CIS Password Policy Guide

Contents

1. **Contributing/Providing Feedback** 1
   - 1.1 Join Our Community 1
2. **Introduction** 2
   - 2.1 Multi-Factor Authentication (MFA) 3
3. **Recommendation Overview** 4
4. **How Important is a User's Password?** 6
   - 4.1 Online: Password Guessing/Spraying 7
   - 4.2 Offline: Password Cracking/Brute Force 8
5. **Recommendation Details** 10
   - 5.1 Key Recommendations 10
     - 5.1.1 Password—or better yet, Passphrase—Length 10
     - 5.1.2 Password Composition/Complexity 11
     - 5.1.3 Password Expiration 12
     - 5.1.4 Password Banning (Deny Lists) 13
     - 5.1.5 Session Lock When Idle 14
     - 5.1.6 Limit Failed Login Attempts (Lockout) 15
     - 5.1.7 Monitor Failed Login Attempts 16
     - 5.1.8 Suspend Accounts on Non-Use 17
     - 5.1.9 Password Hints 17
   - 5.2 Optional Recommendations 18
     - 5.2.1 Password Strength Indicators on Creation 18
     - 5.2.2 Password Display 18
     - 5.2.3 Password Managers 19
     - 5.2.4 Allow Paste 20
6. **Multi-Factor Authentication** 21
   - 6.1 What is MFA and Why is it Important? 21
   - 6.2 Concerns with MFA 22
7. **Summary/Conclusions** 25
8. **Appendix: What Makes a Good Password?** 26
   - 8.1 Good Password Basics 26
   - 8.2 More Advanced Topics 27
9. **References** 29
10. **Bibliography** 30

1. Contributing/Providing Feedback

The Center for Internet Security, Inc.® (CIS®) is an independent and trusted nonprofit cybersecurity partner of public and private organizations around the world. Our consensus-based best practices provide organizations of all sizes with specific and actionable recommendations to enhance cyber defenses.

CIS harnesses the power of collaboration from a global community of cybersecurity experts to safeguard public and private organizations against cyber threats. This document has been developed with input from a variety of contributors who have helped shape the result.

We wish to thank everyone who helped create this CIS Password Policy Guide.

Editor
H. Philip White

Contributors
Eric Pinnell
Jen Jarose
Curt Dukes
Phyllis Lee
James Globe
Justin Burr
Curt Maughs
Richard Vargas
Lee Myers
Stephen Jensen
Johnathan Christopherson
Kevin Zhang
James Trigg

1.1 Join Our Community

CIS is constantly looking to grow our community of contributors for this project, and the many other guidance projects currently in development. To get involved, first get a free account on CIS WorkBench (https://workbench.cisecurity.org/). From here you can access the CIS Password Policy community (https://workbench.cisecurity.org/communities/113) and many other technology communities (Windows, Linux, Network Devices, Cloud Providers, etc.).

In these communities you can ask questions and provide feedback that will be viewed and discussed not only by CIS personnel, but with the broader community of Subject Matter Experts (SMEs) who develop CIS guidance.

If you have any questions or need help getting involved, please send an email to: benchmarkInfo@cisecurity.org

We thank you for being part of our community, and look forward to your contributions!

2. Introduction

Passwords are ubiquitous in modern society. If you have an account on a computer system, there will likely be at least one password that will need to be managed.

Passwords have been used in computer systems since the earliest days of computing. The Compatible Time-Sharing System (CTSS), an operating system introduced at MIT in 1961, was the first computer system to implement a password-based login.

Passwords are the easiest form of computer security to implement, and there have been many variations. Over the years, security experts have tried to make passwords harder to crack by enforcing various system specific rules on the creation and use of passwords (referred to as Password Policy in this document).

As easy as they are to implement however, the wide variety of password policies have had a debatable effect on the overall security of computer systems. What is not in debate is that they have often led to confusion and frustration for users. Some of the larger players in the Information Technology (IT) standardization area (NIST, Microsoft, etc.) have recently developed new password policies based on two primary principles:

1. Leveraging real-world data on how attackers work
2. Making it easier for users to create, remember, and use secure passwords (the human factor).

The goal of this document is to consolidate this new password guidance in one place. Ideally, a single comprehensive password policy can serve as a standard wherever a password policy is needed.

This document has been created using the same methods and communities that are used to develop and maintain the CIS Controls® and CIS Benchmarks™ standards, including additional real-world input from the CIS-managed Multi-State Information Sharing and Analysis Center® (MS-ISAC®) and Elections Infrastructure Information Sharing and Analysis Center® (EI-ISAC®). It is not the intention here to reinvent the wheel, but rather to apply standards and existing documented best practices in a single source.

This guidance was not created to focus on the password itself, but the overall goal of what a password is. Passwords provide strong user authentication and help to keep attackers out of systems. However, even the strongest password requires other protections to be in place to be most effective (e.g. Multi-Factor Authentication (MFA), account lockouts, account monitoring, etc.). MFA is a highly effective security technique, and even though there is a section devoted to it (see 6. Multi-Factor Authentication), it deserves a special mention here.

2.1 Multi-Factor Authentication (MFA)

MFA should be the first choice for all authentication purposes. This is especially true for administrator access or other privileged accounts. We also recognize that MFA in general does not make password policy irrelevant, since implementing MFA can be a technical challenge that many systems do not support. In addition, even when using MFA, having a reasonable password policy is useful, since a password is used as one of the MFA factors (see 6. Multi-Factor Authentication).

In general, authentication methods rank from best to worst as follows:

1. **MFA:** This should be the goal for all user authentication where possible.
2. **Password Manager:** These tools generate and store unique lengthy/complex passwords per account, and can greatly increase the security and usability of passwords when supported.
3. **Human generated/remembered passwords:** This guide is primarily for this case, since it is not going away anytime soon. However, much of the guidance is relevant for all three of these methods.

3. Recommendation Overview

The following are CIS recommendations for a Password Policy in a tabular form for easy review. You can find more details on each of the recommendations in 5. Recommendation Details.

