reuse a password that you use with banking or other financial sites.

Computer Security Beyond Passwords

Good strong passwords are just one part of computer security. For the online world, they may be the most important part. Here are a few other things to think about, though.

Physical Access

Most people do not need to worry about this, but the truth is that if someone gets physical access to your computer, the game is over. I am not an expert, but if I have physical access to your computer, I can get to all the

data and install any software or hardware that I want. I can even change your password! That is me; imagine what the experts can do.

The best way to protect against this (assuming you cannot physically protect the computer itself) is to use full-disk encryption. MS Windows, Apple, and Linux all offer full-disk encryption. But, guess what? Yup — you'll still need a strong but memorable password to use to encrypt and decrypt your hard drive.

You can also check for physical keyloggers and other new hardware attached to your computer. Spouses and parents of teenagers are often the source of these sorts of security issues.

Two Factor Authentication

Two-Factor Authentication (2FA), also known as Multi-Factor Authentication (MFA), can be a strong way to protect your computer and your online accounts. Basically, 2FA requires another piece of information in addition to your password. Sending a short code or number to your email or to a smart phone is a common (but not totally secure) form of 2FA. Email is just totally insecure. The problem with sending codes (to your email or to a smart phone) is that it has to happen each time you log in. For most of us, this is really not an issue. But, more than a few celebrities and journalists have been attacked even with 2FA because the bad guys convinced the phone company to send the message to a new phone — a phone controlled by the bad guys (see "social engineering", below).

A better way to do 2FA, if your bank or other web site supports it, is to use an authentication app, such as Google Authenticator. These still use short codes that you enter when you log in. The difference is that you set up the authenticator app ahead of time. The web site and the app on your phone do some math magic and after that they both will have the same codes when you try to log in. That set up step is the only time that there is a danger of someone in the middle attacking the app (remember those "person-in-the-middle" attacks we talked about?). After the app and the server are set up, you just get the current code (it usually changes every 30 seconds). The server checks against its current code, and if they agree (and your password was correct), it lets you in. After that first set up, the app and the server do not communicate directly. No one can convince the phone company to send a message to another phone, because there is no message to send.

Another very strong 2FA method is the security key. YubiKey was one of the first widely available security keys for consumers. Although it is capable of much more, a YubiKey is basically very similar to an authenticator app, just on a small usb key instead of a phone. Instead of entering a code, the user plugs in the usb key and it sends the code. Just like the authenticator app, the security key is only vulnerable once, during initial set up.

A strong password plus two factor authentication is better than a strong password and no 2FA. So, usually it is better to set up 2FA if you can, even if that 2FA is "just" an SMS message to your phone.

Social Engineering

We talked a bit about social engineering earlier. Basically, social engineering attacks the social, human, element of security. If you try to trick someone into doing something, you are using social engineering.

Phishing emails look like emails from legitimate web sites, but are really from the bad guys. They are trying to get you to go to a fake, look-alike site where the bad guys are waiting to watch you "log in" to what you think is the real site. If you have ever gotten an email from a bank or other site that says something about needing you to log in to avoid some bad outcome (usually your account being locked or suspended), then you have seen a phishing email.

Avoid phishing emails by **never** clicking on links from emails, especially if it is asking you to go log in to a site. If it seems maybe legit, just go to that web site the way you usually would — from a bookmark, a link on your desktop, or typing the address directly into the location bar of your browser (the place where it says "www.google.com" or the name of whatever web site you are visiting).

Note that it doesn't matter if you know the person the email is supposedly from or not, or if it is an "external" email or not. The bad guys are getting better and better at faking it. They will even sometimes reply to an actual email (if they can get into someone's email account) so it looks like your friend really is suddenly asking you to send them some money. Do not trust any link to any site that you would log in to or download something from in any email, period. Better to go search for it (whatever "it" is) yourself and find the site that way. As Rachel Tobac said, be "po-

litely paranoid" — if your friend suddenly emails asking for money, call them (different communications channel!) and double check.

