KnowBe4
Human error. Conquered.

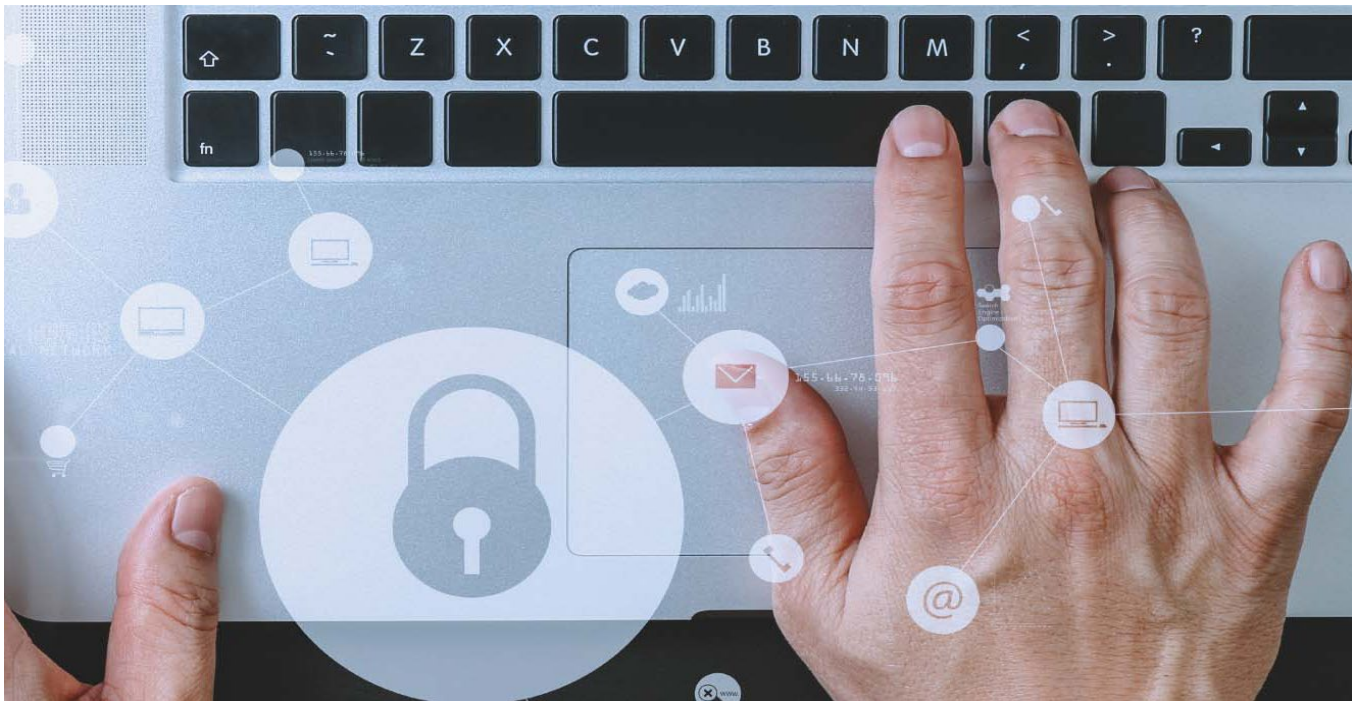
What Your Password Policy Should Be

Table of Contents

INTRODUCTION

Passwords are part of every organization's security risk profile that should be taken seriously by IT security professionals like you. Just one weak password with access to an organization's critical systems can cause a breach, take down a network or worse. Whether we like it or not, passwords are here to stay as a form of authentication for at least another decade or so. There is no time like the present to review your organization's password policy and update it for the betterment of your organization's overall security.

This whitepaper will cover the critical components of a recommended password policy for any organization around the world. It will also summarize the advantages and disadvantages of using passwords, provide a detailed breakdown of the various types of passwords attacks and defenses, as well as KnowBe4 tools that you can utilize to implement better passwords throughout your organization.



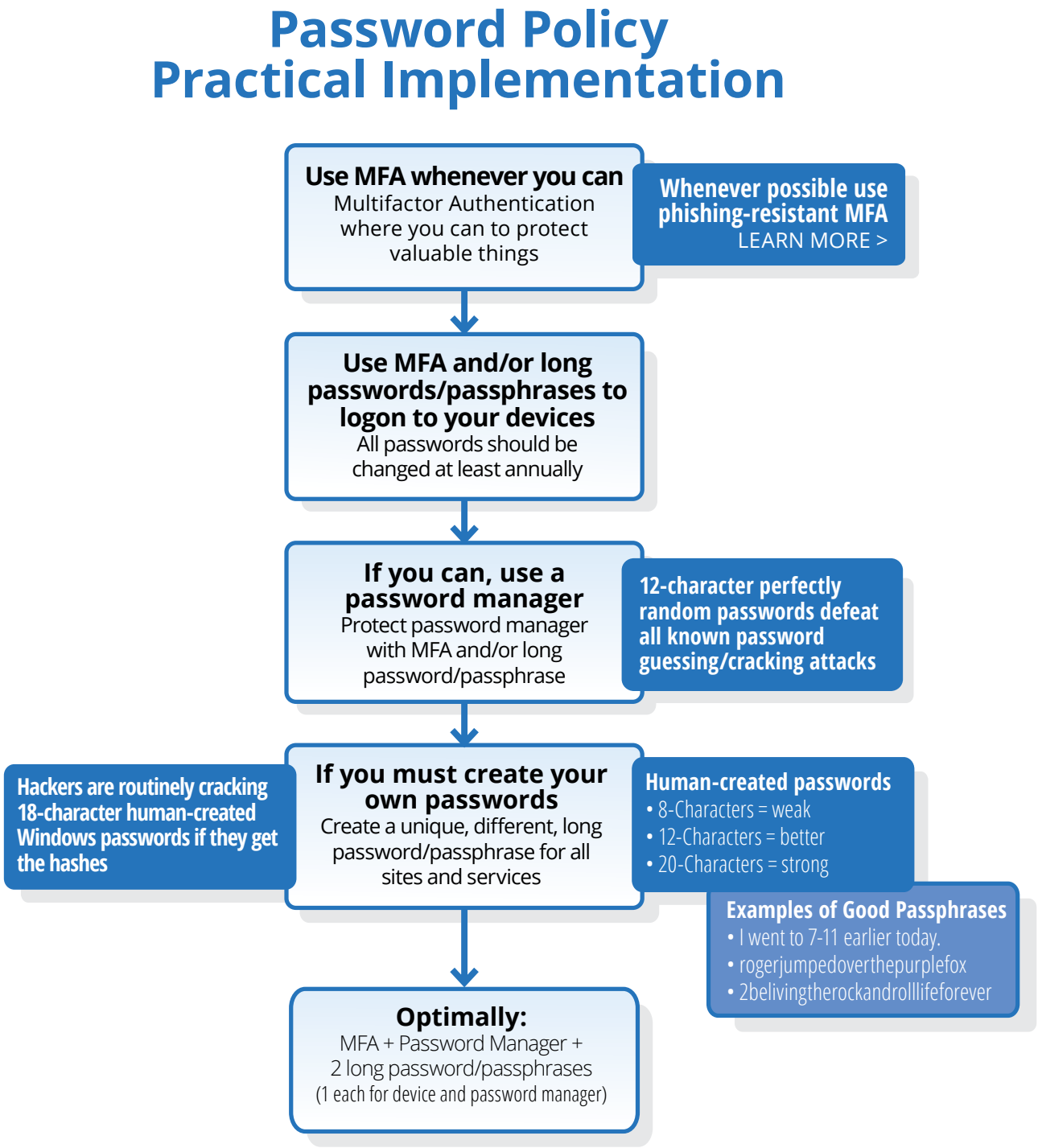
RECOMMENDED PASSWORD POLICY SUMMARY

Your personal and enterprise passwords and policies should follow these recommendations.

The reasons your passwords and password policies should follow these recommendations are due to the methods attackers commonly use to compromise passwords and the defenses it takes to mitigate those threats.

This whitepaper will discuss the various password attacks that justify the recommended password policies. If you do not have time to read this entire whitepaper, just implement the recommended password policy to the right. But if you read this whitepaper, you will gain a good understanding of why these specific policies are recommended and why you should follow them (or not, if you decide to make a different risk decision).

Passwords and password policy comes down to risk acceptance and individual risk decisions. This whitepaper will discuss the involved issues and let you make your own personal and professional risk decisions as they apply to passwords and password policy. This whitepaper will make you a far better-informed password and password policy implementer.



BUT AREN'T PASSWORDS GOING AWAY SOON?

The "end of passwords is coming soon" prediction is a frequent theme in cybersecurity that never seems to arrive. If you look at the sheer amount of passwords that the average person still uses today, a passwordless world is either never going to happen or at the very least, likely to be a decade or more off. This is despite nearly universal agreement among cybersecurity experts that passwords are too easy to compromise and that something better needs to replace them. Today's conventional wisdom is that multi-factor authentication (MFA) needs to be used wherever possible to better secure logins until something more advanced, like "zero-trust" security architectures, can be implemented at scale to replace passwords forever.

Note: To understand more about zero trust, read NIST's excellent Zero Trust Architecture primer: https://csrc.nist.gov/publications/detail/sp/800-207/final.

But none of the technologies touted as being better password replacements are widely supported. MFA is the most common suggested replacement, but is currently supported on less than 2% of the world's websites and services. Conversely, passwords are still very popular. Sites and services supporting them are everywhere. Most websites and services ONLY support passwords for login authentication. Everyone is used to using passwords and they have many of them.

Although the number of passwords that the average person has varies according to the study or survey, most say the average person has from seven (https://www.answers.com/Q/Average_number_of_passwords_per_person) to 191 (https://www.securitymagazine.com/articles/88475-average-business-user-has-191-passwords) passwords. Some of the discrepancy between the lower and higher estimates likely has to do with how the average number of passwords statistic is defined. Most people have a far smaller number of unique passwords that they use across a far larger number of websites. Most studies show that the average person has from three to 19 different passwords that they distribute across at least 150 unrelated websites and services.

Far from being a passwordless society, today, most of us use a combination of multiple, unrelated MFA solutions and multiple passwords to authenticate to various sites and services. The average user often has one or more MFA solutions (e.g., for work, banking, stocks, social media sites, etc.), plus a whole lot of passwords all to manage at the same time. So, until something changes drastically in the digital world, most of us are going to be stuck with a whole bunch of passwords to create, use and manage.

What About Other Types of Passwords?

Most sources talking about password issues are only discussing user-based passwords. There are many other uses of passwords. Passwords are used by most operating systems (OS) to run built-in and critical services, daemons and processes. The operating system is only running because of a bunch of key accounts (e.g., on Microsoft Windows, they include LocalSystem, NetworkService, LocalService, etc.), all which only work because of system-generated passwords. Most applications run in the security context of a security account that uses passwords; passwords that can be compromised and exploited. Trying to remove passwords out of operating systems would require a fundamental redesign of all the involved operating systems (e.g., Microsoft Windows, Apple iOS, Linux, Android, etc.), a redesign so radical that most existing applications and services would have to be re-written from the ground up. The moment that the OS is redesigned to get rid of all passwords, nearly every existing application would cease to work on those operating systems.

On many networks, the computers that use them must have a computer account, and that account has a login name and password. On a Microsoft Active Directory network, every computer joining the network has a login name and password, which is created when the computer "joins" the network and is used every time the computer "logs in" to the network or is asked to periodically re-authenticate.

Most Wi-Fi wireless networks require an SSID (Service Set Identifier) password to join, and hundreds of attacker tools exist to sniff, hack and guess those passwords. Many MFA solutions, like Microsoft's Windows Hello, require a password (or PIN) before you can use the MFA solution to log in. Many MFA solutions, like using smartcards on Microsoft Active Directory networks, create, store and use passwords (actually, password hashes) behind the scenes. Many compromised environments, which switched to MFA in order to get promised greater security, were shocked to learn that passwords (or hashes) used "behind the scenes" were still able to be compromised by an attacker to exploit their environment. Your cable modem likely uses a password to connect to your Internet service provider. The firmware or software that your cable modem (and Wi-Fi router, etc.) likely has user accounts and passwords simply to function.

This is to say, that there are billions of passwords used by all sorts of things beyond just the traditional user logins that most regular people do not know about or think about. Although non-user login accounts may operate differently and be somehow restricted in use, all of their passwords can be stolen and hacked just like any other password. And, currently, it is very hard and nearly impossible to operate them without passwords.

Many security experts know that these types of non-user passwords exist and think about how to better secure them or replace them. But getting rid of passwords on these types of devices is even harder than getting rid of user-based passwords, and so far, getting rid of user-based passwords has proven to be impossible even after three decades. Getting rid of non-user passwords would be exponentially harder. The difficulty of the issue is that even if you got rid of every user-based password, there would be dozens to hundreds of passwords used by the things users used. And any of those passwords could be compromised to exploit the user.

Most security experts know it would be very hard to get rid of those other types of passwords and would be happy if we could just replace user passwords because that is where most of the existing risk is right now. But there is no guarantee that if user passwords are completely replaced that attackers will not just start trying to compromise all the other types of passwords with more vigor. And if those non-user passwords are compromised, any users using those devices or networks can be exploited just the same. All the effort to get rid of user-based passwords is just part of a bigger problem.

This is not to say that there is no benefit in trying to better secure user passwords. Currently, user passwords are one of the biggest risks in computer security. If we solve the problem of password insecurity (or move to something else, like MFA or zero trust), it removes a huge percentage of current cybercrime risk.

This document is going to cover the various attacks against passwords and what your defenses should be (personally, professionally and enterprise-wide) to significantly reduce the risk of compromise due to password use and management. It will show you the "WHY" behind why particular password policies are recommended. It will end by listing useful password auditing tools offered by KnowBe4. Any reader following the recommended password policies and using the related tools will be far better protected against password abuse until the day a truly passwordless society finally emerges.

THE GOOD AND BAD OF PASSWORDS

In the rush for the world to eradicate passwords, it is easy to forget that passwords are easy to use, cheap and used all over the world for a reason. This section of the paper will quickly discuss the good, and the bad, of passwords.

The good things about passwords include:

- Fairly ubiquitous
- Easy to implement
- Cheap to implement, operate and support (web portals)
- Easy to change, if needed
- Easy to remember
- Can be used securely

If users create and use passwords in a way that minimizes hacking attacks and if the systems that store them protect them so they are not accessed or guessed by attackers, they can be a secure way to log in. Passwords are fairly easy to use. Everyone quickly learns how to use them. Two-year-olds to centenarians can use them. There is almost no one who cannot learn to use passwords to log in.

Contrast that with MFA solutions. MFA solutions are always more expensive and difficult to use (no matter what MFA vendors might say). A good representative presentation to see or hear about MFA training challenges is "Why Johnny Can't Use 2FA and How We Can Change That" (www.rsaconference. com/industry-topics/presentation/studies-of-2fa-why-johhny-cant-use-2fa-and-how-we-can-changethat).

You can download the slide deck here: https://published-prd.lanyonevents.com/published/rsaus19/sessionsFiles/13259/IDY-T07-Studies-of-2FA-Why-Johhny-Can%E2%80%99t-Use-2FA-and-How-We-Can-Change-That.pdf

The talk was related to a university study looking to see how well regular end users could understand and use instructions for basic MFA solutions. In their study, the researchers gave the participants a popular and simple MFA device. The users had to plug the device into their computer's USB port and press one button to activate and use the device to log in. Each user was given instructions on how to register and use their device. Plug in something to a USB port and push a button. How hard could that be? The failure rate was over 70 percent—a 70 percent failure rate for something that simple.

But there are reasons why nearly all cybersecurity experts agree that passwords need to be replaced with something better. Passwords are also fairly easy to steal, hack and guess. Why? Here are some of the reasons:

- It is difficult for humans to create strong passwords that withstand common password attacks
- Newly created passwords are easy to forget
- Passwords can easily be reused across multiple, unrelated websites and services
- Passwords are easy to share with other people
- Passwords are difficult to manage when you have many

- Passwords can be easy to guess if not enough complexity is required and/or if the login portal does not limit the amount of guesses
- And, of course, the worst problem is that passwords are easy to steal

The next section of this paper, "Password Attacks" will be about this last problem. But if you want to know why passwords are still hanging in as the number one method of login authentication despite the obvious problems, it is because passwords are easy to use, cheap to implement and are widely supported. So far, no other potential replacement can claim all three characteristics at the same time. As long as that remains true, passwords will still be with us. For that reason, everyone needs to understand what attacks could possibly be successful against their password.