Control	Description	CIS Recommendation
KEY RECOMMENDATIONS		
Password Length (Min)	This is the system enforced minimum number of characters in a valid password.	**PW Only Account:** 14 Characters – Encourage and teach Passphrase use. **MFA Account (PW Factor):** 8 Characters
Password Length (Max)	This is the system enforced maximum number of characters in a valid password.	No limit.
Password Composition	This is the system enforced character makeup of a valid password (allowing or disallowing certain character types, or numbers of certain character types).	Allow all character types in a password. **PW Only Account:** Require at least 1 non-alphabetic character. **MFA Account (PW Factor):** No composition requirement.
Password Expiration	This is the system enforced number of days a password remains valid (forces a password change).	Change immediately based on events, with a one-year expiration "backstop" (annual).
Password Banning	This is the system enforced check on new password creation against an internal deny list of known bad, weak, or recently used passwords.	Top 20 or more common bad passwords checked on new password creation. **Previously Used PW List:** Last 5 or more. **Password Change Delay:** 1 day or more.
Session Lock When Idle	This is the system enforced duration before locking the current session when it is idle (no user activity).	Set to 15 minutes of idle time or less and the Session Lock login should be the same type as the normal account login.
Limit Failed Login Attempts	This is the system enforced login delay or account lockout based on consecutive bad login attempts.	Temporary account lockout (15 minutes or more) after 5 consecutive failed attempts or time doubling throttling (in minutes) between each retry (0, 1, 2, 4, 8, etc.) with a permanent account lockout (IT reset required) after 12 retries.
Monitor Failed Login Attempts	Log and continuously monitor bad login attempts.	Alert key personnel when above login limit is reached.
Suspend Accounts on Non-Use	Suspend the account if it is not being used.	Automatically suspend the account after 45 days without a valid login.
Password Hints (Login)	This is the system allowing user defined password "hints" at login.	No

Control	Description	CIS Recommendation
OPTIONAL RECOMMENDATIONS		
Password Strength Indicator	This is a system feature (many times a 1–10 measurement) showing the strength of the password.	Provide some form of password strength indication on creation.
Password Display	Allowing a display of the password the user is entering.	**On creation:** Allow display of entire password. **On entry:** Allow temporary display of each character as entered.
Allow Password Managers	Use of external password management products.	Yes, encouraged especially in cases where users need to manage strong passwords on multiple accounts.
Allow Paste	Allowing a cut/copied password to be pasted into password fields.	Yes, only to facilitate Password Manager usage in some scenarios.

The overall goal of an effective password policy is to allow users to easily make reasonably hard to guess passwords for system access and then monitor and limit access attempts to detect/prevent misuse.

4. How Important is a User's Password?

Due to their ubiquity in accessing computer systems of all types, it is clear that passwords are important. But is there a tradeoff between security and usability? In trying to develop password policies to make the resulting password more secure, has the system become less secure because of human behavior? Alex Weinert's (Microsoft) seminal article "Your Pa$$word doesn't matter"[1] summarized some of the common myths around password strength, human users, and real-world attacks shown in the following table.[2]

Attack	Frequency	Difficulty: Mechanism	User assists attacker by	Does password matter?
Credential Stuffing (Breach replay, list cleaning)	Very high: +20M accounts probed daily in MSFT ID systems	Very easy: Purchase creds gathered from breached sites with bad data at rest policies, test for matches on other systems. List cleaning tools are available.	Being human: Passwords are hard to think up. 62% of users admit reuse.	No: Attacker has exact password.
Phishing (Man-in-the-middle, credential interception)	Very high: 0.5% of all inbound emails.	Easy: Send emails that promise entertainment or threaten, and link user to doppelganger site for sign-in. Capture credentials. Use Modlishka or similar tools to make this very easy.	Being human: People are curious or worried and ignore warning signs.	No: User gives the actual password to the attacker.
Keystroke logging (Malware, sniffing)	Low	Medium: Malware records and transmits usernames and passwords entered, along with everything else, so attackers have to parse things.	Clicking links, running as administrator, not scanning for malware.	No: Malware intercepts exactly what was typed (actual password).
Local discovery (Dumpster diving, physical recon, network scanning)	Low	Difficult: Search user's office or journal for written passwords. Scan network for open shares. Scan for creds in code or maintenance scripts.	Writing passwords down (driven by complexity). Hardcoding passwords in code repositories.	No: Actual password discovered.
Extortion (Blackmail, insider threat)	Very low: Cool in movies though.	Difficult: Threaten to harm or embarrass human account holder if credentials aren't provided.	Being human.	No: Actual password disclosed.
Password spray (Guessing, hammering, low-and-slow)	Very high: Accounts for at least 16% of attacks. Millions of accounts probed daily. Sometimes +100K accounts broken per day.	Trivial: Use easily acquired user lists, attempt the same password over a very large number of usernames. Regulate speed and distributed across many IPs to avoid detection. Tools are readily and cheaply available.	Being human: Using common passwords such as qwerty123 or Summer2018!	No: Unless it is in the handful of top passwords attackers are trying.
Brute force (Database extraction, cracking)	Very low	Varies: Penetrate network to extract files. Can be easy if target organization is weakly defended (e.g. password only admin accounts). More difficult if appropriate defenses of database, including physical and operation security, are in place. Perform hash cracking on password. Difficulty varies with encryption used.	None	No: Unless you are using an unusual password (and therefore, a password manager) or a really creative passphrase.

Of the common attacks Alex listed above, only two are affected by password strength:

- **Online:** Password Guessing/Spraying
- **Offline:** Password Cracking/Brute Force

Let's look at these in more detail.

4.1 Online: Password Guessing/Spraying

Password guessing or hammering is where an attacker uses common passwords, such as those found on the internet, against a single user account. Password spraying (also known as the low-and-slow method) is a variant of a brute force attack, where an attacker uses these same password lists, but targets many publically available, or easily determined (i.e. common account name format), user accounts.

In the password guessing case, it seems that password strength would matter here, but in reality limiting failed logins (see 5.1.6 Limit Failed Login Attempts (Lockout)) and login monitoring (see 5.1.7 Monitor Failed Login Attempts) are much more important. With reasonable limits and monitoring in place, the fact that a modern CPU can issue billions of password guesses a second really does not matter, since the account will get locked out and someone notified after only handful of tries. Avoiding account lockouts is why password spraying has become more a more prevalent form of attack, but for it to be effective:

1. The target ususally has a large user base to spread the attack over, since retrying the same account too many times will trigger a lockout (see above).
2. The target usually has a well-known User ID format, since the attackers will typically target a large number of accounts.

Even in these cases a stronger password is not the best solution. Instead a longer, more complex and more diverse set of User IDs is much more effective and much easier for users to manage.[3] In addition, having "Canary Accounts" (valid minimal access accounts that follow the User Id policy, but are never meant to be accessed) is another way to detect password spraying type attacks (login attempts on these accounts).

4.2 Offline: Password Cracking/Brute Force

This is the only case where the strength of a given password makes a difference. In this case, an attacker somehow got a hold of a target company's account/password database, and assuming that this database is using some form of hashed passwords (instead of plain text passwords which would be trivial) now these passwords need to be cracked by one of the many freely available tools that do this (i.e. John the Ripper or L0phtCrack) or a special program designed by the attacker to do this.