Yet another reason to use a password manager is because it will help protect you from phishing attacks. If you do click on a phishing link and end up at a fake, look-alike site, your password manager will **not** be fooled, and it will **not** fill in your user name and password. Notice this — if you really were at "paypal.com" for example, your password manager would see that and offer to fill in your user name and password. But, if you are at "peypal.com" or "paypa1.com", your password manager will not find a match for the site and will not fill in any information. (This is true more for online / browser-based password managers. Local managers may not be able to check the web page you are logging into — they are more likely to just paste wherever you tell them to. This is not their fault, it is just a reality of local software vs browser extension — the extension has access to information the local software does not.)

Phishing is just like real-life fishing; the bad guys throw out some bait and hope someone bites. They want to catch as many people as possible, but are not necessarily aimed at any individual. This is not the case with spear phishing.

Spear phishing is a type of phishing that depends on knowing about you and making the email look as legitimate as possible. For example, a company accountant might get an email from a vice-president of the company. The VP tells the accountant to transfer some money to a new bank account. There may even be an invoice attached. It might all look totally legit. If the accountant does not double check with the VP, he or she might fall for the spear phishing attempt and end up transferring a lot of money to the bad guys. (This has happened, probably many more times than we know.)

Backups

There is a saying "there are two types of people: those who make backups and those who haven't yet lost all their data." Please don't be the person who doesn't make backups until it is too late. Backup your data. Ideally, you should have two backups — one local (maybe an external hard drive) and one "off site" (perhaps a hard drive you leave somewhere else, or more likely a network backup).

Backups protect against theft, disk drive failures, and *ransomware*. Ransomware is malware that gets into your computer and encrypts all of your files and demands money to decrypt them. It doesn't hit just regular people; several cities, hospitals, school districts, and even police departments have been hit with ransomware. The rise of digital currencies has made ransomware quite profitable for groups with the expertise to do it.

Recently, ransomware groups have upped their game to a new level: now they will also exfiltrate (copy to their computers) all of your data! Then, they will search for stuff to blackmail you with. Usually it goes something like "pay us and we will give you the software to decrypt your data and we will delete your data from our servers." If you don't pay quickly, the group will dump some of your data on the Internet, and then maybe raise the ransom amount. This happened recently to both an Apple supplier and the Washington D.C. Police! (In the D.C. case, the bad guys threatened to release the names of confidential informants and others whose lives might be endangered by exposure.)

Like all malware, ransomware first needs to get into your system. For many people, this happens through links in phishing emails. So, please be careful with links in emails.

Sometimes, a good set of backups can rescue you from ransomware without you having to pay the bad guys. But, you should reformat the computer's hard drives and reinstall the operating system before restoring your data from a backup — if the ransomware is active (if you don't reformat the drives, it might be active), it could then reinfect your operating system and infect your backup. That would be bad. Worse, the bad guys have also taken to hanging out in infected computers — sometimes for months — before encrypting anything, which often means that when they do start encrypting, they also know about, and encrypt, your backups. Some even wait until like 2:00 AM on a Sunday morning (your time — they are probably in another country) to start the encryption, in hopes no one will notice for as long as possible.

The Cloud

If the only copy of your data is "in the cloud" (iCloud, Google Drive, Dropbox, etc.), then it is **not** backed up. If you use Google Drive, for example, you should consider downloading important information every now and again, just so you have it somewhere in addition to Google's servers.

Closing Notes

These were a few important considerations for computer security beyond passwords. I highly recommend you set up 2FA on important financial web sites, even if nowhere else. Others, such as physical access, most of us may not really have to worry too much about. In 2020 and 2021, ransomware is actually a pretty big threat, but more for businesses and other organizations. Regular users can, and do, get hit with it, but the professionals probably won't bother with one or two home computers — they can make a lot more money hitting a local government or a nice, juicy, small business.

If you keep your system updated (like, religiously), avoid (or at least double check) all links in emails, and make backups, you should be fairly safe (but I am not a lawyer; this is not legal advice). More importantly, for the topic of this book, if you use strong passwords, you should be as safe as you can make yourself against attacks against your log ins to web sites and your computers. To be as safe as possible, use a password manager with random passwords. Lie about your security question answers. (Basically, go reread Chapter 2.)