PASSWORD ATTACKS

There are many ways to compromise passwords. In general, the various password attacks can be summarized as:

- Password theft
- Password guessing
- Password hash theft and cracking
- Unauthorized password reset or bypass

These methods are used millions of times a year by various threat actors. The methods can be used by a single threat actor to obtain one or more passwords for their own uses, for reselling, or to populate larger login credential lists that can be used, viewed or sold. Today, there are many existing "password dump" stolen password lists available on the dark web and Internet that contain millions to billions of previously compromised passwords. Here are some example articles discussing very large password dump lists available for sale and/or viewing:

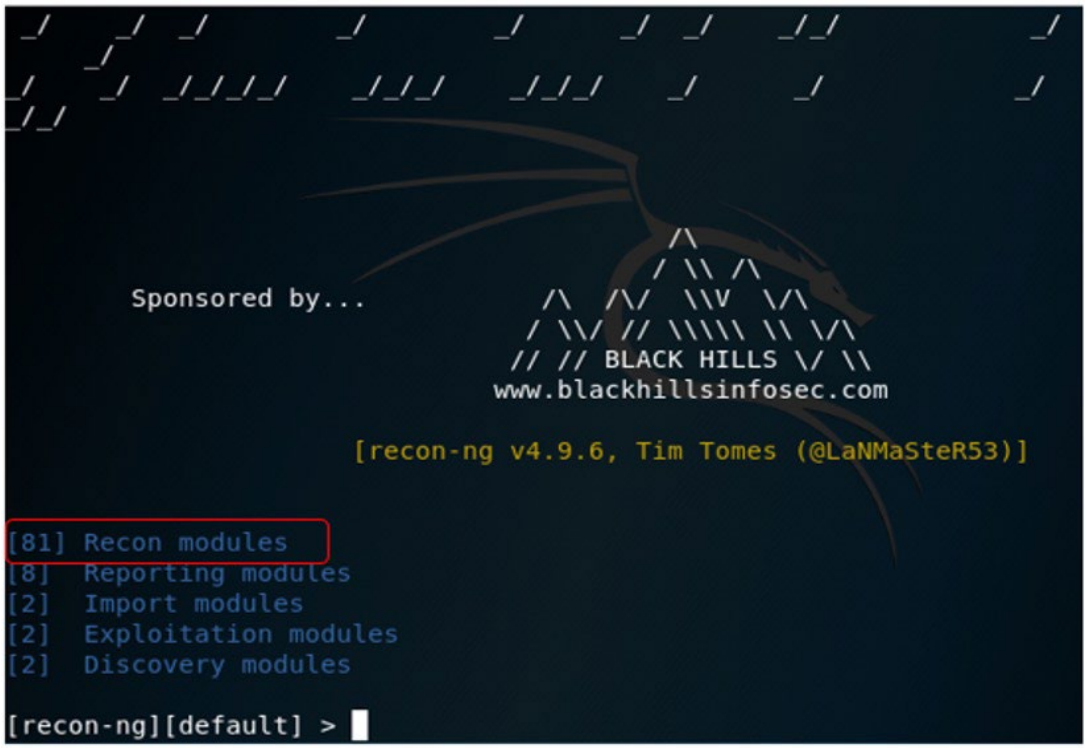
- https://noqreport.com/2021/06/08/8-4-billion-passwords-hacked-leaked-online-check-to-seeif-yours-is-among-them/
- https://www.welivesecurity.com/2020/07/09/billions-stolen-passwords-sale-dark-web/
- https://www.forbes.com/sites/daveywinder/2019/02/01/2-2-billion-accounts-found-in-biggestever-data-d/?sh=608bd0cf2eb6
- https://securityledger.com/2019/01/four-more-collections-700-million-stolen-passwords-discovered/
- https://threatpost.com/billions-passwords-cyber-underground/163738/

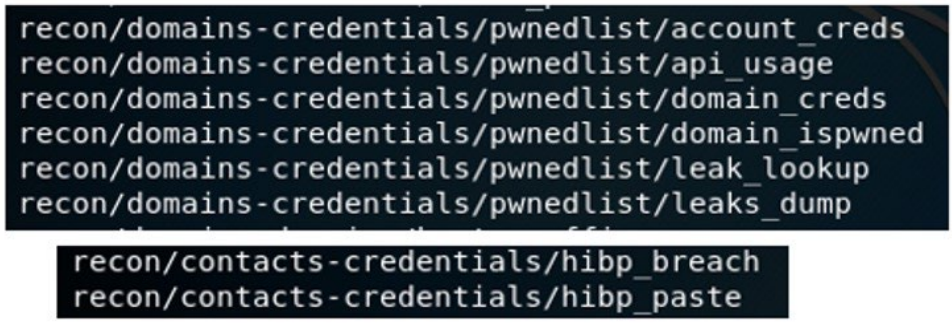
With tens of billions of login passwords available for viewing or buying, there is a growing chance that over time, nearly everyone will have a login credential stolen and placed on one of these password dump lists.

Attackers and researchers can view or purchase these password dump lists as separate files, and there is even a wide variety of free and commercial software tools that allow querying of one or more of these known password dump files if the attacker is looking for specific login credentials belonging to a particular target organization or individual. The attacker can simply put in the target's domain name (e.g., knowbe4.com) or individual login name (e.g., rogerg@knowbe4.com) and find out if that target's login credentials exist on one or more password dump lists.

Below is an example screenshot of a popular open-source tool, recon-ng (https://github.com/

lanmaster53/recon-ng). Anyone can download recon-ng and use its “Recon modules” to query one or more known password dump lists. A user simply types in the target domain name or login name, and then uses one or more recon modules to query known password dumps.





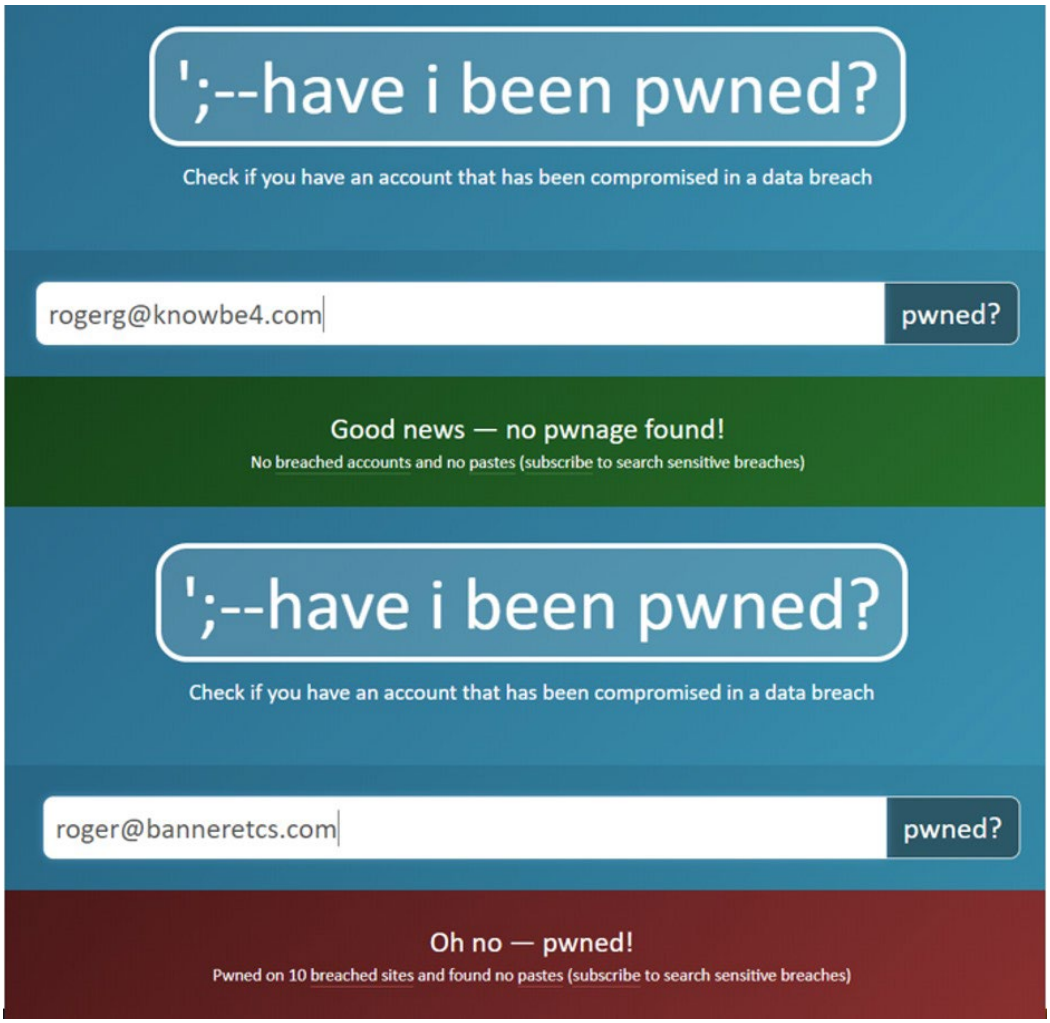
It will search the password dump files and bring back any matches to the submitted query. There are dozens of tools just like this on the Internet, both free and commercial. There are many commercial services, used more by legitimate users than hackers, but potentially by anyone who has the money to use them, that will bring back not only every possible compromised password a person has had stolen, but lots of other information, including what websites they are registered on, what applications they have installed and many times, physical locations, including home and work. These commercial services know a lot about you and make obtaining that information fairly easy.

They are many ways to find out if one of your login credentials has been stolen in a publicly known breach or are located in a known password dump list, including these links below:

- https://www.knowbe4.com/breached-password-test
- https://haveibeenpwned.com/
- https://sec.hpi.de/ilc/

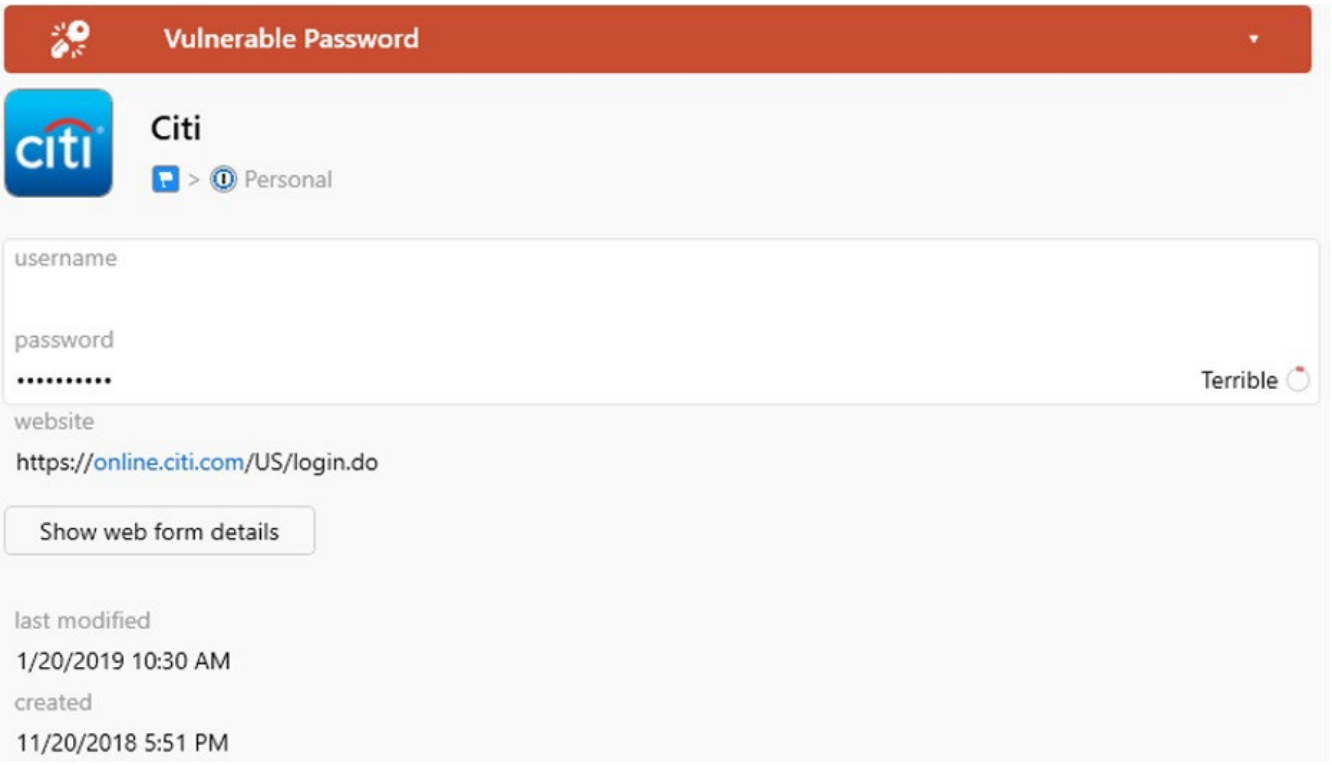
The first link is KnowBe4's free Breached Password Test tool. It is a great tool for running various breached password queries against all the passwords in your environment all at once. The other two links are free services, which will allow you to easily look for the breach status of individual login accounts, one at a time.

The figure below shows two login credential checks using the very popular HaveIBeenPwned (https:// haveibeenpwned.com/) created by Microsoft's Troy Hunt. The first example check shows a return for an email account that is not associated with any breaches. The second example is for a return showing 10 different breaches associated with another login ID.



What is important to note, although it is not explained by the results, is that in this specific example of tested login credentials, none of the breached passwords were due to a mistake of the account holder. All ten of the password breaches were due to compromises of legitimate websites (e.g., Facebook, Adobe.com, etc.) the account holder had registered on to get very popular services. Websites get breached all the time. That is why it is important for everyone to periodically change their passwords, and immediately if it is involved in a known breach.

Some password managers, which allow users to easily create, use and manage different, unique, strong passwords for each website and service, will notify a user if a login account they are using for a website or service has been compromised. The figure below shows an example "password was compromised" warning from a popular password manager.



All stolen passwords were at some point, uncompromised, and only known by the user and the involved authentication system (and possibly other authorized parties), but somehow subsequently became compromised by unauthorized attackers. The next subsection covers how those passwords are originally stolen.

Password Theft

The theft of passwords from victims can occur many ways, including:

- In-person
- Social engineering
- Hackers or malware on the endpoint
- Network eavesdropping
- Stolen credential databases
- Visible in publicly accessible code

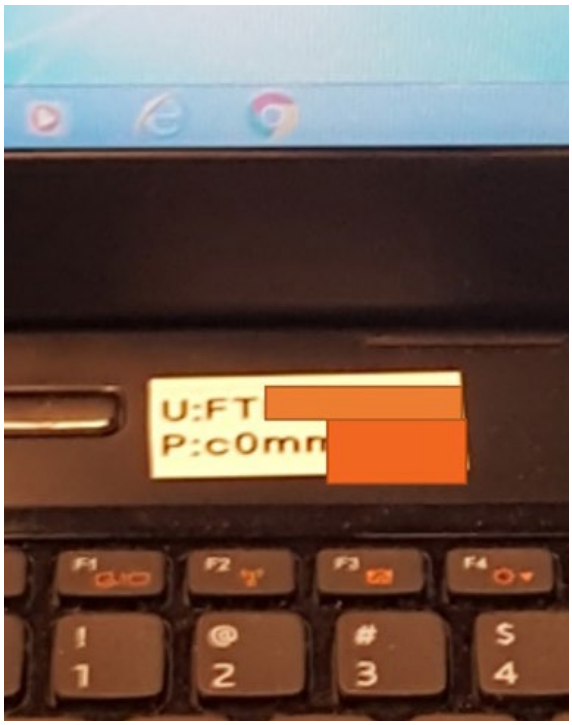
Each method will be discussed in more detail below.

In-Person

Although not used by attackers for large scale theft, individual passwords can be stolen by an in-person attacker. One of the most common methods is an attacker visibly looking for and recording someone's password as the victim types it in. This is known as shoulder surfing. It is especially easy to do when shorter four-digit personal identification numbers (PINs) are used, but can often easily be captured even when long and complex passwords are used. There are videos

on the Internet showing people (e.g., celebrities, Congressmen, etc., typing in very simple PIN codes on their cell phone (e.g., 7-7-7-7, 1-2-3-4, etc.) for the whole world to see. There are videos showing people using their cell phone to record someone's password as they type it, which is then later re-typed by the attacker.