We are not going to go into detail about how these programs work (there are many resources available on password cracking[4]), but at a high-level, the attacker can do some of the following:

1. Build a Cracking Rig: Standard computer equipment with a high end graphics card (GPU) can easily generate and try a few billion common hashes (MD5, SHA1, NTLM, etc.) a second. Readily accessible crypto mining rigs can easily achieve 100 billion hashes per second, and well-funded attackers (nation states) can easily do 100 to 1000 times higher rates than this.
2. Taking the crypto mining rig case, and assuming 96 easily typed characters, we get 96 possible values for each position in the password. The time to mindlessly try all possibilities are:

Password Length	Possible Permutations	Time in Hours
6	782,757,789,696	0.002
7	75,144,747,810,816	0.21
8	7,213,895,789,838,340	20.04
9	692,533,995,824,480,000	1,923.71
10	66,483,263,599,150,100,000	184,675.73

3. Unfortunately, there are ways to speed this process up even further:
   a. Research the target to figure out the hash algorithm and any organization-specific password rules (min/max length, complexity, etc.).
   b. Use password lists obtained from previous breaches (on the order of 500M passwords which currently can be freely obtained), then salt and hash these passwords to see if they match any hashes in the target database. Statistically this will break approximately 70% of user passwords.
   c. If this does not work, the attacker can build a list of all popular phrases, song lyrics, news headlines, top search engine trends, Wikipedia, popular articles, etc. These lists are available pre-canned in various hash-breaker communities. This may pick up another 5-7% of user passwords.

   d. Finally, the attacker can use predictable patterns (e.g. always start with a capital letter, follow with 3-6 lower case letters, 2–4 numbers and add an exclamation mark at the end) to create a higher probability subset of guessable passwords out to perhaps 12 characters. This will pick up another 5–7% of user passwords.

4. Because of the salt, all that was for ***one*** account in the database, but with close to an 85% probability of success in a relatively short period of time. The attacker must now start over at step two for the next account whose password they want.

Some additional points to make here:

1. This approach only works if the attacker has the target's account/password database. How did the attacker get this? If the attacker has access enough to get this, then the target is likely already owned.
2. If it is not the target's account/password database, then the cracked password still needs to be tried against an actual account on the target's system (see 4.1 Online: Password Guessing/Spraying).
3. Realistically, a human is highly unlikely to make a strong password that can withstand a dedicated attacker's attempts to crack it (length, complexity, etc.). If this is imperative, then use a long/complex machine-generated password like those created and managed by a password manager program (see 5.2.3 Password Managers).
4. Password cracking capability is ever-increasing, so are we going to keep making passwords longer, more complex and harder to remember to try and keep up for this single use case? It seems like a more comprehensive approach is a better way to go.

Password strength above a somewhat trivial level does not matter very much when it comes to cracking passwords. So why not have a policy that encourages reasonably strong passwords that are easy for users to create, remember, and use?

This guide is designed to provide that.

5. Recommendation Details

This section will discuss each of the previous recommendations in more detail.

5.1 Key Recommendations

These recommendations should be considered mandatory for all systems, where they are technically feasible to implement.

5.1.1 Password—or better yet, Passphrase—Length

Allow a long passphrase, but don't enforce a long passphrase.

In keeping with the overall goal of having users create a password that is not overly weak, an eight-character minimum password length is recommended for an MFA account, and 14 characters for a *password only* account. In addition, maximum password length should be made as long as possible based on system/software capabilities and not restricted by policy.

In general, it is true that longer passwords are better (harder to crack), but it is also true that forced password length requirements can cause user behavior that is predictable and undesirable. For example, requiring users to have a minimum 16-character password may cause them to choose repeating patterns like ***fourfourfourfour*** or ***passwordpassword*** that meet the requirement but aren't hard to guess.[5] Additionally, length requirements increase the chances that users will adopt other insecure practices, like writing them down, re-using them or storing them unencrypted in their documents.

Having a reasonable minimum length with no maximum character limit increases the resulting average password length used (and therefore the strength).[6]

5.1.1.1 Passphrases

Teaching users techniques like passphrases will result in longer, more secure passwords.

A passphrase uses a series of words that may or may not include spaces; *correcthorsebatterystaple* is the example passphrase in the famous XKCD comic on the subject (https://xkcd.com/936/). Although passphrases often contain more characters than passwords do, passphrases contain fewer "components" (four words instead of, say, 12 random characters).[7,8]

In the end, it is all about length versus ease to remember, so teaching users tricks like passphrases which allow them to create longer and easier to remember passwords is a good practice.

NOTE For more detail on what makes a good password passphrase please see Appendix: What Makes a Good Password?

Recommendation summary:

Password Only Account	In cases where a password is the only authentication on a given account, then it is prudent to use a minimum length to 14 characters, and teach users to use passphrases. Maximum length should be as long as possible based on system constraints. **NOTE** This type of account should be targeted to migrate to using MFA and away from password-only authentication where possible.
MFA Account: Password as a Factor	In cases where a password is used as one of the factors in an MFA system, then the 8-character minimum length is considered sufficient. It is prudent to implement the other guidance in this document as well (especially account lockout and monitoring). Maximum length should be as long as possible based on system constraints.

5.1.2 Password Composition/Complexity

Allow the use of any type character, and limit enforcing specific characters types.

Password composition or complexity requirements are often used to increase the strength of a user-created password of a given length. For example, a complex password would need some amount of characters from all three of the following categories:

- Uppercase characters
- Lowercase characters
- Non-alphabetic characters
  - Numbers
  - Special characters like <*&(^%$>!

There is no standard for password composition in use today, so it is very common for these requirements to vary from system to system (e.g., system one allows special characters, but system two does not).

Password composition requirements are a poor defense against guessing attacks.[2, 9] Forcing users to choose some combination of upper-case, lower-case, numbers, and special characters has a negative impact. It places an extra burden on users and many will use predictable patterns (for example, a capital letter in the first position, followed by lowercase letters, then one or two numbers, and a "special character" at the end). Attackers know this, so dictionary attacks will often contain these common patterns and use the most common substitutions like, $ for s, @ for a, 1 for l, 0 for o.

Passwords that are too complex in nature make it harder for users to remember, leading to bad practices. In addition, composition requirements provide no defense against common attack types such as social engineering or insecure storage of passwords.

Recommendation summary:

Password Only Account	In cases where a password is the only authentication on a given account, allow any character to be included in the password and required at least one non-alphabetic character (Number or "Special Character"). **NOTE** Requiring at least one non-alphabetic character increases the search space beyond pure dictionary words, which makes the resulting password harder to crack.
MFA Account: Password as a Factor	In cases where a password is used as one of the factors in an MFA system, allow any character to be included in the password with no complexity requirement.

5.1.3 Password Expiration

Change passwords based on events, with an annual "backstop."

Excessive password expiration requirements do more harm than good, because these requirements make users select predictable passwords, composed of sequential words and numbers that are closely related to each other.[10] In these cases, the next password can be predicted based on the previous one (incrementing a number used in the password for example). Also, password expiration requirements offer no containment benefits because attackers will often use credentials as soon as they compromise them.

