Extract from "Keeping Your Data Secure II: The Human Factor"

Chapter XXX: Passwords

Passwords of one type or another have been in use for millennia. But using a password to secure important data, or control physical access to buildings and systems is rapidly becoming insufficient, or at least insufficient on its own. The Holy Grail of password security would be to find a method that can't be beaten by a determined and resourceful attacker, but so far the best we can do is to slow the attacker down a lot and try to make their life inconvenient.

In this respect, password strength is the most important consideration. Here's my definition of a stronger password: "one that takes longer to crack". Note that I didn't say "one that can't be cracked", for I don't believe a practical password that can't be cracked exists. But if you use harder-to-guess passwords then the attacker will have to start digging in their toolkit for other methods to get at your password than just typing in your name, your dog's name, your date of birth, your car registration number… I could go on. People really do use things like this as their password.

To give you a feel for what you're up against, let's suppose I am attacking your computer system remotely, and I need to get past a password that you have set.

The Attack: Phase One

The first thing I'm going to do is trawl the internet for anything I can find out about you, starting with Facebook, LinkedIn and other social networking sites, and moving on to credit reference agencies, government sites, registers of voters and anything else that will help me build as complete a picture as possible of YOU. If you're a blogger, so much the better – a goldmine of personal information can often be found in people's blogs. I will also, of course, be Googling you.

I'll be looking for anything that will tell me what your password might be, as a laser-targeted precision attack with one or two password attempts is much less likely to set system alarms off than an attack where I have to try one or two hundred passwords. Or one or two million. If I discover that you are a supporter of a football team, then I'll perhaps try the team's name, or maybe their nickname or the name of their ground.

However, if you're a really wealthy individual, I might be attacking *you* specifically, probably to get at your money. So I will add searches of news sites to my list, and also consider looking at company registration documents, officer and stockholder details and published accounts of any business I discover you're involved with.

If you're rich, I would probably be more inclined to invest more time and energy in the attack in other ways too, and maybe indulge in a spot of "Dumpster Diving". Basically this means I'll be going through your trash, looking for anything with information on it that might help me attack you. Most people these days are alert to the fact that having their old bank statements blowing around a landfill site is not a great idea. Even your average supermarket carries cheap cross-cut shredders, so the chances of finding useful documents like that still in one piece in your rubbish are dropping – but they're not zero, so I may strike it lucky. And besides, there may be other things in your trash that you haven't considered you needed to shred, but that could still give me useful information. Such as the tickets to the opera that you attended last week. They probably won't have your name on them, but I wonder if this week's password is "Madame_Butterfly" or something similar?

I might also try some social engineering on your secretary or perhaps other staff further down the food chain in your organisation – I'd probably avoid top executives' secretaries if possible, as they're likely to be clued up on social engineering techniques.

Social Engineering is a big topic in its own right, so I devote a later chapter to it. But for now, let's not stray too far from the topic at hand – I'm trying to crack your password. And let's assume that all of the passwords I've come up with using the methods I've tried so far were unsuccessful.

 Author: Stephen Gibbs

Extract from "Keeping Your Data Secure II: The Human Factor"

So, I'll move the game up a notch.

The Attack: Phase Two

I'm now going to have to resort to a little automation. I am still tippy-toeing around your system, as I don't want to risk setting off alerts that you might have built into your system if too many attempts to log in fail. But I also don't want to be sitting typing everything in by hand, as this next stage could take hours or even days.

Usually there is a time limit set on this type of alert, so as long as I don't try too many incorrect passwords an hour I won't set off the alarms. But I have to say, in my experience this sort of alarm is very rarely set at all, even though pretty much all network software worthy of the name is capable of generating one in circumstances where lots of different passwords are tried in quick succession. So, on most systems, I could try as many passwords as I liked for as long as I liked until I hit on the right one. However, I'm still being cautious, so I'm not going to resort to that method – yet.

What I'm going to do is compile a list of common passwords, literally passwords that are common to many people, all around the world.
Lists of such passwords exist, and it's not too hard to figure out why. Most people don't walk around with random password generators in their pockets, so when they're confronted with a situation that requires them to come up with a password, right now, quickly, they'll resort to using something easy for them to come up with on the spur of the moment, but hard for them to forget as well. Which is why in Phase One of this attack I tried your social security number, and all that other personal stuff that I found out about you on Facebook, as your password.

Anyway, a quick Google for "Lists of common passwords" will demonstrate to you that many people are extremely predictable when it comes to choosing a password. And some of the passwords commonly chosen are sexual swearwords or other offensive terms, so if you're curious and Google this yourself to see what words are commonly used as passwords, don't say I didn't warn you.