We've reached the end.

Thank you for reading *Every Sentient Being's Guide* to Password Security*. Let's take a look again at our goals. Back in Chapter 1, I said that when you were finished with this book you would be able to:

- explain to your friends and family, in general terms, what passwords are, how they work, and why they are important;
- decide how strong a password needs to be in any given situation;
- create strong passwords that are appropriate to the type of login;
- use software to securely store passwords, and other security data, for your computer logins, financial web sites, and other accounts.

I hope that *Every Sentient Being's Guide* to Password Security* has helped you accomplish these goals. If you have any suggestions, corrections, comments, criticisms, please email me at chris@ChrisSpackman.com.

Chapter 11

Suggested Readings

Recommended Sources

Here are some people, podcasts, and web sites I read or listen to frequently. They are great for slightly-tech-savvy people learning about, and keeping current with, computer security issues.

ArsTechnica has news on everything geeky / techy from security to board games. When they deal with security topics, they do a great job of explaining.

Bruce Schneier is a security expert who has written several of the best security books out there. His more recent works are especially good because they are less about the technical and cryptographical, and more about big-picture computer security. He has a blog at https://www.schneier.com/ and a monthly newsletter (subscribe at https://www.schneier.com/crypto-gram/) that is a digest of posts from the blog.

The Guardian is a general news site. They were involved with the Snowden disclosures, and they do a great job of covering big-picture security issues.

Naked Security is a podcast from Sophos, a security software and hardware firm. They do not go into as much depth as Steve Gibson does, but they generally cover similar news. They can add their own expe-

riences, though, of helping customers detect, mitigate, and recover from real-world attacks.

The Register is a tech site with a biting sense of humour (British spelling intentional — like The Guardian, they are a British site).

Security Now is a great podcast on the TWIT.tv network. It features Steve Gibson (of SQRL). A bit more technical and in-depth than some of the other resources, but very informative, and he does a great job explaining concepts.

This Week in Tech (TWIT) is another great podcast on the TWIT.tv network. The panel changes each week, and they discuss a lot more than just security. It is a great way to keep up to date with general tech news as well as any newsworthy security events / incidents / information.

Chapter 12

Software and Web Tools

Below are links to software mentioned in this book as well as a few other ones similar to the ones mentioned.

I am intentionally not linking to password managers here, because any links to existing software may be out of date by the time you read this, and because there may be newer tools that I didn't know about when I wrote this. I do recommend checking Wikipedia's List of Password Managers to compare before you try.

If you really want to check the ones that were current when I wrote this, see Chapter 9 on page 56.

Online Tools

Below are a few tools you can use to check your own passwords and your online security. Please, though, **never** put your real passwords into any site on the Internet (unless you logging in to that site, of course).

How Secure Is My Password? tells you how long it would take a typical computer to brute force your password. I suggest putting in a similar password, not your real one. For example, if you have a 17 character password with 3 common words with 2 numbers and 1 special character, such as "7DogEclipse!Sell9", put in something like "6CatAccount*Sold7". As long as the relative numbers of types of characters is the same, you should get a good idea of how secure your password is *from a brute force attack*.

Have I Been Pwned? is *the* resource for checking if some company lost your email address to hackers. If they did, you should probably change your password at that site.

GRC.com (The Gibson Research Coorporation) has many useful online and offline tools available.

Chapter 13

Copyright & Copyleft

About The Author

My name is Chris Spackman. I am an ESOL (English to Speakers of Other Languages) teacher in Columbus, Ohio. I taught EFL (English as a Foreign Language) in Japan for many years. I have used personal computers since at least the days of the Commodore 64 and more recently have experience with MS Windows (3.1 and up), Mac OSX, and Linux. I have used Linux since 1998 and have been a happy Gentoo Linux user for many, many years. You can reach me at: chris@ChrisSpackman.com

Version: 2021-05-a **Last Updated:** Monday 31 May 2021