Instead, immediate password changes should be based on key events including, but not limited to:

1. Indication of compromise
2. Change of user roles
3. When a user leaves the organization.

Not only does changing passwords every few weeks or months frustrate the user, it's been suggested that it does more harm than good, because it could lead to bad practices by the user such as adding a character to the end of their existing password.

In addition, we also recommend a yearly password change. This is primarily because for all their good intentions users will share credentials across accounts. Therefore, even if a breach is publicly identified, the user may not see this notification, or forget they have an account on that site. This could leave a shared credential vulnerable indefinitely. Having an organizational policy of a 1-year (annual) password expiration is a reasonable compromise to mitigate this with minimal user burden.

NOTE Although outside the scope of this document, there are organizations that use machine-generated one-time use passwords for every account access. In these cases, the password for the given account may change many times per day. This type of system is extremely secure, but also very uncommon.

Recommendation summary:

Password Expiration	Event-based password expiration with an annual change as a backstop.

5.1.4 Password Banning (Deny Lists)

Check for known bad passwords.

Organizations should ban the use of common bad passwords. This reduces susceptibility to brute force and password spraying attacks. A few examples of commonly used passwords include; ***abdcefg***, ***password***, ***qwerty***, ***iloveyou*** and ***12345678*** (A more complete list of common passwords can be found at https://en.wikipedia.org/wiki/List_of_the_most_common_passwords).

When processing requests to create or change a password, the new password should be checked against a list that contains values known to be commonly-used, expected, or compromised. For example, the list should include, but is not limited to:

- Passwords obtained from previous breaches
- Dictionary words
- Repetitive or sequential characters (e.g. *aaaaaa*, *1234abcd*)
- Context-specific words, such as the name of the service, the username, and derivatives thereof
- Previously used passwords for this account with a change delay
- If possible, personal identification information for the user (date of birth, surname, etc.)

This check should happen immediately upon password creation. If the user's password fails the deny list check, the user should be notified that the password cannot be used with a brief explanation of why it cannot be used. The user should then be required to input a new password.

Password Deny Lists are a relatively new tool, but are becoming more common as user credentials breaches are becoming more prevalent. For more information on some example tools, please visit the URLs listed below:

- Azure Active Directory Password Protection: https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-password-ban-bad
- NIST Bad Password Check API: https://nist.badpasswordcheck.com/

NOTE Care must be taken here since it is possible to make the deny list so comprehensive that is becomes very difficult for a user to make a valid password.

NOTE In general, previously used passwords for this account should be part of the deny list. In addition, how frequently a password can be changed should be limited to prevent a user from immediately changing the password multiple times to get back to the original one.

Recommendation summary:

Deny list	Should check against the top 20 or more most commonly used bad passwords.
Previously used password	As part of the deny list, the previously used 5 passwords for this account should be included.
Password change delay	A delay of at least one day is needed to prevent rapid consecutive password changes.

5.1.5 Session Lock When Idle

Session lock when idle is a prudent precaution.

There is no benefit to allowing a system or a session to stay active when a user is not actively working. Knowing when a user is active can be achieved via user input detection (keyboard input, mouse movement, etc.), which has historically been the primary way of achieving "lock when idle."

NOTE Logging in after such a lock should follow all the recommendations of a normal login in terms of failed login attempts and monitoring (see 5.1.6 Limit Failed Login Attempts (Lockout)). Also, Session Lock login should be of the same type and the normal account login, so if the normal account login used MFA, then the Session Lock login for this account should use MFA as well, and not a reduced authentication method like password only.

Recommendation summary:

Session Lock When Idle	Lock the user system or current session after 15 minutes of being idle (no user input) and the Session Lock login should be the same type as normal account login.

5.1.6 Limit Failed Login Attempts (Lockout)

To limit password guessing, temporarily lock the account after a predefined number of failed login attempts.

Ignoring cases where the attacker gets a user password via other means (social engineering, insecure password storage, etc.), since password strength is essentially meaningless in these cases, the goal of creating strong passwords is to prevent an attacker from easily guessing the password and gaining access to a targeted account or system. This means the attacker has to try the password on the targeted system, so limiting the number of attempts the attacker has is more important than any other password strength measure. A temporary (15 minute) account lockout after 5 consecutive failed login attempts has proven to be an effective solution against online password guessing and brute force attempts.[3]

Temporary lockout is designed to not put undue burden on users and IT administration when a legitimate user enters in their password incorrectly, but is rather designed to thwart unauthorized attempts. For example, forcing a more permanent account lockout (one requiring IT involvement) within a large multi-time zone organization can cause undue burden with little practical advantage. A reasonable alternative in some environments is to allow a specific number of temporary lockouts, and if that number is exceeded, then permanently lock the account and require IT involvement to unlock.

Another technique that is gaining popularity is *throttling*, which progressively increases the delay before the next login attempt can occur. Throttling techniques can vary but typically look like the example below.

1. First failure, immediate retry allowed
2. Second consecutive failure, one minute wait
3. Third consecutive failure, two minute wait
4. Fourth consecutive failure, four minute wait
5. Fifth consecutive failure, eight minute wait
6. Permanent account lockout (needs IT involvement)

This restricts the number of guesses an attacker may attempt, while giving users multiple opportunities to remember their password. This technique is not as common in current systems as account lockouts based on number of attempts, but is gaining popularity in internet-based systems. Both techniques provide a good balance of security, usability, and reduced IT burden.

The overall goal, with both methods, is it to make it difficult for an attacker to guess a password by trying it against an account, while allowing a user the ability to correct a mistakenly entered password in a reasonable way.

NOTE Regardless of the method used to limit failed login attempts, it must be paired with some form of monitoring and alerting to be successful (see 5.1.7 Monitor Failed Login Attempts).

NOTE Limiting failed login attempts does not prevent testing a password guess via *password spraying* techniques (i.e. using the same password against many different user accounts). Instead, it tends to encourage attackers into using this technique to avoid lockouts, which is an obvious anomaly and detectable via network monitoring (see 5.1.7 Monitor Failed Login Attempts).

Recommendation summary:

Failed login attempts	Temporary account lockout (15 minutes or more) after five consecutive failed attempts.
Throttling	Time doubling throttling (in minutes) between each retry (0, 1, 2, 4, 8, etc.) with a permanent account lockout (IT reset required) after 12 retries.

5.1.7 Monitor Failed Login Attempts

Login monitoring is a must; it is arguably the most important recommendation.

The goal of strong passwords is to prevent unauthorized users (attackers in particular) from gaining access to systems or accounts. Therefore, *logging* is a key component to investigate attempts at gaining access to a user account, whether this be a regular user or an administrator account. To achieve this, at a minimum, failed login attempts must be monitored and key personnel alerted to the events.