Previously trusted insiders are known for shoulder surfing their coworker's passwords and using them to log in as that coworker in an attempt to hide the attacker's involvement in a malicious scheme. Shoulder surfed passwords are also used by previous employees to log in to their previous employer's systems after they have been separated from employment. When the malicious activity is finally discovered, the rogue activities will be initially traced back to the account of an otherwise innocent employee. The innocent employee has to prove their innocence to get investigators to look for the real culprit.



As counterproductive as this may seem, passwords are often posted in plain view of anyone who looks. I think nearly every reader has seen a television news report where the reporter is interviewing someone at a busy office and a password shared by staff is obviously written on a whiteboard where it is then viewed by everyone. The lead author of this document has seen passwords visible on desks and boards while he checked in at the front desk of a hotel. Most IT people will tell you that they have seen passwords written down near someone's desktop computer or on the proverbial Post-It™ note under or on someone's keyboard or laptop. The figure to the left is a picture of such a real-world circumstance, where login credentials are clearly visible taped on a laptop.

Many passwords are learned by others simply by asking. Many people have told trusted others (e.g., coworkers, spouses, friends, employees, etc.) their private passwords for various reasons only to have that trust proven misplaced. You should never give another person your password.

You would be surprised how many people will give a complete stranger their private passwords. Several studies have been done where people "on the street" were asked by researchers

for their real passwords in exchange for a small gift (e.g., candy bar, cheap pen, etc.). A surprisingly high percentage of people are willing to give their passwords to strangers, with most studies

showing up to 40% of people are willing to give their real passwords to researchers in these studies. If you find this fact hard to believe, you do not know human behavior as well as you think you do.

Long-time U.S. talk show host, Jimmy Kimmel, created a reoccurring video segment where people in his production team ask random strangers for their real passwords while standing on a street in Hollywood, CA. You can search for them by typing in "Jimmy Kimmel password video." Here is one:

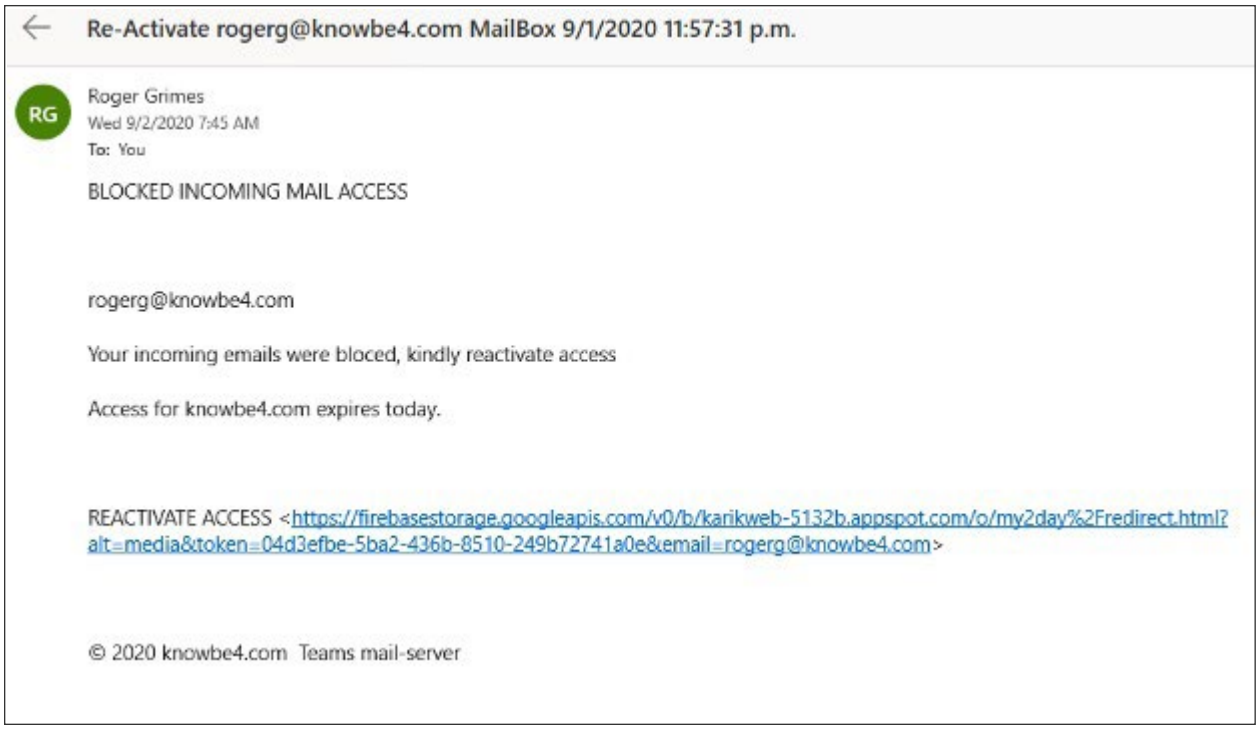
https://www.youtube.com/watch?v=opRMrEfAIiI. After viewing it, most IT people are trying to figure out if they are crying from laughing or sadness of how easy it was to get people to reveal their passwords.

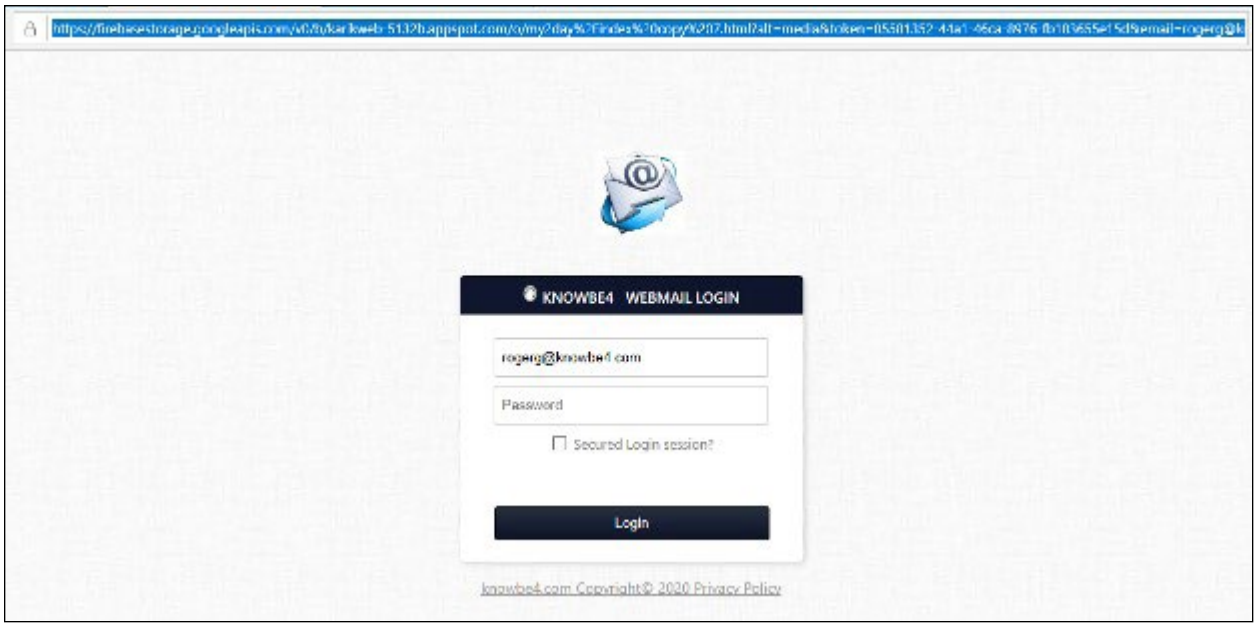
Social Engineering

Easily the most common way passwords are compromised, especially at large scales, is social engineering. Social engineering is responsible for the vast majority of malicious hacker and malware attacks and have been since the beginning of computers. Here are some relevant links to back up that claim:

- https://blog.knowbe4.com/90-of-all-cyber-attacks-on-organizations-involve-social-engineering
- https://blog.knowbe4.com/phishing-remains-the-most-common-form-of-attack
- https://info.knowbe4.com/threat-intelligence-to-build-your-data-driven-defense
- https://blog.knowbe4.com/70-to-90-of-all-malicious-breaches-are-due-to-social-engineeringand-phishing-attacks

Social engineering attacks most commonly arrive as emails (known as phishing), but can arrive using other communication mediums such as websites, social media, SMS (Short Message Service) and voice calls. In most cases, the social engineering messages trick a user into providing their login credentials by providing a URL (Uniform Resource Locator) link, which takes the potential victim to a bogus, but realistic-looking, login portal. If the user provides login credentials, they are captured and stored by the attacker. A growing number of phishing attacks now actively confirm if the login credentials the user is typing in are valid before storing the captured password credentials. Below are some common examples of phishing messages and related login screens trying to capture a potential victim's passwords:





facebook

Dear Facebook user,

In an effort to make your online experience safer and more enjoyable, Facebook will be implementing a new login system that will affect all Facebook users. These changes will offer new features and increased account security.
Before you are able to use the new login system, you will be required to update your account.
Click here to update your account online now.

If you have any questions, reference our New User Guide.

Thanks,
The Facebook Team

Update your Facebook account

Update

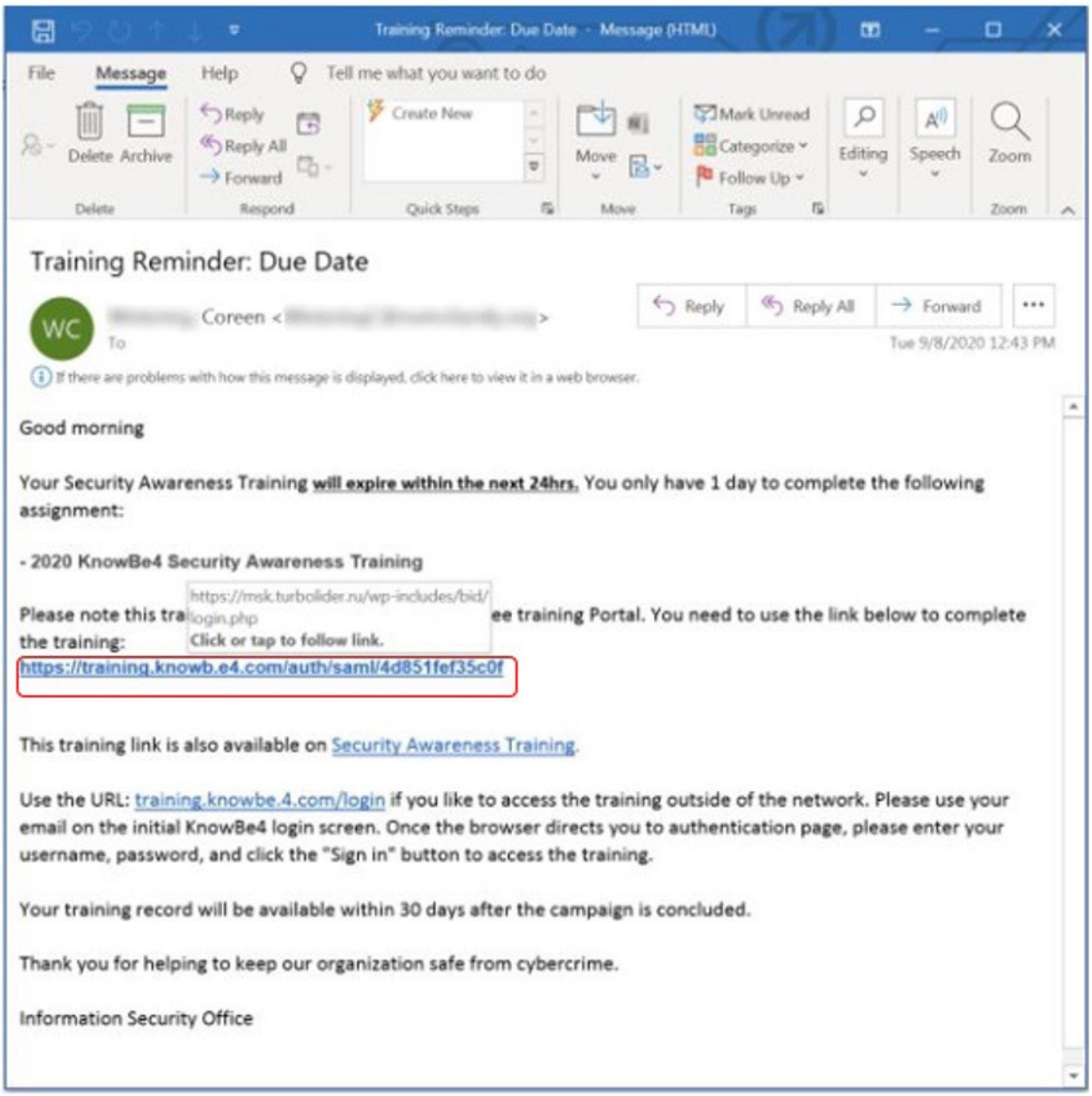
Subject: Account suspended!

LinkedIn

Your LinkedIn accaunt was suspended due to spam messages. To unlock your account open this link www.llinked.ni a

Thank you for using LinkedIn!

The LinkedIn Team



Social engineering attacks can be prevented by using a good combination of policies, technical defenses and security awareness training. This will be covered in more detail in the “Other Supporting Policies” section below.

Hackers or Malware on the Endpoint

A common IT security saying is that if an attacker has control of a device (either through physical access, remote access or malware), it is not the victim’s device anymore. It is very true. When a device is “owned” (or “pwned”) by an attacker, they can do anything to the device that the compromised victim could do, as allowed by the combination of hardware, operating system, software and currently exploited security context. At the very least, they can record (i.e., “keylog”) passwords as victims type them in. They can also search for passwords in memory, stored in software (such as browsers) and on disk. There are hundreds if not thousands of hacker and malware programs which can be used to steal passwords.

Many, if not most, malware programs aggressively look for and steal passwords on the devices where they are active. One of the most common password-stealing malware programs is a bot known as Trickbot. It first arrived in 2016 and quickly turned into the most used program and re-

used code by professional cybercriminals to steal passwords. Many other malware programs and cybercriminal gangs either "drop" (i.e., use) the whole Trickbot trojan to steal passwords or use its programming code (released publicly after a few years) to steal and extract passwords using their own custom malware and password stealing tools.

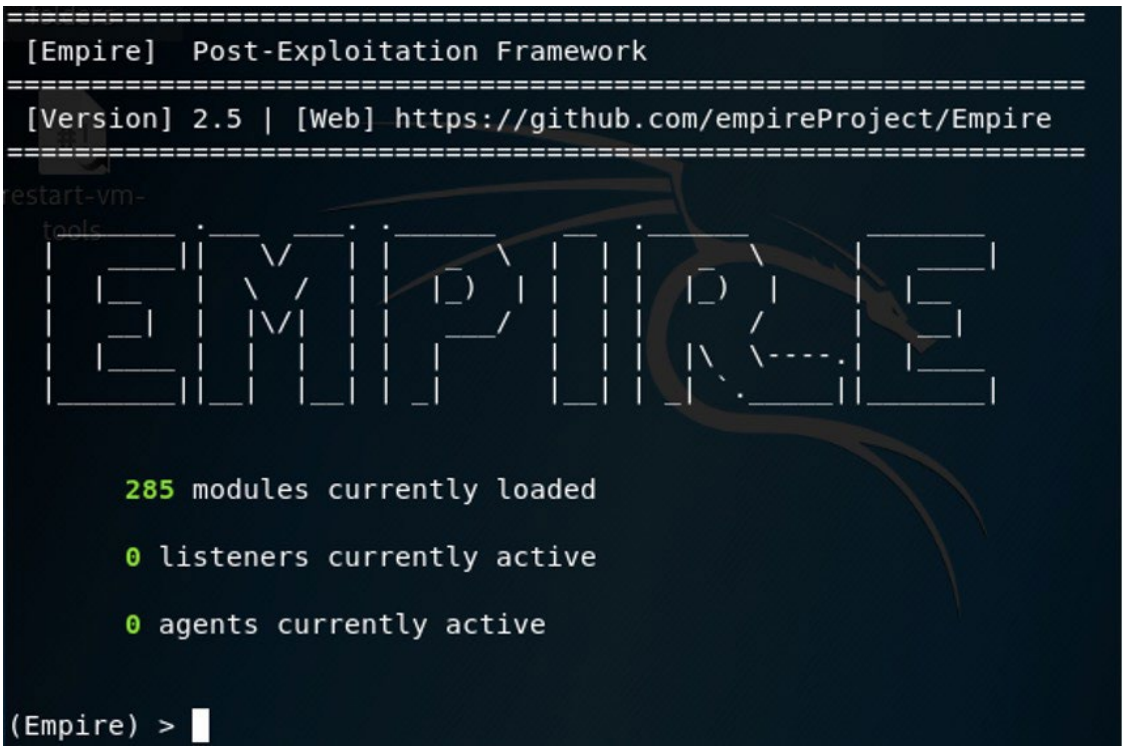
For more information on Trickbot:

- https://us-cert.cisa.gov/ncas/alerts/aa21-076a
- https://us-cert.cisa.gov/sites/default/files/publications/TrickBot_Fact_Sheet_508.pdf
- https://www.malwarebytes.com/trickbot
- https://www.cisa.gov/uscert/ncas/alerts/aa21-076a

There are many popular hacking tools that can be used to look for and extract passwords on compromised devices. The most commonly used password-stealing programs include:

- Mimikatz
- Empire
- Metasploit

The figure below is an example of the Empire hacking tool.