Armed with my list of common passwords I will write (or download) a small software program that will try them all, one after the other, at the speed I stipulate. I'll probably set it to try one every 10 minutes at first, and see if that gets your electronic attack dogs set on me. It probably won't, because as I said, these alarms are not usually switched on. And if I'm patient and cautious, and leave it set at just one attempt every 10 minutes, then I can still try the 1,000 most popular passwords in under a week.

If the first go doesn't crack your system password then I'd try the same words but with an initial capital letter, then with the number '1' added at the end, and then with both these variations at once.

Once again, let's assume that these methods are unsuccessful.

The Attack: Phase Three

This time the gloves are off. Next up, I'm going to use a Dictionary Attack. This is exactly what it sounds like – I'm going to attack your system with a dictionary. And no, that doesn't mean I'm going to bash your computer with a book – this is a remote attack, remember? No, what this means is I will now load an electronic dictionary into the software that I used before, and the software is going to try every single word in the dictionary, one by one, and see if it can find out your password that way. The English language has a great number of words in it, although even the experts can't agree how many. Let's just say that if I use a dictionary attack I'm going to need to try around 100,000 words to cover most of the possibilities. And at the speed of the last attack in Phase Two, that would take me two years.

Still, I've tried at least 1,000 passwords last week, and no alarms went off, so I'm going to assume I can safely try a faster rate of password attempts now. I'll increase the speed to 60 attempts a minute, one a second. In computing terms one second is an eternity, so this won't noticeably stretch the capabilities of the system I'm attacking. This is important, because if I throw too many attempts at the

 Author: Stephen Gibbs

system under attack, it might simply crash under the strain. And then even the laziest of system administrators would probably have a quick look in the system log files to see why it went down…

60 password attempts a minute might seem a lot to a human, but I'd expect a modern computer system to cope with that with ease.

So, trying 100,000 passwords at 60 attempts a minute means this next phase of the attack will take me at most just under 28 hours. I say at most, because, of course, the attack stops once I crack your password. And if it is a word found in a standard dictionary and it begins with the letter 'A' then I'll probably crack your system in under an hour!

However, if this phase of the attack is unsuccessful then I would once again try the same words, modified first of all by capitalising the first letter, then by adding a number '1' to the end of them, and finally by trying both of these modifications together. At worst this means roughly four and half days before I have to give up and bring out my biggest weapon.

As before, let's assume that these methods were unsuccessful. Although in the vast majority of attacks, I'd probably have cracked most people's passwords by now.

The Attack: Phase Four

The most powerful weapon in my armoury is a Brute Force attack, because it tries all possible passwords. One by one. For as long as it takes. I'm having to resort to using a Brute Force attack because I haven't been able to crack your password by any of the more clever means.

Passwords can be made up of any of the characters that can be typed using a standard keyboard. That gives these options:

- 26 lower-case letters, a – z
- 26 upper-case letters, A – Z
- 10 numerals, 0 – 9
- 38 special characters, \|<,>./?#~'@;:]}[{=+-_)(*&^%$£"!¬`¦…€

That's a total of 100 characters, very convenient for the maths that are coming up!

For every character in your password, I could try any of these 100 different characters, in any and every combination. That's a huge number of possible password permutations, and the more characters there are in your password, the bigger the number of permutations gets, by a factor of 100. Let me illustrate.

If your password is only one character long, then I'm going to guess it in at most 100 attempts.
If your password is two characters long, then I'm going to guess it in at most 100 x 100 attempts, or 10,000
If your password is three characters long, then I'm going to guess it in at most 100 x 100 x 100 attempts, or 1,000,000.
And so on.

A typical password is often 7 characters long, which means there are 100,000,000,000,000 possible character combinations. That's 100 billion. At 60 goes a minute, that would take my attack over 3 million years! So, I'm not going to stick to one attempt a second any longer. I've prodded your system for a few weeks now, and nothing has happened, so I am going to risk cranking the speed up, over a day or so to ensure that there are no sudden drops in the performance of your system that might alert you or some other system user what I'm up to.

There will still be plenty of symptoms of what I'm doing that could tip someone off that the system is under attack. For a start the processor is going to be busier, which means that your other software is going to run a little slower. Also, at times I might be using a fair portion of your network bandwidth, and

that might trigger alerts. In short, there are many ways that my attack might get spotted – but Brute Force attacks like this happen every day, and many of them are successful without ever being detected.

If I upped the number of attempts to 100 a second, then my brute force attack on your 7 character password would still take me nearly 32,000 years, so I'm only going to start at that speed and then gradually increase it to one million attempts every second. At that speed the crack would take, at most, a little over three years. However, if your password is 8 characters long, that goes up to 318 years!