The following is what is suggested, at a minimum, to ensure failed logon attempts are monitored:

- Log all failed login attempts
- Alert key personnel when a temporary or permanent account lockout has been triggered
- Log and alert key personnel about login attempts from unexpected geographical areas
- Log and alert key personnel about login attempts at unexpected times
- Log and alert key personnel about login attempts on "Honeypot Accounts"

NOTE This monitoring can take many forms depending on the type of system involved. Domain/enterprise connected systems can be centrally monitored by using some form of Security Information and Event Management (SIEM) system, which merges and monitors event logs from connected systems. At the other end of the spectrum, a single system or Internet of Things (IoT) device can use something as basic as an email or SMS text to tell a user that the limit has been exceeded. The overall goal is to have the correct person be contacted when failed logon attempts happen.

Recommendation summary:

Monitor failed login attempts	Alert key personnel when above-login limit is reached. **NOTE** This recommendation is one of the most effective ways to detect and prevent unauthorized account access.

5.1.8 Suspend Accounts on Non-Use

Unused accounts need to be automatically turned off.

It would be ideal if administrators would immediately disable accounts for people who are no longer authorized (left the company, changed departments, etc.). Unfortunately, this is not always the case, so it makes sense to have a back-up in case this doesn't happen. Suspending an account after X days of non-use (we suggest 45 days) can act as that back-up plan. If a user has not logged into that account within 45 days of the last valid login, the system will automatically disable the account. The user can get it re-enabled, but is required to contact IT to reinstate it and justify why the account is still needed.

Recommendation summary:

Suspend accounts on non-use	Automatically suspend the account after 45 days without a valid login.

5.1.9 Password Hints

Do not allow password hints.

Password hints can allow users to self service when they cannot remember their password, but the risk can outweigh the benefit. There is no known reliable way to ensure the hint supplied by the user isn't too obvious or easily obtained (social engineering: Facebook, Twitter, etc.), and can allow an attacker easy access to the system using this method. A better approach is to allow for an easy to remember password (passphrase) when it is created.

Recommendation summary:

Password hints at login	Do not use.

5.2 Optional Recommendations

CIS considers these recommendations optional and may be addressed after implementing the key recommendations in Section 5.1. These recommendations are more environment-specific (for example if a user does not manage multiple passwords, then there is less of a need for a password manager).

5.2.1 Password Strength Indicators on Creation

Strength indicators are helpful, since most people want to make a strong password.

When creating a new password, the system should offer guidance to the user, such as a password-strength indicator, to assist the user in creating a strong password. This is particularly useful when used with password deny lists (see 5.1.4 Password Banning (Deny Lists)), since it guides the user to creating a stronger password that is likely not on the deny list.

Recommendation summary:

Password strength indicator	Although these are by no means perfect, providing some form of this is useful. They have been shown to increase password strength, and decrease user frustration when creating a new password.

5.2.2 Password Display

There are two main use cases for password display:

5.2.2.1 On Password Creation

Allowing a user to display their password on creation is better than a confirmation field.

To assist the user in creating a password, the system should offer an option to display the password, instead of a series of dots or asterisks, until they enter it. This allows the user to verify their entry if they are in a location where their on-screen password is unlikely to be seen. This works much better than a blind confirmation field for mistyped passwords.

5.2.2.2 On Password Use

Allowing a user to briefly see what they are typing in a password field reduces entry errors.

The system should optionally permit the user's device to display individual entered characters for a short time after they type each character to verify correct entry (then replaced with an asterisk or dot). This can be particularly useful on mobile devices where the text fields are small and hard to see.

Recommendation summary:

Password Display	**On creation:** Allow display of entire password. **On entry:** Allow temporary display of each character as entered.

5.2.3 Password Managers

Encouraging the use of an approved password manager lets users create strong passwords that are not reused on multiple systems.

A password manager is like a book of a user's passwords, locked by a master key that only that user knows. On the surface that might sound bad. What if someone gets the user's master password? That's a reasonable fear; but assuming the user has chosen a strong, unique, and memorable master password they're not using anywhere else, or better yet MFA, password managers are effective. Like anything else in IT security, passwords managers aren't 100% fail safe, but they are a great alternative for users who need to manage multiple strong passwords for different accounts. It reduces reusing the same password for multiple accounts, storing passwords in plain text on their system, or writing them down and storing them in an unsecure location.

In addition to password managers storing user passwords, they also help users create and save strong, unique passwords. This means whenever users go to a website or application, they can pull up their password manager, copy their password, paste it into the login box. Often, password managers come with browser extensions that can fill in a saved user's password for them in a secure manner.

If a password manager (preferably one) is used, it is suggested that organizations restrict this to a small approved list of software that provide features the organization needs. This will make it easier to maintain the software (upgrades), patches, and track any published vulnerabilities and their mitigations.

NOTE System-generated passwords created by a password manager are much stronger than human-created passwords because they use a sequence of characters with greater minimum length and composition (complexity) requirements. Human users do not have to memorize the password. Instead, the password manager stores them for the user.

NOTE Logging into an approved password manager should follow all the recommendations of a normal system login in terms of failed login attempts and monitoring (see 5.1.6 Limit Failed Login Attempts (Lockout)).

NOTE Password Managers are designed to remember the entire login credentials for an account. This is generally a user name and password. This means that a user can also create a complicated unique username for any given account and the Password Manager will remember it for the user. This makes any breached credentials even less useful to the attacker, since not even the username will be the same on another account. Of course this assumes the target system allows some flexibility in the username composition.

Recommendation summary:

Password Managers	Use of these should be actively encouraged for use with password-only authentication systems (especially if the user needs to manage access to multiple of these systems). **NOTE** Where feasible, using MFA instead of just a master password to gain access to the Password Manager is preferred.

5.2.4 Allow Paste

Allow Paste in password fields when using a password manager.

It is recommended that systems permit users to use the paste functionality when entering a password, since this facilitates the use of password managers (see 5.2.3 Password Managers).

The main fear companies have with allowing paste to enable a Password Manager is that passwords are stored in the clipboard. Yes, this is true; when a user accesses the copy/paste function, the copied content is kept in a clipboard where it can be pasted as many times as they want. Any software installed on the computer (or any person operating it) has access to the clipboard and can see what has been copied. However, most password managers erase the clipboard as soon as they have pasted the password into the website, and some avoid the clipboard altogether by typing in the password with a virtual keyboard. These features can be part of the Password Manager selection criteria.

Recommendation summary:

Allow paste	Paste should be allowed in cases where password manager use is used. **NOTE** The primary point here is using a password manager is much more secure even if paste needs to be enabled for some application(s) to work properly.

6. Multi-Factor Authentication

In this section we will discuss that passwords alone are insufficient for strong account security. Instead, implementing some form of Multi-Factor Authentication is desirable.

6.1 What is MFA and Why is it Important?

Passwords alone are not the best security solution.