Empire is a set of Microsoft PowerShell™ scripts developed by legit researchers supposedly as a way to educate defenders about how easy it was to use PowerShell and other hard-to-detect malicious scripts to attack and compromise IT environments. It contained almost 300 different modules that could be used against Microsoft Windows, Apple OS and Linux environments. It was deprecated in 2019 by its developers supposedly because they believed that their education goal was met. Either way, by then, the vast majority of hackers and attackers either used Empire to do their illegal, malicious deeds or used it as a model to develop their own malicious scripts. Even with Empire "retiring," there are a plethora of freely available hacking tools which can be used to

steal passwords on compromised devices.

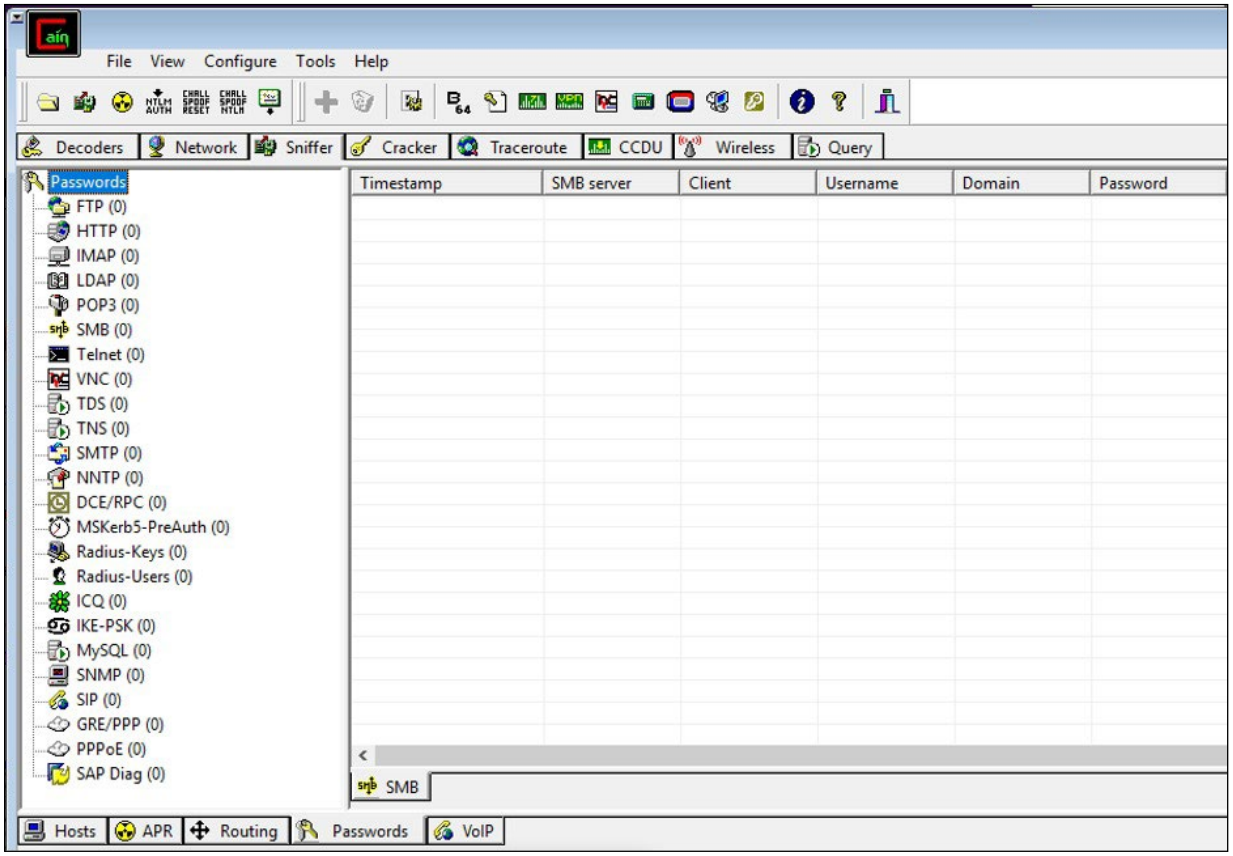
Suffice to say, if malware or malicious hackers are discovered on a device, consider all passwords used or stored on that device during the time of the breach to be stolen and in the hands of malicious hackers.

Network Eavesdropping

Similar to malicious hackers or malware on individual endpoints, malicious hackers or malware installed on or eavesdropping on a network can steal passwords, password hashes and related authentication negotiations. A common scenario is an attacker "sniffing" (e.g., capturing, listening) on an unprotected or insecure Wi-Fi network, who is able to record passwords, hashes or other authentication information, as it passes over the network between endpoints. In most cases, the attacker must have first successfully implemented a "man-in-the-middle" (MitM) attack, where the attacker's listening device is able to capture network traffic headed between targeted endpoints.

There are many hacker tools for eavesdropping on networks and stealing passwords. One of the earliest and most popular network password sniffing tools was Cain & Abel (shown below).

Note: You absolutely should not attempt to download or use Cain & Abel today because it has not been supported since the 1990's and is usually filled with malicious software by the people and sites offering it for download

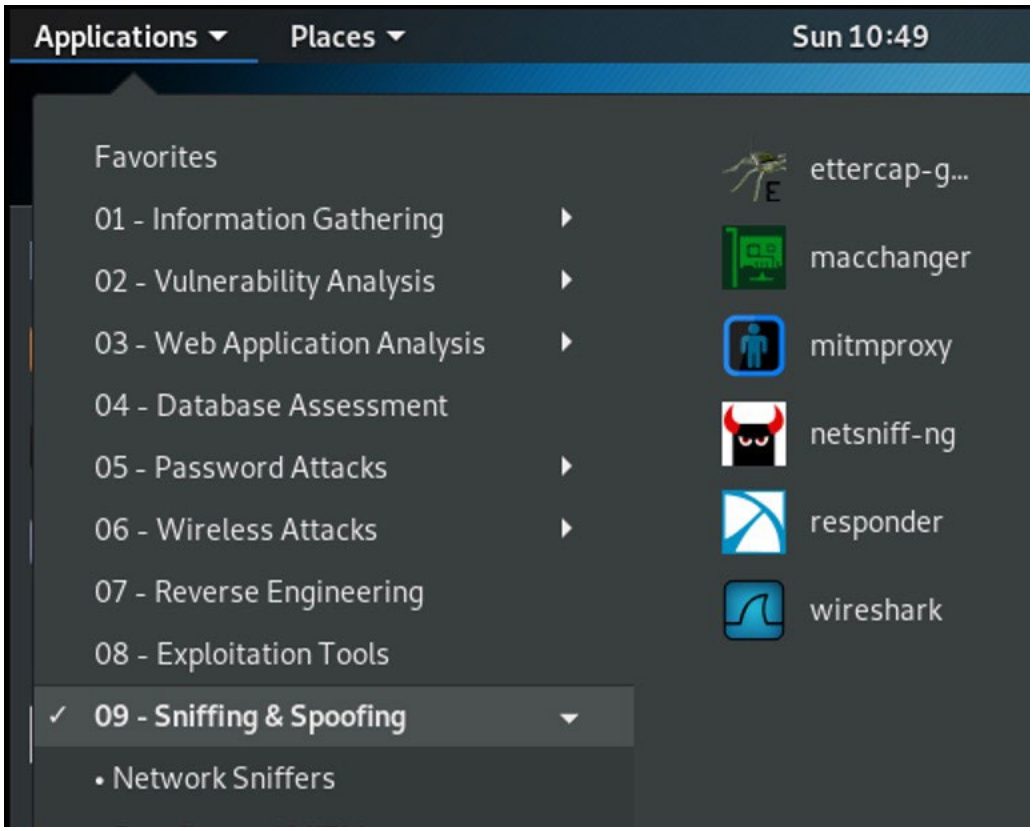


Cain & Abel is an old hacker tool and is included here simply for demonstrative purposes only. But you can see that it allowed users to listen in and collect over two dozen different types of network protocols and intercept any included passwords captured in unprotected network streams. Many malicious hackers in the 1990's used to connect to targeted networks, execute

Cain & Abel, wait a few minutes and end up with dozens of captured passwords. Another popular Firefox browser extension for capturing network passwords was Firesheep (https://en.wikipedia.org/wiki/Firesheep), released in 2010. It allowed anyone downloading and using it to sniff for login credentials on Wi-Fi connections.

The extensive use and demonstration of how easy it was to sniff passwords off of networks led to most network authentication connections being protected against easy eavesdropping by the use of the HTTPS (Hypertext Transfer Protocol Secure) protocol. HTTPS uses Transport Layer Security (TLS) and asymmetric and symmetric cryptography to secure communications between protected endpoints. The widespread use of HTTPS and other secured network streams made it harder to sniff passwords off of networks.

Still, if network traffic can be "MitM'd" or a user can be tricked into visiting a malicious website, there is a good chance that passwords and/or hashes can be captured or determined by an attacker. There are dozens to hundreds of hacker tools that can be used to sniff password or password hashes off of compromised network connections. Today, many penetration testers use specialized Linux distributions, such as Kali Linux (https://www.kali.org/). As shown below, Kali Linux comes with a handful of tools built for sniffing on networks, and in particular, for capturing passwords.



Wireshark is a very popular network sniffer, although it is not built directly for looking just for passwords. MitMproxy is a tool designed for creating MitM attacks, necessary for capturing passwords from network streams. Ettercap is a network sniffing tool with many features explicitly designed to make capturing network-passed passwords easier. Responder is a hacking tool that allows someone to set up "fake" or MitM websites from which passwords can be captured. There are dozens of other

network sniffing tools which are freely available on the Internet that anyone can use.

There is an excellent example video on this by Kevin Mitnick, the world's most famous hacker and KnowBe4's Chief Hacking Officer: https://www.youtube.com/watch?v=xaOX8DS-Cto. In the video, Kevin uses a simple phishing email to trick a potential "victim" (in his demo, he is both the victim and the attacker) to create a MitM situation using the software tool, Evilginx, where the victim's password credentials are captured. More on this demonstration attack in the "Password Hash Theft and Cracking" section below.

Stolen Credential Databases

One of the primary ways that attacks get passwords (or password hashes) is the theft of authentication databases. This can be done against a lone device or against a multi-user database, website or service. Most operating systems store user passwords or hashes in files within the operating system. For example, Microsoft Windows stores local login credentials in a set of files known as SAM (Security Accounts Manager). Login passwords for Active Directory networks are stored on domain controllers in a file called NTDS.DIT. Login credentials for Apple, Linux, BSD operating systems are stored in other files often known as passwd and shadow. Applications requiring password authentication can use other existing authentication providers (like Active Directory or AD Azure) or may use their own authentication credential databases. No matter where those files are stored, if those files can be copied or accessed, the passwords or hashes they contain can be hacked and exploited.

The most popular type of password attack, by far, is password theft using social engineering, and it likely results in millions of stolen passwords each day.

It is very common for attackers who have gained access to a network to immediately download every login credential they can find as one of their first steps. For example, most attackers of Active Directory networks will attempt to go from one compromised Windows device to "owning" the entire network by downloading every login credential and hash from a domain controller as soon as they have the security credentials necessary to access the domain controller NTDS.DIT files (e.g., Enterprise Admin, Domain Admin, local Administrator or System on a domain controller). It is routine to hear stories of sophisticated attackers going from a single compromised network device to having every password of every user and computer in a network within a few hours (or even minutes) of the compromise of the first network device.

Billions of passwords have been stolen from compromised websites and services. It is almost rare to hear of a popular website or service which has not been compromised by hackers and their passwords or hash databases stolen. When that happens, every registered user's login credentials are now in the hands of an attacker. We likely do not hear about the majority of compromised login credential databases because the websites and services themselves do not know they have been breached.

Because most users reuse the same password (or password pattern) across multiple, unrelated websites and services, the compromise of one shared password can more easily lead to additional websites and services used by the same user. For example, a malicious hacker compromises a website where a user bought clothing for their dog; and then tries the newly stolen credentials (using the same login name) at other, more popular websites, like Amazon, Facebook, Instagram or the user's work account. Many compromises of work accounts happened because the user reused the same login credentials or passwords between their personal websites and services and work. And it is very hard to enforce that a user is not reusing the same password between work and personal websites.

Visible in Publicly Accessible Code

It has long been a big problem that login credentials, including passwords, are often left in programming code viewable on publicly-accessible sites and services; or in "beta" code uploaded to public code sharing sites like GitHub (https://github.com/). In many instances, the programmers inserted the login credentials into the programming code as a way to quickly test some feature or functionality without being slowed down by having to re-authenticate separately during multiple rounds of repeated testing, and then forgot to remove the "hard coded" credentials when the code was publicly available.

How Many Passwords Are Stolen?

All in all, these previous types of password theft lead to an enormous amount of password theft each day. How many passwords are stolen? It is estimated that millions of passwords are compromised each and every day.

One source who tracks password hacking criminal gangs reported (https://krebsonsecurity.com/2021/09/gift-card-gang-extracts-cash-from-100k-inboxes-daily/) that a single hacking group he was following "averages between five and ten million email authentication attempts daily, and comes away with anywhere from 50,000 to 100,000 of working inbox credentials." That is only a 1% success rate, but it is still 50,000 to 100,000 successfully gained passwords by a single hacking group each day. If they worked 365 days a year, it would equate to over 18 million compromised accounts each year, by a single group.

As the Brian Krebs article above also revealed, in 2020, Microsoft reported (https://www.microsoft.com/security/blog/2020/05/07/protect-accounts-smarter-ways-sign-in-world-passwordless-day/) that one in 250 of Microsoft's 240 million active customer accounts are compromised by password attacks every month. That equates to 960,000 password compromises each month. That is over 31,000 a day. Google revealed (https://security.googleblog.com/2019/05/new-research-how-effective-is-basic.html) that hundreds of thousands of password attacks are attempted against their customers every day. It is clear that password attacks, led by social engineering attempts, are the biggest portion of attacks and result in millions of stolen passwords every day. These stolen passwords often end up in password dump lists where anyone willing can reuse them.

Not all password attacks involve direct theft of passwords. The following sections remaining under "Password Attacks" cover attacks against passwords that do not include outright theft.

Password Guessing

Many password compromises involve attackers successfully guessing one or more passwords. It can be done to any online, active, accessible login portal that accepts the involved victim's login credentials. Common password guessing example portals are Microsoft Remote Desktop Protocol (RDP) login screens, email logins, File Transfer Protocol (FTP) portals, Secure Sockets Shell (SSH) login screens and application programming interfaces (APIs). The latter of which is often neglected by defenders, a fact that hackers love. There are many methods of password guessing, covered below.