Realistically, to go any faster I'd really have to have the computer system I'm attacking in my possession.

In fact, given enough time and no-one stopping me, a Brute Force attack is nearly always going to be successful. I say nearly always, because if you change your password regularly, say once a week, and I have a Brute Force attack in progress at the time, then your new password might be one of the passwords that my attack program has already tried. If I was to run it again I'd get in, but how am I ever going to know that? So I'd think my attack had failed.

Changing your password frequently and having a strong password are pretty much your only defences against a Brute Force attack, and even then it's not a guaranteed defence except in the circumstances I just described. So you need to set good alerts and to occasionally check your system activity levels and logs as well.

However, if I really, really wanted to get at your password then there are still other tricks I can try, but the risk level for me as the attacker now goes up with some of them. Options still available to me include hacking into an easier (more vulnerable) system on your network and attacking from there, perhaps by installing a network sniffer program, or installing malware of some description on your system, probably by sending you an e-mail with an attack payload and getting *you* to install it! Then there are the numerous social networking attacks I could try, and of course good, old-fashioned blackmail of one of your staff, if I can dig up some dirt on one of them. But, if all else failed, somehow I would have to get into your premises and install a key logger.

Strong Passwords

As you must have realised by now, the longer a password is, the harder it is for a Brute Force attack to work. But longer doesn't always mean harder to crack, as the dictionary attack will still get you if you've used a word found in a dictionary. And don't forget that a large number of people use one of the common passwords I mentioned earlier, or easily discovered information such as a pet's name
For your password to be considered strong, it needs to be longer than 7 characters, should not be a word or words found in a dictionary, and should use a mixture of upper and lower case letters, numbers and special characters. For example, "password" is dreadfully weak as a password, but "p@55W0rd" would be much tougher to crack.

What about ^Dyrbrhonnd@, which was created using key substitution – a simple technique which also makes it easy to remember but hard to crack. It's my name, Steve Gibbs, only with every key press moved one key to the right on the keyboard, and a couple of special characters thrown in for good measure. Or how about ~1Wlaac#, which was from Wordsworth's poem "I wandered lonely as a cloud". These techniques work well for creating strong passwords.

Multi-Factor Authentication is more secure than a password alone

The best security systems devised so far use three-factor authentication. The three factors are:

- something the person *knows*, such as a password and user name, or mother's maiden name, place of birth and so on;

- something the person *has*, such as a smart card that must be inserted before you can type in a password; and
- something the person *is*, determined using biometric data, captured using techniques such as a fingerprint or retinal scan.

Banks and other financial institutions have been some of the earliest adopters of new techniques to use more of these three factors, making modern online banking systems much more difficult to outwit. For example, devices such as one-time password generators are commonly issued to customers of online banking. The bank customers then have to use the device, with their bank card and PIN number, to generate a password each time they wish to do something potentially risky, such as adding a new payee to their on-line banking settings. This is using two factors, something a person has (their bank card and the card reader/password generator) and something the person knows (the PIN number for the card).

Another method that many organisations are starting to use to verify online payments is to send a text message to a customer's cell phone with a one-time password, and this too is better than accepting the card alone as evidence that you are the registered owner of the card – provided that you had already given them your mobile number. The hope here is that the text message will only go to the correct person, but of course this fails the test of stopping impersonation in all circumstances, as it is entirely possible for a card to have been stolen along with the owner's cell phone.

There is a bottom line to this discussion of passwords; there is no such thing as an unbreakable password, if the attacker is given time and access. You should do anything you can do to make your password harder to guess and more difficult to crack using a dictionary or brute force attack, or across a network.

Most attacks are fairly random and automated, so don't get paranoid and think that everyone has a hacker trying to get at them – they don't. But the odds are that you will be involved in a hacking incident or a data loss one way or another. Data about me personally has been lost twice in the last couple of years, that I know of anyway, once in the HMRC data loss and once by my Doctor. And, so far, no-one (other than me or my wife) has emptied my bank account…

To sum up this chapter:

- Make your password at least 8 characters long, and preferably 12 or more.
- Use a combination of all the different character types in your password; lower-case letters, upper-case letters, numerals and special characters.
- Don't use words from a dictionary, or other information that could be personal to you but easily discovered from, for example, social networking sites.
- Don't use the same password for every system or website that you log into. It only takes one of these systems to be compromised and your password is then "out there".
- Change passwords reasonably regularly, and don't re-use passwords for at least a year or two.
- Don't write them down, ever. But if you must, for goodness' sake write them down somewhere secure – not on a post-it note stuck to your monitor!
- Don't share your password(s) with anyone.