MFA, sometimes referred to as Two-Factor Authentication (2FA), is a security enhancement that allows the user to present two, or more, pieces of evidence (referred to as factors) when logging in to an account. MFA has proven to be a successful way to help with account compromises. This is due to the fact that the attacker needs to gain multiple pieces of information from the user instead of one. This proves to be troublesome for attackers and they generally don't compromise MFA accounts. MFA factors can fall into any of these three categories:

- **Something You Know:** A password or Personal Identification Number (PIN)
- **Something You Have:** A smart card, security token, an authentication application or a Short Message Service (SMS) text to the user's mobile phone
- **Something You Are:** A fingerprint or retina pattern

The user's factors must come from two different categories to enhance security, so entering two different passwords would not be considered multi-factor authentication.

MFA is the most secure user authentication method available on the market today, and this additional security measure has minimal impact on usability.

NOTE "Two-step" or "multi-step" authentication is not the same as 2FA or MFA. "Two-step" or "multi-step" authentication involves the subsequent presentation of one or more additional authentication steps to the target system after the first step is successfully performed. Each of these steps may or may not have a different authentication factor involved. In essence, each step is an independent "gate" and success in each step gets the user closer to the goal of target system access. 2FA or MFA authentication is a stronger approach that involves the presentation of all the factors simultaneously to form one credential that the target system checks for validity. The target system passes or fails the credential as a whole with no indication of what factor failed. It is possible to use 2FA or MFA as one of the steps in a "two-step" or "multi-step" authentication process.

6.2 Concerns with MFA

Complexity to implement is the primary issue.

MFA is great tool and it is gaining usage in a wide variety of systems and accounts. For example, most web-based banking applications today support 2FA via an authentication application or an SMS text to the user's mobile phone. It is not yet as ubiquitous or as standardized as password usage. Some additional points with MFA are:

1. In many cases, MFA involves a password as a factor, so a good password policy still applies.
2. In the vast majority of cases, MFA is an add-on to the Website, Application, or Operating System developed by a third party. It is suggested that organizations limit MFA add-ons to a small approved list of software (preferably one) that provide the features the organization needs. This will make it easier to maintain this software (upgrades) and track any published vulnerabilities and their mitigations and patches.
3. *Something You Know* is considered the weakest factor and *Something You Are* the strongest.
4. *Something You Know* is considered the most cost effective and easiest to implement and *Something You Are* the most expensive and hardest to implement (need a physical reader).
5. *Something You Have* is the most common form 2FA in use today (combined with a password). The most common forms of this method are:
    a. **Phone call, email, or SMS text:** The user will login to a system with a username and password then they will be presented with the requirement to enter a unique code as the second factor. The system will send this unique code to the user's predefined phone number (voice call), email, or mobile number (SMS). In some cases, the user may select which medium is used, but in all cases the code will expire after a set time period or once entered.
    b. **An authentication application on a user's mobile device:** From a user perspective this is very similar to the SMS method, except the unique code number is generated by an application on the user's mobile device. There is a growing number of these applications and not all of them are interoperable:
        i. **Google:** https://support.google.com/accounts/answer/1066447
        ii. **Lastpass:** https://lastpass.com/auth/
        iii. **Microsoft:** https://www.microsoft.com/en-us/account/authenticator
        iv. **Authy:** https://authy.com/
        v. **RSA:** https://www.rsa.com/en-us/products/rsa-securid-suite/rsa-securid-access/securid-software-tokens

c **A Physical Security Token:** There are two common token types:

i **Unique ID**: From a user perspective this is the same as using an Authentication application, except the unique code is generated and displayed on a separate physical device:

- **RSA:** https://www.rsa.com/en-us/products/rsa-securid-suite/rsa-securid-access/securid-hardware-tokens
- **Fortinet:** https://www.fortinet.com/products/identity-access-management/fortitoken-200.html

ii **Physical Presence:** These devices are designed to be installed, or close, to the device (via USB or NFC) and indicate the person's presence. Some of these also incorporate a Unique ID via an accompanying application:

- **Yubico Yubikey:** https://www.yubico.com/products/
- **Google Titian Security Keys:** https://store.google.com/product/titan_security_key_kit

Two Factor authentication methods are more secure than passwords alone, but they each have their downfalls that should be considered:

1 *Something You Know* methods are basically passwords and should follow the password policy guidance in Section 5.

2 Of the *Something You Have* methods, authentication applications and physical tokens are considered very secure, but SMS texts have issues:

a SMS texts depend on the Public Switched Telephone Network (PSTN), which is outside the control of the user or your IT administration.

b The PSTN is essentially a bunch of networked computers sending voice calls and SMS texts to each other via a protocol known as Signaling System 7 (SS7). This protocol has known, previously exploited, vulnerabilities to re-direct SMS text messages to an attacker's phone.

c In more mundane cases, attackers have used social engineering and/or website hacking techniques to convince a target's mobile service provider's support personnel that the user's phone was lost or damaged and to assign the number to a new phone (the attacker's).

NOTE Due to these issues we considered not recommending SMS as a 2FA method, but because of its ubiquity combined with it arguably being better than nothing, it remains. We strongly urge anyone using SMS for 2FA to develop a plan to move off this method as soon as feasible.

3 *Something You Are* methods are considered by many to be the most secure, because they are based on physical attributes of a given user that are impossible to change (Finger Print, Retina pattern, etc.). This seems like the perfect mechanism, but there are many issues with using them as the primary means of authentication, and they should be paired with a secret non-biometric attribute (like a password).[11] Taking a fingerprint as an example:

   a A user's fingerprint, like all biometrics, is not secret like a password or private encryption key and can be captured by anyone following the user around. The biggest problem with a non-secret authentication factor is that they are easy to copy for malicious reuse.

      i For example, in June 2015 more than 5.6 million U.S. fingerprint records were stolen by a Chinese group. Anyone who had ever applied for a U.S. security clearance had their fingerprints stolen.

   b Whenever a user's fingerprint is scanned it must be matched with a stored fingerprint for verification. This stored fingerprint data is a representation of the actual fingerprint and not truly unique.

      i For example, a user's fingerprint is turned into series of points noting where major peaks, valleys, and sharp changes happen. These large deviations are marked with a point, with the overall fingerprint being stored and evaluated looking much more like a star constellation than a real fingerprint.

   c How would the user go about changing this? Unfortunately, it is not like a password that can easily be changed, or a Unique ID that is valid for 5 minutes, once a fingerprint has been breached, it's permanent.

The goal here is to inform users of the pros and cons for some of the MFA methods, since all methods of MFA have their place in IT security. In the end, making the decision to implement MFA will greatly depend on the level of security required, budget, and policies and procedures put in place by decision makers.

7. Summary/Conclusions

The overall goal of this document is to consolidate much of the modern password guidance into a single concise document, and give the real-word reasons why these recommendations are being made. It is a compilation of significant previous work (see 10. Bibliography), and the open community consensus process CIS uses for all its guidance work.