User Preferences

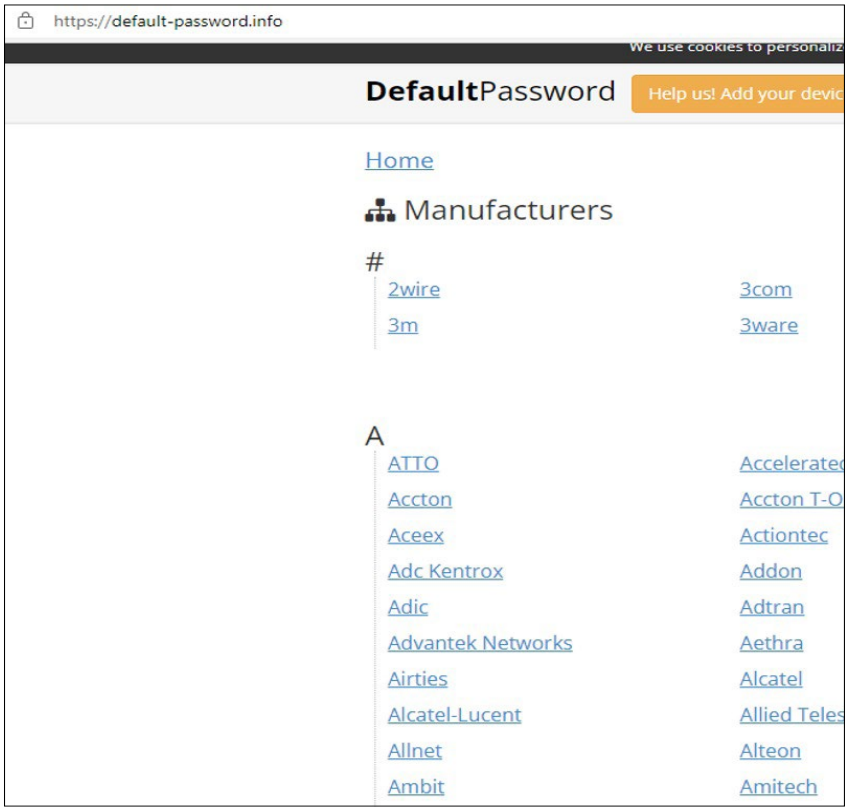
Users like to create passwords around their interests. People who love playing golf, often use passwords containing golf-related passwords (e.g., Titlist, tee, 3handicap, Tiger, Augusta, etc.).

People who have a favorite sports team often make passwords related to their sports team. Parents often use passwords containing their spouse's or children's names. Many malicious hackers, if they gain physical access to a person's work environment, will look around the potential victim's work area for the nearest pictures or hobbies. You would be amazed how many people's passwords contain something visible from a few feet of someone's desk chair. This is a big problem with human-created passwords.

Default Passwords Not Changed

Sometimes, the guesses cannot really be called guessing. Many devices are created with the same default passwords. Many times, there are passwords which are "hard-coded" by developers and cannot be changed. First-time passwords created in many corporate environments are very common or are created using the same simple pattern. In any of these cases, if the default password or pattern is known by attackers, they can use it to more easily guess passwords.

Anyone can type in "default password lists" into any browser search engine and find hundreds of known default passwords. There are hundreds of websites dedicated to default password lists. One example website listing default passwords is shown below.



Many times, the default passwords listed on these websites are old, but just as often, the passwords are still able to be used even though they were publicly known many years ago. Some of the biggest compromises in the world involved default passwords known by the public that were not (or could not easily be) changed by the involved vendor. Having these built-in, default passwords seems almost negligent to the average IT security practitioner, but it is not an uncommon issue. Many vendors simply do not know how to allow users to set new passwords without at least first starting with a commonly known password. After decades of abuse, the percentage of devices and systems with hard-coded and known default passwords is falling, but new instances pop up every single day.

Trying Common Passwords

There are also passwords that are so commonly used by multiple people that they are listed on "common password" lists. Surprising to some, despite the best efforts of the IT security team to educate people on how to pick non-common passwords for decades, the most commonly used passwords, such as Password and qwerty123, continue to be on the most commonly used password lists each year. Below is a figure showing the top 8 (out of 25) most commonly used passwords over time. You will see a lot of similarities over the years.

Common passwords often also include the user's name, birthdate or company name. For that reason, all password authentication systems should check for and deny any of the commonly used passwords or password patterns.

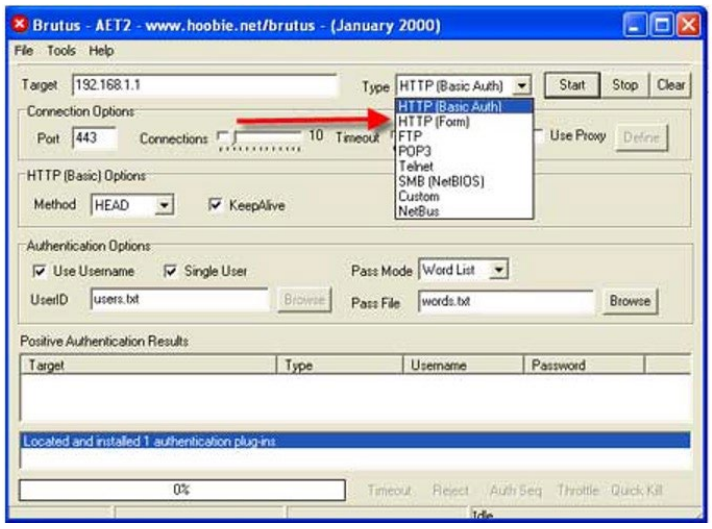
Top 25 most common passwords by year according to SplashData

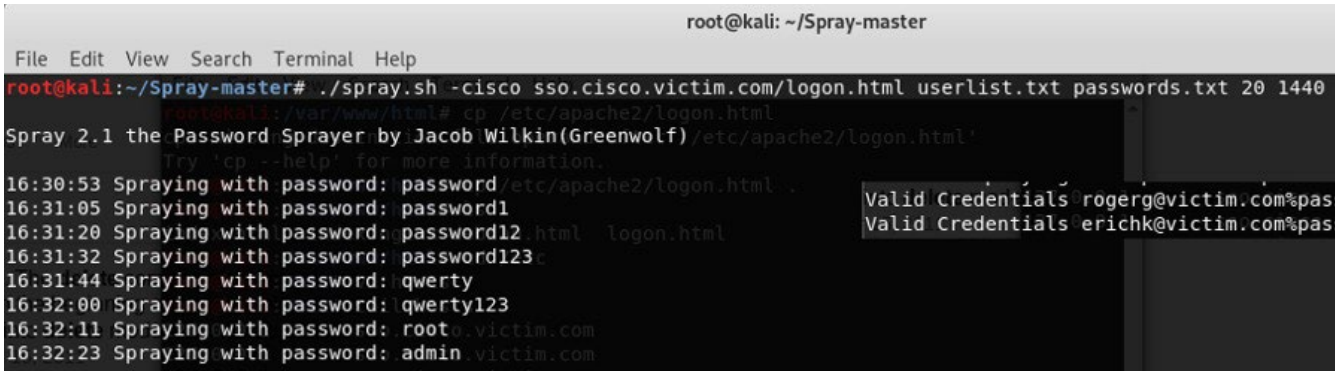
Rank	2011[4]	2012[5]	2013[6]	2014[7]	2015[8]	2016[3]	2017[9]	2018[10]	2019[11]
1	password	password	123456	123456	123456	123456	123456	123456	123456
2	123456	123456	password	password	password	password	password	password	123456789
3	12345678	12345678	12345678	12345	12345678	12345	12345678	123456789	qwerty
4	qwerty	abc123	qwerty	12345678	qwerty	12345678	qwerty	12345678	password
5	abc123	qwerty	abc123	qwerty	12345	football	12345	12345	1234567
6	monkey	monkey	123456789	123456789	123456789	qwerty	123456789	111111	12345678
7	1234567	letmein	111111	1234	football	1234567890	letmein	1234567	12345
8	letmein	dragon	1234567	baseball	1234	1234567	1234567	sunshine	iloveyou

Automated vs. Manual vs. Malware Guessing

Password guessing by attackers can be done one at a time or by using tools that automate the guessing. Manual password guessing by the hacker is most useful when the hacker knows something about the potential victim where they can use that personal knowledge to more quickly guess at the potential victim's password. Most password guessing is done by hackers using password guessing hacking tools. There are dozens to hundreds of these password guessing tools on the Internet.

Below are some screenshots from two password guessing tools.





With automated password guessing, the attacker must use a "password dictionary," which includes all the passwords to be guessed or is used as a starting point from which the password guessing tool adds different characters to guess at passwords containing more complexity. Password dictionaries are all over the Internet and contain from 100 to tens of thousands of included passwords.

The attacker configures their password guessing tool to use their selected password dictionary and then "points" the tool to the target's login portal. The tool then begins trying to log in as fast as it can or as fast as it is configured to guess. Password guessing can be done as fast as the login portal allows, which may limit guessing to just a few times a minute or could be as high as thousands of times a second. Attackers may be able to open up and use many simultaneous login sessions at once. How fast an attacker can guess depends on the involved login portal, the software involved, any deployed mitigations and how it is all configured. Many login portals only allow so many bad guesses in a certain time period before they temporarily lock out the account being used to do the login guessing (known as account lockout). So, an attacker needs to learn the account lockout policy of the involved target before guessing, if possible, and not guess at too fast a rate, which if exceeded, results in an account lockout (and prevents additional guessing).

Still, the longer and more complex a password is, the harder it is to guess or crack. In this author's current knowledge, the longest publicly confirmed successful password guessed in a real-world hacking attack is 10-characters long (https://www.world-today-news.com/municipality-of-hof-van-twente-hacked-by-simple-passwordwelkom2020-now/). This was done against a login portal without any account lockout controls enabled. The attacker was able to guess more than 100,000 times a day for over a year. So, it is generally believed that your password needs to be 12-characters or longer, with complexity, to withstand most known password guessing attacks, especially if account lockout controls are not enabled or cannot be confirmed.

Unfortunately, your password will have to be significantly longer to survive a password hash cracking attack (covered below), and for that reason alone, 12-character passwords are not considered strong if there is a chance the attacker can get your password hash. Read on.

Password Hash Theft and Cracking

In most operating systems and many applications, the "plaintext" (i.e., as typed) passwords that users type into login prompts are used and stored as their cryptographic hash representative forms. A cryptographic hash is an algorithm that takes any submitted plaintext content (in this case, a password) and runs it through an algorithm that creates a cryptographically unique result (called the hash result or hash). If the cryptographic algorithm is a good one, the hash result uniquely represents only the uniquely inputted password. Whenever the same password is inputted into the cryptographic hash, it turns out to have the same hash result. If any two different inputs (in our case, different passwords) are inputted, they create different hash results.

Password authentication systems have long converted typed in plaintext passwords into their cryptographically-derived hash results in order to make it harder for hackers to discover the plaintext passwords. A good hash algorithm creates a hash result which makes it difficult for an attacker to figure out the plaintext password it represents from the hash alone. When you type a login password into Microsoft Windows, it is converted into a hash and stored in the local SAM password database, or if it is an Active Directory network login credential, the password is stored as a hash in the NTDS. DIT file, which is stored on all involved domain controllers.

Many publicly known password hash cracking solutions can guess tens of trillions of guesses a second.

The idea is that if an attacker sniffs the Windows login authentication stream or gets a hold of the Windows password files (SAM or NTDS.DIT), then they do not immediately see nor are they able to use the plaintext passwords. The attackers would only get the hash of the plaintext password. The hash is often able to be used to authenticate to various sites and services, but frequently, especially in login portals, only the user's plaintext password can be typed in and accepted. Most attackers, even after learning a victim's password hash would rather have the plaintext version of the password because it is able to be used in more places.

Common Password Hash Algorithms

Hash algorithms usually result in fixed-length hash results regardless of the input (i.e., passwords in this case). Common hash lengths range from 128 to 512 bits. There have been many different generally accepted hash standards over the years, including Message Digest 5 (MD5), Windows LANManager (LM), Windows NT (NT) and Secure Hash Algorithm-1 (SHA-1). All of these previous standards, except for NT, which is the default password hash used by Microsoft Windows, are considered weak and broken. Today, the most popular hashing algorithm used by non-Windows operating systems is Secure Hash Algorithm-2 (SHA-2 or SHA2), although many of the most secure distributions use a hash, known as BCrypt, created by cryptographic luminary, Bruce Schneier. The table below shows some hash outputs for the word "frog" using common example hashes.

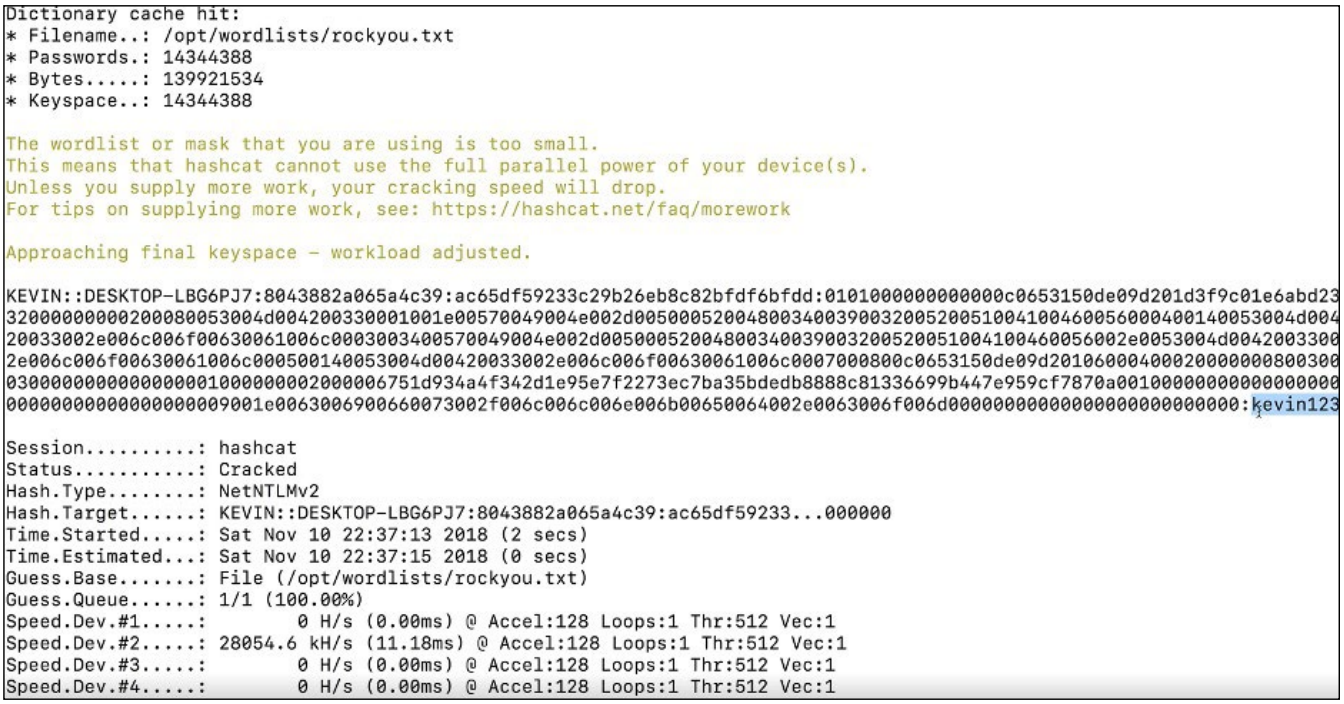
Hash Algorithm	Hash Result for frog
Message Digest5 (MD-5)	938c2cc0dcc05f2b68c4287040cfcf71
LANManager (LM)	71CF7241255BBEB4AAD3B435B51404EE
Windows NT (NT)	E3EBB26FE8A631171D218D084C76C982
SHA1	b3e0f62fa1046ac6a8559c68d231b6bd11345f36
SHA2-256	74fa5327cc0f4e947789dd5e989a61a8242986a596f170640ac90337b1da1ee4
BCrypt	$2y$10$5lSoGVbVHgmVVvV2J5Cxt.RFJyjVA38InpRbIP/GZo5vQAetjnv9S

As you can see, the hash result is quite different for each hash algorithm and theoretically very hard to impossible for an adversary to simply guess at its plaintext equivalent. But there is a way to convert hashes to plaintext passwords with a password hash cracking tool and pre-computed password hash dictionary. Attackers obtaining a potential victim's hash (covered more below), can compare the stolen hash to a database holding pre-hashed potential passwords. Using the

example hash above, a password hash "cracking" database might already hold the word frog and its pre-computed LM hash equivalent of E3EBB26FE8A631171D218D084C76C982. Thus, a password hash cracking tool could take the stolen hash of E3EBB26FE8A631171D218D084C76C982 and then look for it in the pre-computed password hash database and upon finding a match, look up that the attached plaintext word is frog and report that to the user.

Password Hash Cracking Tools

There are dozens of free and commercial password hash cracking tools. You may hear them called "rainbow tables," although rainbow tables are a commonly used type of password hash cracking tool/database, and it is not equivalent to all types of password hashing tools/databases. For decades, one of the most common password hash cracking tools was John the Ripper (https://www.openwall. com/john/). Its popularity has been replaced in the last few years by a password hash cracking tool known as hashcat (https://hashcat.net/hashcat/). A screenshot example of hashcat being used in a demonstration attack is shown below.