CIS understands that not every system can implement all these recommendations because of various system limitations, but over time, CIS expects that systems will support most, if not all, of these recommendations. None of the recommendations here are technically difficult to implement, and each of them have current real-world implementations, but finding a single application or system that supports all of them is difficult.

For specifiers or purchasers of these types of systems, this document is intended as a guide to define the password policy aspects of an organization. For system developers, this guide can be used for the type of support that may be needed.

CIS believes that this guidance will become a common reference for other security standards and specifications, and over time achieve the password policy commonality users are looking for.

8. Appendix: What Makes a Good Password?

This appendix covers some common thoughts on password creation. It is not meant to be a definitive reference, but more of a guide based on some best practices. Passwords are always going to be an imperfect security feature, but they are essential in helping to secure systems when used properly. The CIS Password Policy Guidance combined with user education can make passwords a reliable and secure tool. To help facilitate user education, here are some key points:

8.1 Good Password Basics

Here are some straightforward concepts to make better passwords. Keep in mind we are not trying to make them impenetrable, but strong.

1. Length is the most important characteristic of a good password: In general, the longer the password, the better.
2. Think pass-phrase, not pass-word: If you think of a single "word," it is difficult to some up with something long and memorable, but if you think of a "phrase" made up of 4 or more smaller words it is much easier.
    a. 14 or more character words:
        i. ***Trichomoniasis***, ***Antidepressant***, ***Fundamentalism***, ***Attractiveness***, etc.
        ii. None of these are very fun to remember, let alone spell correctly
    b. 14 or more character phases (with and without spaces for readability):
        i. With spaces: ***My Aunt Lives in Georgia***; Without: ***MyAuntLivesinGeorgia***
        ii. With spaces: ***The Ford Mustang is the Best***; Without: ***TheFordMustangistheBest***
        iii. With spaces: ***Cape Cod is a Fun Place***; Without: ***CapeCodisaFunPlace***
3. Avoid patterns: Do not use sequences of numbers letters or keyboard patterns like:
        i. ***12345671234567***, ***abcdefgabcdefg***, ***passwordpassword***, ***abc123abc123ab***, ***qwertyuqwertyu***, etc.

4. Don't reuse a password or use similar passwords on multiple systems: Especially between home and work accounts. The primary reason is if someone discovers one of your passwords, you do not want them to now be able to access multiple of your accounts.
    i. This is arguably the toughest of the four basic ideas, but you can use tricks to help, like bands/songs/movies/actors to help create a relevant and memorable phrase:
        - Financial account: With spaces: ***Pink Floyd Money***; Without: ***PinkFloydMoney***
        - Store account: With spaces: ***Superstore Cloud Nine***; Without: ***SuperstoreCloudNine***
        - Medical account: ***MASH Hawkeye Pierce***; Without: ***MASHHawkeyePierce***

8.2 More Advanced Topics

Now, you want to go beyond the basics and make some truly spectacular passwords!

1. Avoid words related to your personal information or common interests: Avoid things people can look up about you on the internet. If you are the president of the local Mustang car club, you should probably not use "Mustang" as a password.
    a. With that said, would the previous example of ***TheFordMustangistheBest*** be okay to use? There are certainly better choices, but the word "Mustang" only makes up 7 of the 23 characters of the passphrase (16 characters are still unknown), so it is still not bad!
2. Limit using dictionary words: In general, the way adversaries attack passwords is by trying various combinations of words in the dictionary first. This is a lot of words to try, but it is A LOT less than trying all the possible letter combinations.
    a. For example, let's take a 3 character password. Assuming only lower-case letters there are (26)3 = 17,576 letter combinations (including things like zxy, rhb, qqt, etc.), but there are much less valid 3 letter words (1,355 via one online dictionary). This is less than one tenth as many.
    b. Things get much worse as you add characters:
        i. 8 Characters – All Combinations: 208,827,064,576
        ii. 12 Characters – All Combinations: 95,428,956,661,682,176
        iii. 16 Characters – All Combinations: 43,608,742,899,428,874,059,776
    c. Unfortunately there are just not that many valid words to keep up:
        i. 3 Character words: 1,355
        ii. 4 Character words: 3,503
        iii. 5 Character words: 6,308

iv 6 Character words: 10,050

v 7 Character words: 12,722

vi 8 Character words: 14,136

vii 9 Character words: 14,044

viii 10 Character words: 11,868

ix 12 Character words: 6,284

x 13 Character words: 4,131

xi 14Character words: 2,506

xii 15 Character words: 1,435

xiii Total Words available (above): 88,342 – Let's call it 100,000

d Hopefully, it is plain to see that trying dictionary words is much faster than trying all the possible letter combinations for a given length password.

e Using multiple dictionary words in your passphrase will make the adversary's work harder as well, but if you really want give them a tough time throw in some stuff not in the dictionary. For example:

i Instead of ***TheFordMustangistheBest***, try ***TheFordMustangis#1!***

ii Instead of ***PinkFloydMoney***, try ***Pink$Floyd$Money$***

iii Instead of ***MASHHawkeyePierce***, try ***M.A.S.H.Hawkeye-Pierce!***

f Another thing you can do is replace some characters with number representations. For example:

i e/E ? ***3***; a/A? ***@***; o/O ? ***0***

ii TheFordMustangis#1! ? ***Th3F0rdMust@ngis#1***

iii Pink$Floyd$Money$? ***Pink$Fl0yd$M0n3y$***

iv M.A.S.H.Hawkeye-Pierce! ? ***M.@.S.H.H@wk3y3-Pi3rc3!***

Doing each of these things will make your password somewhat stronger and a bit more resistant to a determined attacker cracking it, but with diminishing returns. Why? Because the adversaries know people use these tricks as well, and take this into account in their password cracking programs.

If you want a truly great password, you need a long one randomly generated and stored by a password manager because you will likely not remember the result, which will look something like this:

- ***GHj*65%789JnF4$#$68IJHr54^78***

In the end, we strongly recommend moving using Multi-Factor Authentication (MFA) wherever feasible, since it eliminates the total reliance on passwords for account security.

9. References

1 Weinert A. (2019, July 09). Your Pa$$word doesn't matter. Retrieved from: https://techcommunity.microsoft.com/t5/azure-active-directory-identity/your-pa-word-doesn-t-matter/ba-p/731984

2 Microsoft. (2020, January 23). Password policy recommendations for Office 365. Retrieved from: https://docs.microsoft.com/en-us/office365/admin/misc/password-policy-recommendations?view=o365-worldwide

3 Florencio D., et al. (2007). Do Strong Web Passwords Accomplish Anything? Retrieved from: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2007-64.pdf

4 Kelley P., et al. (2011). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. Retrieved from: https://www.archive.ece.cmu.edu/~lbauer/papers/2012/oakland2012-guessing.pdf

5 Komanduri S, et al. (2014, Aug) Telepathwords: Preventing Weak Passwords by Reading Users' Minds. Retrieved from: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/TelepathwordsUSENIX2014.pdf

6 Shay R., et al (2010, July) Encountering Stronger Password Requirements: User Attitudes and Behaviors. Retrieved from: https://cups.cs.cmu.edu/soups/2010/proceedings/a2_shay.pdf

7 Lampe J. (2014, January 06). Beyond Password Length and Complexity. Retrieved from: https://resources.infosecinstitute.com/beyond-password-length-complexity/

8 Wolford B. (2019, March 05). Let's settle the password vs. passphrase debate once and for all. Retrieved from: https://protonmail.com/blog/protonmail-com-blog-password-vs-passphrase/

9 Mourouzis T., et al (2018) The Evolution of User-Selected Passwords: A Quantitative Analysis of Publicly Available Datasets. Retrieved from: https://arxiv.org/ftp/arxiv/papers/1804/1804.03946.pdf

10 Zhang Y., et al (2010) The Security of Modern Password Expiration: An Algorithmic Framework and Empirical Analysis. Retrieved from: https://www.cs.unc.edu/~reiter/papers/2010/CCS.pdf

11 Grimes R. (2019, January 04). 6 reasons biometrics are bad authenticators (and 1 acceptable use). Retrieved from: https://www.csoonline.com/article/3330695/6-reasons-biometrics-are-bad-authenticators-and-1-acceptable-use.html

10. Bibliography

1 Microsoft. (2020, January 23). Password policy recommendations for Office 365. Retrieved from: https://docs.microsoft.com/en-us/office365/admin/misc/password-policy-recommendations?view=o365-worldwide

2 Microsoft. (2019, May 22). Security baseline (FINAL) for Windows 10 v1903 and Windows Server v1903. Retrieved from: https://docs.microsoft.com/en-us/archive/blogs/secguide/security-baseline-final-for-windows-10-v1903-and-windows-server-v1903

3 Grassi P, et al. (2017, June). NIST Special Publication 800-63B, Digital Identity Guidelines, Authentication and Lifecycle Management. Retrieved from: https://pages.nist.gov/800-63-3/sp800-63b.html

4 NCSC. (2018, November 19). Password administration for system owners. Retrieved from: https://www.ncsc.gov.uk/collection/passwords/updating-your-approach

5 Gov of Canada. (2019, April 01). . Password Guidance. Retrieved from: https://www.canada.ca/en/government/system/digital-government/password-guidance.html

6 Cobb, M. (2012, June 01). Password security best practices: Change passwords to passphrases. Retrieved from: http://www.computerweekly.com/tip/Password-security-best-practices-Change-passwords-to-passphrases

7 Robarts, S. (2014, April 17). Three alternatives to using passwords. Retrieved from: http://www.gizmag.com/three-alternatives-passwords/31695/

8 Ross, J. (2014, January 28). How to Change Employees' Poor Password Habits. Retrieved from: https://iapp.org/news/a/how-to-change-employees-poor-password-habits

9 Walters, R. (2015, April 07). Insecure Passwords or Insecure People? Retrieved from: http://www.infosecurity-magazine.com/opinions/insecure-passwords-insecure-people/

10 Warman, M. (2013, January 15). 90% of passwords 'vulnerable to hacking'. Retrieved from: http://www.telegraph.co.uk/technology/news/9802062/90-of-passwords-vulnerable-to-hacking.html

11 Hicock R. (2016, May). Password Guidance. Retrieved from: https://www.microsoft.com/en-us/research/publication/password-guidance/?from=http%3A%2F%2Fresearch.microsoft.com%2Fpubs%2F265143%2Fmicrosoft_password_guidance.pdf

12 Hunt T. (2018, February 06). How Long is Long Enough? Minimum Password Lengths by the World's Top Sites. Retrieved from: https://www.troyhunt.com/how-long-is-long-enough-minimum-password-lengths-by-the-worlds-top-sites/

13 Xavier J. (2019, January 04). The Ultimate Guide to Strong Passwords in 2019. Retrieved from: https://blog.fleetsmith.com/password-security-guide/

14 Varghese B. (2019, April 30). Passwords vs Passphrases: Which Is Better For Your Cyber Security?. Retrieved from: https://health.usf.edu/is/blog/2019/04/30/Passwords-vs-Passphrases-Which-Is-Better-For-Your-Security

15 Sacha B. (2017, January 12). Let them paste passwords. Retrieved from: https://www.ncsc.gov.uk/blog-post/let-them-paste-passwords

16 Lamppu S. (2019, January 31). Active Directory – Banned Passwords. Retrieved from: https://samilamppu.com/2019/01/31/active-directory-banned-passwords/

17 Design Smarts. (Accessed 2020, January 27). Why you should allow users to see their password. Retrieved from: https://designsmarts.co/show-password/

18 Wroblewski L. (2015, January 22). Showing Passwords on Log-In Screens. Retrieved from: https://www.lukew.com/ff/entry.asp?1941

19 Dunn J. (2019, February 21). Password managers leaking data in memory, but you should still use one. Retrieved from: https://nakedsecurity.sophos.com/2019/02/21/password-managers-leaking-data-in-memory-but-you-should-still-use-one/

20 O'Flaherty K. (2019, February 20). Password Managers Have A Security Flaw -- Here's How To Avoid It. Retrieved from: https://www.forbes.com/sites/kateoflahertyuk/2019/02/20/password-managers-have-a-security-flaw-heres-how-to-avoid-it/#11d2954c4e16

21 Microsoft. (2014, October 23). Passwords Technical Overview. Retrieved from: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/hh994558(v=ws.10)

22 NIST. (2019, December 09). Back to basics: Multi-factor authentication (MFA). Retrieved from: https://www.nist.gov/itl/applied-cybersecurity/tig/back-basics-multi-factor-authentication

23 Goodin D. (2017, May 03). Thieves drain 2fa-protected bank accounts by abusing SS7 routing protocol. Retrieved from: https://arstechnica.com/information-technology/2017/05/thieves-drain-2fa-protected-bank-accounts-by-abusing-ss7-routing-protocol/

24 Akers A. (2019, January 19). SMS Intercept Attacks and Why SMS Multi-Factor Still Matters. Retrieved from: https://auth0.com/blog/why-sms-multi-factor-still-matters/

25 Armstrong K. (2019, September 11). Why it might be time to ditch SMS for MFA. Retrieved from: https://www.hipaaone.com/2019/09/11/why-it-might-be-time-to-ditch-sms-for-mfa/

26 Doffman Z. (2019, October 07). FBI Issues Surprise New Cyber Attack Warning: Multi-Factor Authentication Is Being Defeated. Retrieved from: https://www.forbes.com/sites/zakdoffman/2019/10/07/fbi-issues-surprise-cyber-attack-warningurges-new-precautions/#78e30c9c7efb