Real-World Password Hash Cracking

Like password guessing, password hash cracking can be mitigated by using longer and more complex passwords. Unlike password guessing, password hash cracking can occur at a far faster speed, only limited by hardware and software. The average password guessing attack can only guess a few hundred to maybe a hundred thousand passwords a day (at best). Password guessing speed is limited by what is allowed by the active, online, login portal. Password hash cracking, however, is done in an offline manner. The hacker has the password hash and can guess at it as fast as possible. Password hash cracking comparisons are often done at speeds exceeding trillions of guesses per second (TH/s). Many publicly known password hash cracking solutions can guess tens of trillions of guesses a second. It is known/believed that many organizations have the capability to do more than 100 trillion guesses per second, although it has not been publicly confirmed. It is estimated by password hash cracking experts that anyone can buy 100 TH/s password hash cracking capability with cloud computers within the budget of many, many adversaries. Since there is no account lockout threat or rate limiting to bother the attacker, they

can guess at password hashes as fast as the hardware and software combination they are using allows.

Not all password hashes can be guessed at with the same quickness. Some hashes are easier to guess against than others. The Windows NT hash is considered to take about medium effort to guess against, SHA-2 is harder, and BCrypt hashes are considered among the hardest to crack. Thus, even if a password is identical, it would take far longer to crack a password hash protected by SHA-2 and BCrypt than for Windows NT hashes.

How Fast Can Password Hashes Be Cracked?

Password hash cracking speed on 45TH/s rig on perfect random passwords

		Password Length						
Hash	Eff. Speed	6	7	8	9	10	11	12
LM	15.81 TH/s	instant	instant	instant	instant	instant	instant	instant
NT	31.82 TH/s	instant	instant	3.5 min	5.5 hrs	3 wks	5.6 yrs	538 yrs
MD5	17.77 TH/s	instant	instant	6 min	10 hrs	9 wks	10.1 yrs	963 yrs
SHA1	5.89 TH/s	instant	instant	19 min	29 hrs	15 wks	30.6 yrs	2.9M yrs
SHA2-256	2.42 TH/s	instant	instant	45.5 min	3 days	37 wks	74.25 yrs	7.1M yrs
SHA2-512	801.9 GH/s	instant	instant	2.25 hrs	9 days	28 mon	225 yrs	21.4M yrs
BCRYPT	11.37 MH/s	18 hrs	9 wks	59 yrs	5.6M yrs	534M yrs	50744M	4820655M

Data from: https://t.co/NKYIrKwUDb

Note: Microsoft Windows also uses PBKDF2 for protecting locally cached login credentials. PBKDF2 is considered only "beat" by BCrypt up to 54-character passwords. But 55-character or longer passwords are better protected by PBKDF2.

It is VERY important to note that the speeds listed above are from brute force attacks against perfectly random passwords. Human-created passwords are almost never perfectly random and can be cracked far, far, more quickly. How many real-world human-created passwords would stand if the adversary guessing against them could guess tens of trillions of times a second?

Not many. It is commonly believed that 60% to 70% of people's passwords would be cracked in a day or two (https://krebsonsecurity.com/2021/07/the-life-cycle-of-a-breached-database/). It is estimated that any eight-character or shorter Windows password will be cracked in under two hours using a normal password hash cracking rig or using $25 worth of cloud CPU time.

This author is aware of penetration testing teams routinely cracking human-created Windows password hashes for passwords up to 18-characters long for both regular user and service accounts. This has been a particular high-risk problem with a Windows attack type known as Kerberos roasting. An attacker, which has gained regular domain user access with normal privileges, can retrieve the NT hash of any account with a Kerberos Service Principal Name (SPN). Windows service accounts, which are often among the most privileged accounts in a Windows network, usually have SPNs. Thus, an attacker can compromise a regular user's account, use it to pull the NT hashes of the service accounts running on the Windows domain, and usually end up with very elevated access, if not complete compromise of the Windows network, usually within minutes to days. If you're interested in learning more about Kerberos roasting attacks, see: https://www.qomplx.com/qomplx-knowledge-kerberoasting-attacks-explained/.

Password hash cracking can guess far more quickly than password guessing and so it takes longer passwords to protect against password cracking attempts. A truly perfectly random password using all possible 94-characters on the standard US keyboard starts tobecome uncrackable at around 10-characters long, although most experts say defenders should use 12-character perfectly random passwords just to be sure.

Truly Random Password Examples

R#Yv&ZCAojrX

ELv!2MibAb>RC?ru

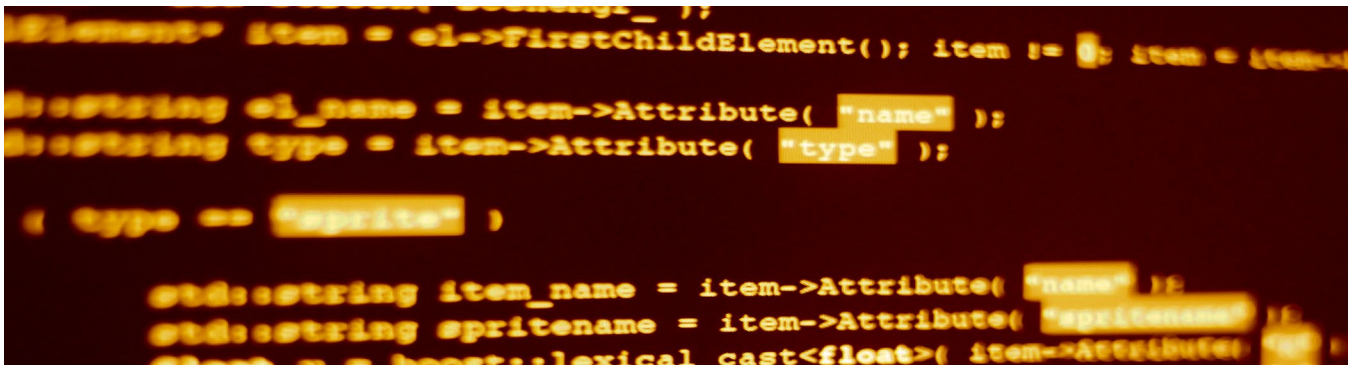
a!#=dH)vvLykiJhu

But human-created passwords with the "normal complexity" that a person usually creates and uses are known to be crackable up to at least 18-characters. And new records are being set every year. There are likely to be cloud-based and supercomputing password hash cracking scenarios, like a nation-state adversary would have access to, that would be able to crack passwords up into at least the 20- to 30- character ranges. Nevertheless, unless you are worried about a nation-state level attacker, it is generally believed that passwords with normal complexity need to be at least 20-characters long to withstand the attacks from more common adversaries. And if you're worried about higher resourced adversaries, use 30-character passwords.

Note: SHA-2- and BCrypt-protected password hashes would not be nearly as easy to crack as NT password hashes, and thus, passwords protected by those forms of hashes can be shorter as compared to Windows passwords and maintain the same level or greater protection.

If worried about most password hash cracking attempts, a user should use at least a truly, perfectly random 12-character password or a 20-character password with normal complexity. If worried about nation-state level attacks, you should go longer. Many password experts recommend easier to create and use passphrases. A 27-character passphrase like rogerjumpedoverthedogandcat or 34-character "I went to Cowboy Chicken for lunch" are easier to create and use than a complex password, and substantially resistant to password hash cracking. Unfortunately, most password requirements are going to require some normal complexity in any password/passphrase you create, so your passphrase will likely have to contain a number and/or character as well. For example, r0gerjumped0verthedog&cat or "I went to Cowboy$ Chicken 4 lunch." It's not ideal and is one of the reasons why using MFA, when possible, and/or a password manager to create and use truly random passwords is a better idea.

All enterprise defenders should routinely take their user's password hashes and test them against a password hashing rig to see how well they fare. Readers interested in advanced password hash cracking should read Hash Crack: Password Cracking Manual (https://www.amazon.com/Hash-CrackPassword-Cracking-Manual/dp/1793458618).



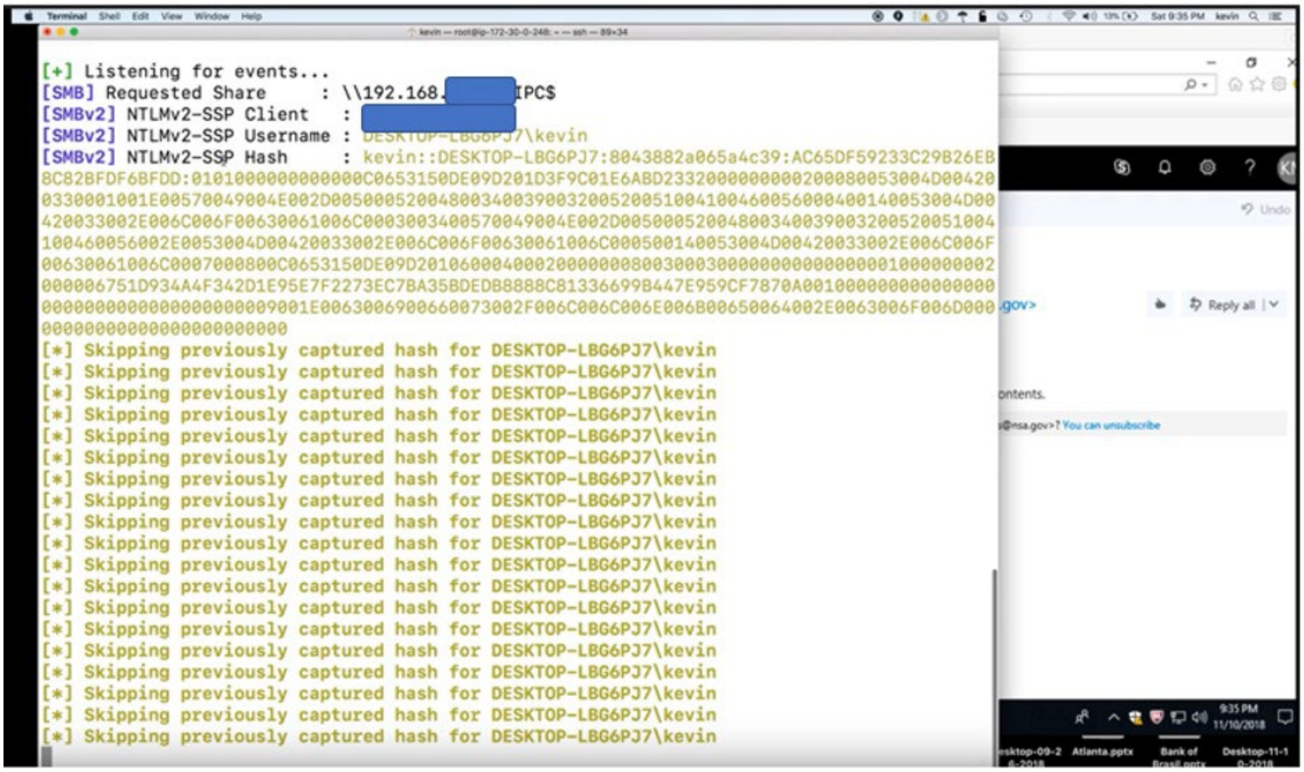
Even more concerning are scenarios where password hashes can be stolen by an attacker, remotely, and without any previous user account compromises on the targeted system or network.

One of the biggest challenges for a password hash cracker is capturing the password hash. This is not trivial in most scenarios, today. Most operating systems and devices understand the importance of protecting password hash databases and the hashes they contain. Hence, it usually takes a top, elevated login credential (e.g., Administrator, root, LocalSystem, DomainAdmin, EnterpriseAdmin, etc.) to be able to see and extract the hashes. An intruder with that level of security context can do anything. It is already game over. They could place key logging trojans and capture the password, no matter how long and complex, as the user types it in. They could alter the operating system not to even care about passwords or simply place themselves in an elevated group where they could come in and out of the system and do anything the hardware and software is capable of. Although attackers routinely have this level of access on most systems they compromise, it is a minimum technological hurdle that must be obtained by the attacker to extract the hashes and begin the password hash cracking.

However, there are scenarios where password hashes can be learned or leaked to an attacker without the attacker first having to obtain super elevated access to the system where the password hashes are stored, such as Kerberos roasting (as covered above). Even more concerning are scenarios where password hashes can be stolen by an attacker, remotely, and without any previous user account compromises on the targeted system or network. These scenarios are not commonly used by attackers, because they can be unreliable and involve particular requirements, but when those types of attacks are possible, a knowledgeable hacker can exploit them.

Stealing a Password Hash Through Email

A good example of this is a Kevin Mitnick demo (https://blog.knowbe4.com/kevin-mitnick-demos-password-hack-no-link-click-or-attachments-necessary). In this demo, Kevin (who is both the victim and attacker again) sends a phishing email to a user. The user opens the email. The email contains some hidden URL links that attempt to download an object from the attacker's web server (running the open-source software called Responder). This causes the victim's browser to begin to try to log in to the attacker's web server using stored network credentials. The attacker captures the login negotiations (shown in the figure below), using Responder, from which he or she then derives the victim's NT password hash. The attacker would then be able to reuse the password hash directly, where allowed, or start to crack the victim's password. For more details on this attack, see: https://www.csoonline.com/article/3333916/i-can-get-and-crack-your-password-hashes-from-email.html. For years, Kevin Mitnick and the author of this whitepaper talked about this type of non-elevated password hash attack as more theoretical than something most readers needed to be worried about. However, in the years since Kevin did his first demo of this attack (which was already "in the wild"), it is starting to show up more often, although still very infrequently. Here are some example real-world attacks using similar methods: https://sensorstechforum.com/adobe-cve-2019-7089-second-patch/ and https://blog.lumen.com/newly-discovered-watering-hole-attack-targets-ukrainian-canadian-organizations/.



In summary, password hash cracking attacks are not as common as password theft attacks, but still regularly occur in the real world. How much risk you assign to password hash cracking attacks will determine how long your passwords should be. If you are worried about them, your passwords need to be 12-character perfectly random passwords or 20-character or longer passphrases. This paper considers password hash cracking a worrisome risk and consequently the recommended password policies will recommend longer and stronger passwords.

Unauthorized Password Reset or Bypass

The last way a password can be hacked (covered in this paper) is by resetting or bypassing a password without the legal authority to do so. Many password authentication systems allow someone to reset their password or to bypass it by going through a series of alternate "recovery" steps. Password recovery portals and procedures are frequent causes for bogus password resets and bypasses.

The most common password reset method is for the user to be sent a link or code to an alternative email account provided by the user when they first set up the login account. The figure below is an example recovery email.

Attackers routinely compromise a user's alternative email address (usually through phishing and social engineering) and then attempt to "recover" the user's primary login account. The recovery message is sent to the victim's alternate email account, which is now in the possession of the hacker. The hacker gets the code, resets the victim's account password and takes over control.

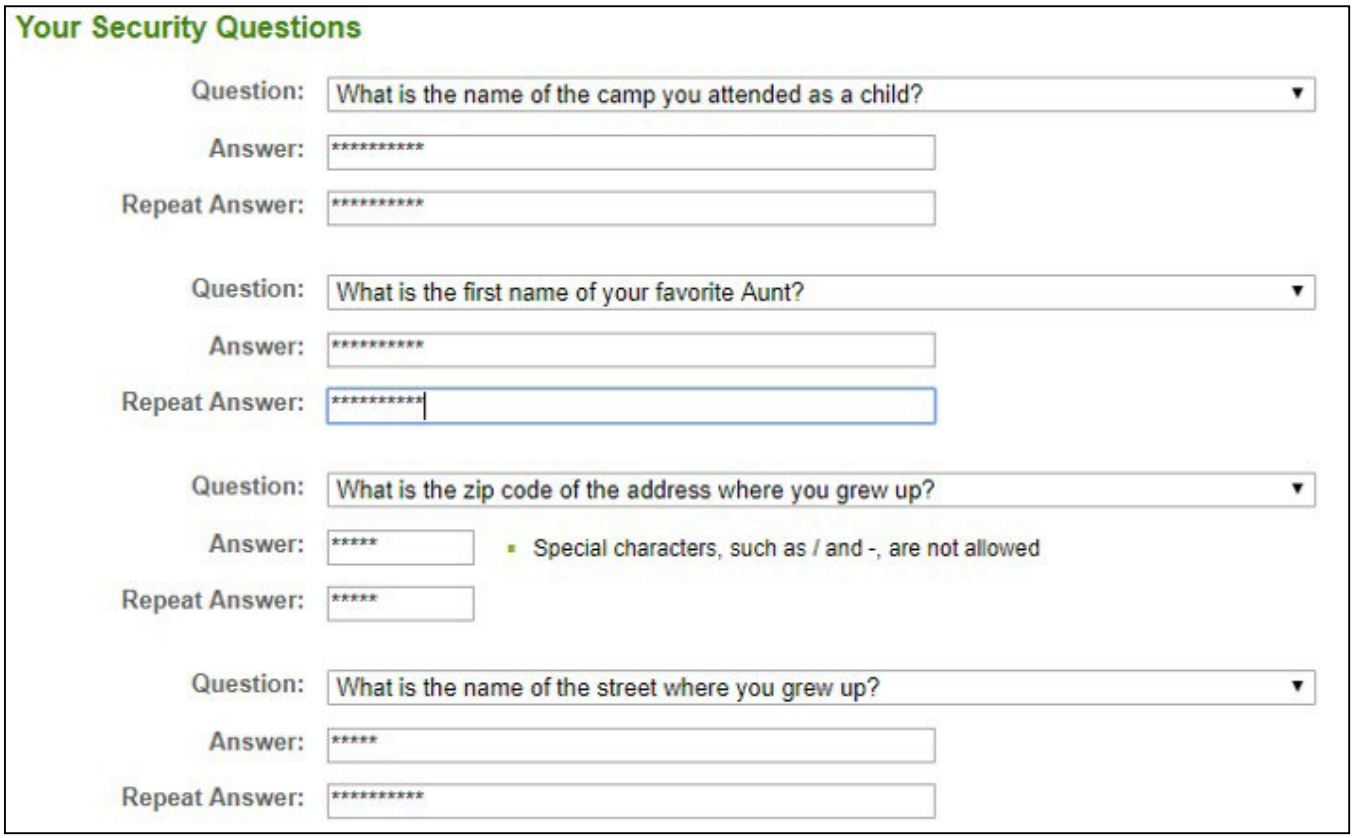
These days, many times, the recovery code is sent to the user's phone number via SMS. In these cases, the hacker is often able to move the victim's phone number to another phone under the hacker's control (known as SIM Swap Attack or Port Out Fraud). Either way, the SMS-based reset code gets sent to the phone number now under the hacker's control and they use that code to take over the victim's account. This type of login account recovery abuse is so rampant that the

U.S. government said SMS-based login protection should not be used to protect valuable data and information back in NIST Digital Identity Guidelines (NIST Special Publication 800-63), released in 2017. These sorts of attacks, using passwords and MFA, still regularly occur, including this 2021 attack example: https:// blog.knowbe4.com/hackers-rob-thousands-of-coinbase-customers-using-phishing-attacks-and-anmfa-flaw.

Attackers also routinely send SMS messages or voice calls to a potential victim pretending to be Microsoft, Google, their bank, etc., and tell the user that their account is in trouble (usually being used fraudulently). They will then tell the user that to verify and help them, they need to send the user a code, which the user needs to repeat back to them. The attackers then start to recover the victim's account on the real login portal, which sends a code to the victim, who thinks it is a verification code from the SMS message or caller, then repeats it back to the attacker (see simulated example to the right).

Once the victim has repeated the sent recovery code to the attacker, the attacker uses it to complete the account recovery and takeover. You need to always verify the identity of anyone calling or texting you and do not take their word as to who they are. It is very easy to fake SMS messages and phone calls.

The worst type of password reset method is known officially as personal knowledge-based questions, but most people know it as password reset questions. In these scenarios, people who have login accounts are asked to answer a pre-set series of questions (such as mother's maiden name, their favorite car or favorite elementary school teacher, etc.) or they can make up their own questions. See the example below.



Anyone wishing to reset their password has to successfully answer one or more password reset questions in order to reset their password. The idea is that it is supposedly harder for an attacker trying to fraudulently reset your account to know the answer to these questions. In reality, most of the answers are far easier for an attacker to guess or find than any possible reasonable

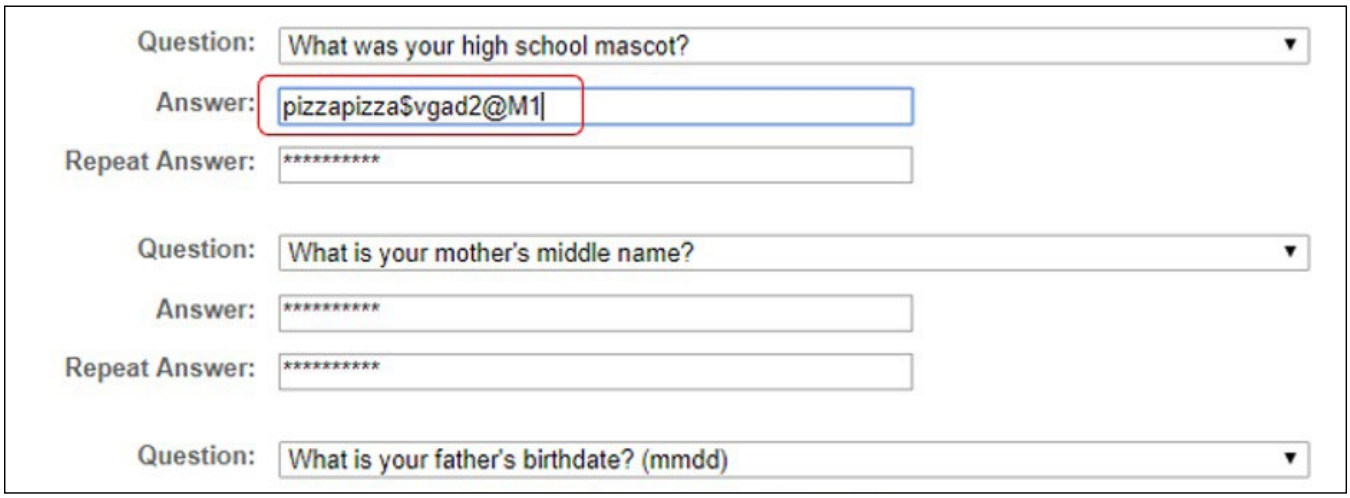
password you could have used. For example, your favorite car is likely going to be a model out of the top twenty car models selected by everyone. There are only about 100 different car models in a given year, and they do not change that much year to year. So, your password may have millions or billions of possible guesses, but your favorite car model might be guessed in a few guesses by an attacker.

In fact, a study done by Google called Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google (http://www.a51.nl/sites/default/files/pdf/43783.pdf) determined personal knowledge questions were terrible at protecting people's security. For example, some recovery questions could be guessed on the first try 20% of the time. Forty percent (40%) of people were unable to successfully recall their own recovery answers and sixteen percent (16%) of answers could be found in a person's social media profile.

Hackers have been successful at using people's password reset questions to take over accounts for decades. These types of attacks have been involved in many well-known attacks against celebrities, including Sarah Palin, when she was running for Vice President: https://en.wikipedia.org/wiki/ Sarah_Palin_email_hack.

Do not put your faith into password reset answers, at least the simple type, any time you can avoid it. Unfortunately, you will not be able to avoid using them, as many major vendors still require that you use them to register an account on their site or service. And even though you may not ever use them, an attacker could fraudulently pose as you on their recovery portal and begin the process.

If you are ever asked to fill out answers to questions, consider answering them falsely (see example below).



There is no law that says your answer has to be the real answer. This is not Jeopardy™. Unfortunately, if you use this method, it means you will likely have to write down both the questions and the nonsense answers in a secure place (e.g., a password manager) in case you need them.

Quantum Attacks

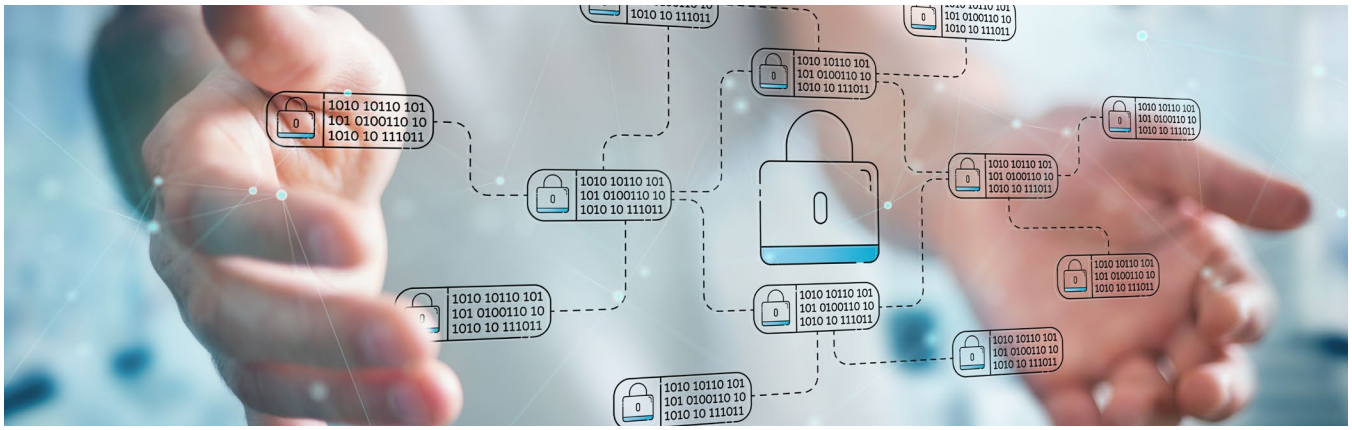
Some readers may wonder if forthcoming, sufficiently capable quantum computers could make password attacks worse. The short answer is yes, some types of attacks, but not the most popular kinds of password attacks. Sufficiently capable quantum computers will be able to crack traditional asymmetric encryption and halve the protective power of symmetric encryption and

hashes. The hash issue is the more important issue. There is a chance that quantum computers could speed up hash cracking which would require password hash lengths to double for the same level of protection, although hashes of 256-bits or longer are considered protective enough even for coming quantum attacks. There is some study into whether traditional password guessing would speed up as well, found in this research paper: https://eprint.iacr.org/2021/1299.pdf.

It is very important to realize that it does not take the capabilities of quantum computers to do most password attacks, and ALL defenders should be far more worried about traditional password attacks than how quantum computers will impact them. You are very, very, very unlikely to be compromised because of a quantum computer any time soon.

Password Attacks Conclusion

There are many ways in which your password can be attacked. The vast majority of password attacks are thefts using social engineering and phishing. Password-stealing malware is also quite common, as is password guessing attacks. Compromising passwords using recovery or reset methods is less common, as is password hash cracking, but they are still real-world threats which occur with enough regularity that they need to be strongly considered.



PASSWORD POLICY ATTACK DEFENSES

This section of the paper will discuss the defenses needed to prevent known password attacks, both on individual passwords and enforced password policy. Your password policy needs to prevent password theft, password guessing, password hash cracking and unauthorized password recovery or bypasses.

What Password Attacks Care About Password Complexity

It is very important to realize that most password attacks do not care what your password is or what characters make up your password. They simply steal it, no matter what your password is. For example, if you use a very long and complex password, say 30-characters long, but get socially engineered into revealing it to a fake website, they will get and reuse it just as easily as if it were a six-character password. Let me say that again. The vast majority of password attacks do not care what your password or password policy is. Only password guessing and password hash cracking attacks care how long or complex your password is. The following attacks do and do not care about your password composition:

So, while you need to create passwords and password policies that prevent password guessing and cracking, the vast majority of the risks are the other types of password attacks, which don't care what your password policy is. And many other attacks, such as against unpatched software (which is the number two most popular hacker attack method overall) and socially-engineered trojan horse programs do not care what your passwords are. Thus, protecting against most password attacks are prevented by making sure to mitigate social engineering and other general types of defenses (e.g., antivirus software, secure configurations, etc.).

Password Policy Components

At the enterprise level, the enforcement of the necessary defenses (if they can be enforced) is called password policy. On some networks, all users and groups are controlled by the same password policy. On other networks and systems, different types of users and groups can have different password policies configured. If this is the case, you typically want strong password policies for elevated users and groups. In systems where the compromise of even "regular users" could be very harmful to the involved organization, the password policy should be sufficiently strong for everyone.

A typical password policy has the following components:

- Minimum password length
- Maximum password length
- Make up and required complexity
- Maximum password age (and sometimes minimum password age)
- History (how many different passwords a user must use before they can reuse a previously used password)

We will discuss each in more detail below, along with some other necessary password policy characteristics.

Minimum Password Length

You want enough required characters in a password to mitigate password guessing and password hash cracking. If a login portal (or API) has typical account lockout enabled, (say six bad logins triggers a lockout), then a password with eight characters and some complexity is usually long and complex enough to stop password guessing attacks. If every login portal (and API) CANNOT be confirmed as having account lockout enabled (it is often beyond your control or unknown), then you need at least a 12-character password with complexity. If you are worried about password hash cracking, as you should be, then your password/passphrase should be 20 characters or longer (especially in Microsoft Windows environments).

Complexity

Complexity is the usage of additional characters to add to the number of possible characters that a password guesser or cracker would need to use to compromise a password. The more possible characters and character types (e.g., letters, uppercase, lowercase, numbers, symbols, etc.) allowed or required, the harder it might be for a password attacker to guess or crack a password. The more random a password appears to be (as compared to regularly constructed words or names), the harder it is for an attacker to guess or crack.

If users were never forced to use complexity, most would use lowercase alphabetic characters only (26 in the English language). When forced to use both uppercase and lowercase letters and numbers, the number of characters becomes 62 (for English users). Most password policies with complexity required entail a combination of uppercase and lowercase letters, a number and a symbol from the keyboard. On a standard U.S.-style keyboard, this works out to 93 or 94 different symbols. Any attacker knows that most user-created passwords are still likely to be composed mostly of lowercase letters, but any increase of allowed or required characters beyond just 26 lowercase characters makes the password guesser's job harder. To prevent easy password guessing and cracking, some type of password complexity should be required.

Alt-Character Passwords

Many people do not know that you can often use dozens to tens of thousands of other characters that are not directly listed on your keyboard keys. How you access them...whether you can access them...depends on your operating system and/or application you are using. In many cases, such as with Microsoft Windows, you can access the additional characters by hitting or holding down your Num Lock, then holding down your Alt key, followed by a three-to-four numeric digit representing the ASCII or Unicode character you want. For example, Alt+0134, and it displays the ĵ symbol. Here is a list of "alt codes": https://www.alt-codes.net/.

The password defender's thinking is that by using an alt code versus a regular letter, number or symbol, that is readily accessible on a keyboard that a password guesser or cracker will not even think to try the alt code. Although possibly good for defending against password attacks, alt characters are a big pain to use, often not supported on the computer, operating system or application, and in general, cause all sorts of problems. For example, many websites and applications appear to take passwords with these special characters involved, but when you go to use them again, they do not work. Either they do not work at all or you have to use a space instead. In general, passwords with alt characters should be avoided.

Entropy

In the world of password creation and defense, the "randomness" of a password is known as

entropy. An increase in the number of used types of characters (called character sets) increases entropy. An increase in the length of a password increases entropy. The more random-looking a password appears, avoiding looking like predictable words and patterns, increases entropy.

The more randomness or entropy a password has, the harder it is for a password guesser or cracker to break. Unfortunately, it is usually harder for a human to create and use high entropy passwords.

Many people think of a complex password looking like this:

- RogerisaG0on
- RogerGrimes3
- R0ger12@G!

These may have more entropy than most user-created passwords, but are not truly super random with high entropy – the types that password guessers and crackers hate. As previously covered above, here is what truly random passwords with high entropy look like:

-]}7Y?@w@?)Nmt4h7
- J.MF.F)RGzHk4y}x
- CYADB_d},R->Z>C2

Users do not want to create or use these types of passwords, but when they are used, it makes it far harder on the password guesser and cracker.

Password Testing Websites

You can "test" if your current or desired password has enough entropy to stop different types of attacks. Here are some websites, known as password checkers, which will tell you how long it would take the average password hacker to guess or crack your password:

- https://howsecureismypassword.net/
- https://password.kaspersky.com/
- https://thycotic.com/resources/password-strength-checker
- http://www.passwordmeter.com
- https://www.howsecureismypassword.io/

Caution: Any website asking you to submit your real password to determine strength could be using your submission against your own interests. Instead, use another similar password with the same complexity characteristics, but not your real password. For example, if your real password is Dogdog32 use CatCat23 instead.

Here are some examples from: https://howsecureismypassword.net

What is important to note is that the password most people would consider complex, Rogergri2, would not take long to crack. The middle example has high entropy, but would be difficult for a human to create and use, especially many of them for different websites and services. The last one, a passphrase, rogerjumpedoverthedogandcat, has neither really high complexity or good entropy, but if the attacker did not know what it was composed of (i.e., English whole words only), they would still have to use the normal character sets in their cracking, and so it is well protected. However, if the attacker learned that it was a passphrase and likely contained only different English words, its "protection" would diminish, as the password hacker would be likely to start guessing or cracking using whole English words instead of individual characters.

Higher entropy is better to prevent password guessing and cracking attacks, but increasing entropy increases the likelihood that human-created and used passwords get reused across multiple websites and services. Instead, use a password manager to create high entropy passwords for each and every website. If a password manager cannot be used, think about using normal password complexity passwords or passphrases instead.

Forced Password Changes

Many password authentication systems allow a maximum password age (i.e., expiration period) to be set. For the best password security, passwords should be forced to be reset at least once a year, if not more often (say every 180 days) for some high security environments. Forced password resets prevent previously stolen passwords which have not been used yet from being used after the expiration date. It also limits how long a password guesser or cracker has to guess at a particular password before the targeted password changes.

There are many authoritative resources, most notably NIST Special Publication 800-63 (https://www.nist.gov/special-publication-800-63), which argue against forced password resets unless the involved passwords are known to be involved in a compromise. The lead author of this whitepaper, and many other password experts disagree with this advice. You can read the lead author's reasoning here: https://www.linkedin.com/pulse/why-passwords-must-periodically-changed-roger-grimes/.

Password History

Some password authentication systems, notably Microsoft Windows and Active Directory, keep a list of a user's past passwords so that they can be prevented from being used again by the user. The thinking is that if a user switches passwords, but then relatively quickly switches back to an older password they used, it increases the chance that a stolen, guessedor cracked password attack could be successful because the user went back to an old password. When password history enforcement is enabled, the user is prevented from using older, previous passwords for a certain period of time or must use a certain number of new passwords (oftentimes for a certain minimum period of time) before they can reuse a previously used password. The idea is that forcing a user to first use a bunch of new passwords for a certain length of time increases the odds that the user will simply use a new password and stick with that instead of going through all the effort needed to reuse an old password.

Password history and preventing users from reusing old passwords decreases password guessing and cracking risk.

Not Easily Guessed

No matter what the other password policies are, passwords should not be easily guessable. This means that they should contain:

- Minimum password length (say eight characters)
- Moderate amount of required complexity
- Not be an old password that the user has previously used
- Not be composed of their name or login name
- Not use any passwords previously identified as “common passwords”

Use a Strong Hash

Passwords should be hashed using strong hashes possible on a system, such as NT, SHA2, SHA3, BCrypt, PBKDF2 or other similar strong hashes. The hashes should contain a salt component, which should be randomly generated and not easy to associate with the login account which contains it. Salting does not contain as much protective power as many people think, but it does prevent easy hash cracking, especially when two identical passwords are contained within the same stolen password hash group. Without a salt, successfully cracking one password hash would immediately lead to an immediate cracking of the other identical password. With a salt involved, it makes cracking the second identical password less immediate.

No Sharing

It is very difficult to enforce this requirement using a technical defense. Usually, education is your best tool. Make sure all users (and yourself) understand how important it is to not reuse identical passwords or even similar password patterns between unrelated sites and services. Users who reuse the same passwords or patterns across multiple sites and services (which most users do), increases the chance that one password compromise will more easily lead to additional compromises of other sites and services where the user is registered.

Emerging Technology – Keystroke Dynamics

Keystroke dynamics is the biometric identification of user passwords based on the unique ways users type in their passwords. Keystroke dynamics measures every keystroke, action and timing, to identify if a password typed in by a user is likely to be the legitimate user who normally uses that password. Keystroke dynamics is a way to prevent stolen passwords from being used as easily by attackers. The attacker or their automated program is unlikely to type in or use the password in the same way, using the same keystroke timing, as the legitimate user. Passwords typed in or used in ways not similar to the legitimate user can be used to reject logins or to place a higher risk rating on the identity logging in so that it can be further followed and reviewed. Keystroke dynamics is not commonly used, but there are some companies offering the technology.

Optimal Password Policy

An optimal password policy would be to use phishing-resistant MFA whenever possible, instead of using passwords. Phishing-resistant MFA is discussed here: https://blog.knowbe4.com/u.s.-government-says-to-use-phishing-resistant-mfa. Unfortunately, most of today’s MFA is susceptible to common forms of phishing. A list of phishing-resistant MFA can be found here: https://www.linkedin.com/pulse/my-list-good-strong-mfa-roger-grimes.

When MFA cannot be used, a password manager should be used to create and use perfectly random 12-character or longer passwords wherever a password manager can be used. This is

because perfectly random passwords become unguessable and uncrackable at 12-characters long (as far as we know). And it is very easy to create and use even longer perfectly random passwords with a password manager. Your password manager should be protected by phishing-resistant MFA and/or a 20-character or longer human-created password. Any device which holds your password manager should be protected by phishing-resistant MFA and/or a 20-character or longer human-created password. Thus, if possible, only two human-created, 20-character or longer passwords would be needed and everything else would be protected by a phishing-resistant MFA or a perfectly random password implemented within your password manager.

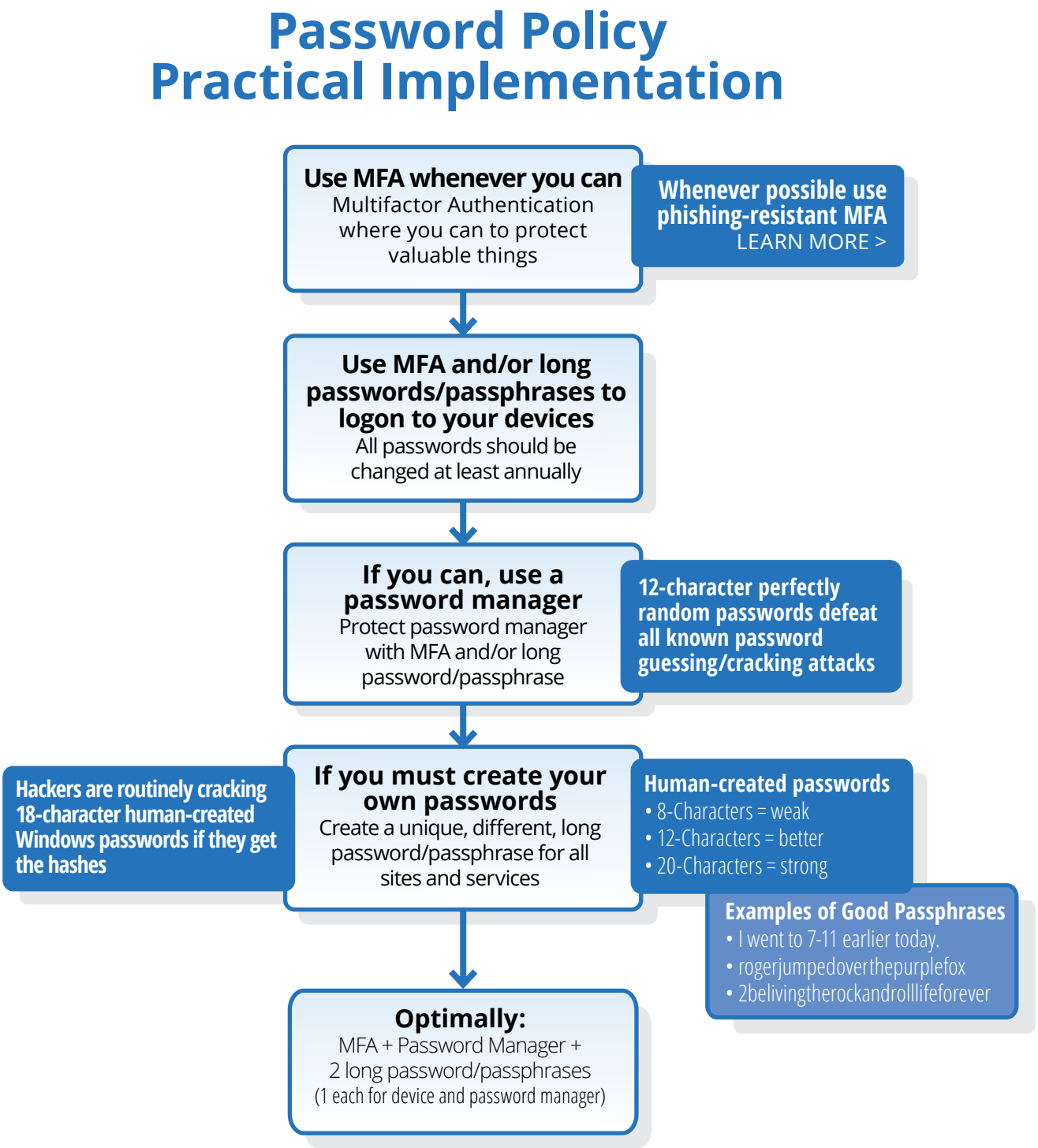
It is noteworthy that password managers can become a single point of failure, which if successfully exploited by an attacker can lead to the quick theft of all passwords and credentials contained within fairly rapidly. This risk is real and must be considered before using or recommending that a password manager be used. However, most attacks that compromise a password manager, could otherwise compromise any device or software storing or using passwords. An attacker could simply install a password-stealing or keylogging trojan and get passwords even without a password manager involved. Today, the vast majority of password risk (after social engineering attacks) comes from weak passwords or users re-using the same passwords across unrelated sites and services, a risk which password managers mitigate. Thus, the risk of using a password manager is seen by the author of this paper to be an acceptable risk as compared to the risk it mitigates. Further discussion can be read here: https://www.linkedin.com/pulse/can-your-password-manager-hacked-roger-grimes.

Each user and administrator must assess the risk of using a password manager, and if the risk is too high, then use other alternatives, whatever they may be. It is also possible that future attacks may become far more commonly directed toward password managers, and start to make the risk of using a password manager not worth it. Accordingly, as with any defense, it may be necessary to re-evaluate the overall risk and threats again at a later date. For now, this paper considers the use of password managers worth the risk mitigation they provide and recommends their use.

While KnowBe4 cannot recommend which password manager to use, there are several good password manager reviews on the Internet, including: https://www.wired.com/story/best-password-managers/.

WHAT YOUR PASSWORD POLICY SHOULD BE

With everything discussed above, password attacks and defenses against them, this is what your password policy should be as summarized into a single graphic:



No matter what passwords are involved or how they were created, they should be changed at a maximum of one year.

OTHER SUPPORTING POLICIES

Passwords cannot be protected by using password policy alone. It takes a set of overlapping, defense-in-depth, best practice policies to support your password defense. These best practices include:

- Educate users about password attacks and defenses, and why following your recommended password policy is important to protect your organization
- Make sure users understand why they need to use a unique password on every site and service
- Mitigate phishing and social engineering (see https://info.knowbe4.com/comprehensive-anti-phishing-guide)
- Do perfect and timely patching of Internet-accessible software frequently targeted by hackers
- Do appropriate security event monitoring, especially for bad or failed logins
    - Alert for an abnormal number of failed logins in a given time period
        - For a single account
        - In aggregate
        - For an unusual number of accounts (stops credential stuffing attacks)
        - Alert for an abnormal number of account lockout warnings
        - Alert on strange network login pathway flows
        - Logins for devices that do not normally log in to other devices
- Use up-to-date antivirus/endpoint detection & response software to detect and prevent password stealing malware
- Do account credential hygiene
    - Remove the accounts you do not need
    - Put MFA and/or strong passwords on the ones you do need
    - Reduce permanent memberships of privileged groups to as near zero as you can
    - Enable “check-out” methods and monitoring of elevated accounts
    - Do an account audit to ensure that all existing (active) accounts have strong passwords
- Appropriately protect the password authentication system, including storage, use and transmission
- Change all default passwords immediately
- Search for and eradicate any login credentials accidentally stored in programming code
- Disable weak hash algorithms, like MD5, SHA1, LM, if used, if possible
- Disable weak cryptography, like DES, etc., if used, if possible
- Secure APIs, including with account lockout and monitoring

- Make sure single-sign-on (SSO) portals use different passwords to log into all unrelated sites and services

KNOWBE4 TOOLS

KnowBe4 has many free tools and services to help anyone better evaluate their organization's password situation, including:

- Breached Password Test (https://www.knowbe4.com/breached-password-test)
- Weak Password Test (https://www.knowbe4.com/weak-password-test)
- Password Exposure Test (https://www.knowbe4.com/password-exposure-test)
- Browser Password Inspector (https://www.knowbe4.com/browser-password-inspector)

CONCLUSION

Passwords are likely to be with us for some time, likely a decade or more. The types of attacks against passwords (e.g., social engineering, theft, guessing, hash cracking, bypassing, etc.) have not changed that much over the last two decades. However, so many passwords are successfully stolen, guessed, and cracked each year that moving to another, better form of authentication is needed. But until passwords are completely replaced, you need to create and manage your personal and professional passwords to minimize risk of a successful attack. This can be done by using a combination of phishing-resistant MFA, password managers, and unique passwords for every site and service. KnowBe4's free password tools can help you better manage your password situation and incorporate its findings into your risk profile.

Additional Resources

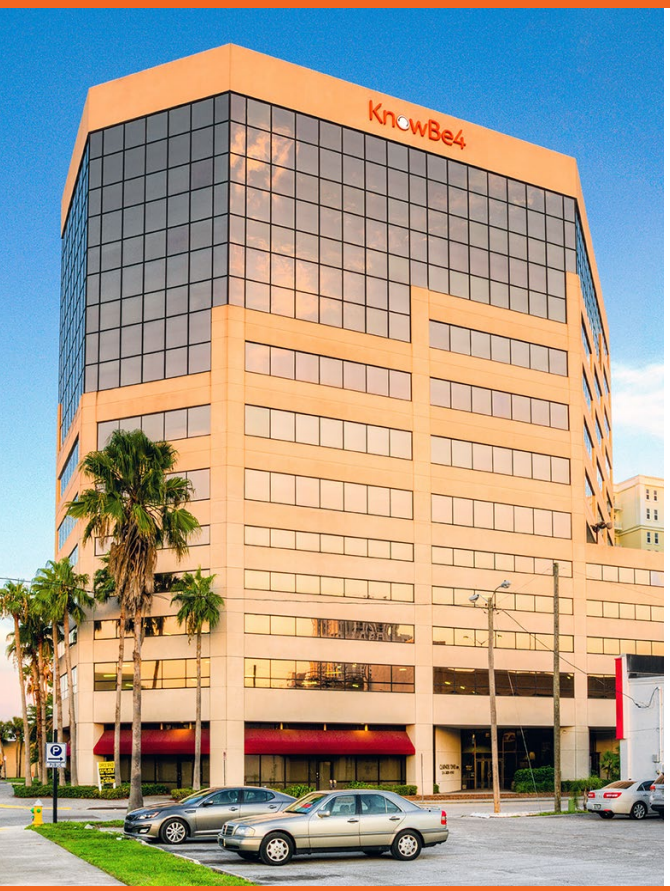
Free Phishing Security Test
Find out what percentage of your employees are Phish-prone with your free Phishing Security Test

Free Automated Security Awareness Program
Create a customized Security Awareness Program for your organization

Free Phish Alert Button
Your employees now have a safe way to report phishing attacks with one click

Free Email Exposure Check
Find out which of your users emails are exposed before the bad guys do

Free Domain Spoof Test
Find out if hackers can spoof an email address of your own domain



About KnowBe4

KnowBe4 is the world's largest integrated security awareness training and simulated phishing platform. Realizing that the human element of security was being seriously neglected, KnowBe4 was created to help organizations manage the ongoing problem of social engineering through a comprehensive new-school awareness training approach.

This method integrates baseline testing using real-world mock attacks, engaging interactive training, continuous assessment through simulated phishing, and vishing attacks and enterprise-strength reporting, to build a more resilient organization with security top of mind.

Tens of thousands of organizations worldwide use KnowBe4's platform across all industries, including highly regulated fields such as finance, healthcare, energy, government and insurance to mobilize their end users as a last line of defense and enable them to make smarter security decisions.

For more information, please visit www.KnowBe4.com

KnowBe4
Human error. Conquered.

KnowBe4, Inc. | 33 N Garden Ave, Suite 1200, Clearwater, FL 33755
Tel: 855-KNOWBE4 (566-9234) | www.KnowBe4.com | Email: Sales@KnowBe4.com

01B11K02