--- Recap Password Management

jeudi 17 octobre 2024 7:01 PM

Password Policies

Now that we have worked through numerous ways to capture credentials and passwords, **let us cover some best practices related to passwords and identity protection**.

Let us meet Mark, a new employee for Inlanefreight Corp. Mark, does not work in IT, and he is not aware of the risk of a weak password. He needs to set his password for his business email. He picks the password password123. However, *he gets an error saying that the password does not meet the company password policy* and *a message that lets him know the minimum requirement for the password to be more secure.*

In this example, we have two essential pieces, a **definition of the password policy** and **the enforcement**.

The **definition** is a guideline, and the **enforcement** is the technology used to make the users comply with the policy. **Both aspects** of the password policy implementation **are essential**. During this lesson, we will explore both and understand how we can create an effective password policy and its implementation.

Password Policy Standards

Some security standards include a section for password policies or password guidelines. Here is a list of the most common:

- **NIST SP800-63B**
- **CIS Password Policy Guide**
- **PCI DSS**

We can use those standards to understand different perspectives of password policies. After that, we can use this information to create our password policy. Let us take a use case where **different standards** use a different approach, password expiration.

Change your password periodically (e.g., 90 days) to be more secure may be a phrase we heard a couple of times, but the **truth is that not every company is using this policy**. Some companies only require their users to change their passwords when there is evidence of compromise. If we look at some of the above standards, some require users to change the password periodically, and **others do not.** We should stop and think, **challenge the standards and define what is best for our needs.**

Password Policy Recommendations

Let us create a sample password policy to illustrate some important things to keep in mind while creating a password policy. Our sample password policy indicates that all passwords should:

- `Minimum of 8 characters.`
- `Include uppercase and lowercase letters.`
- `Include at least one number.`
- `Include at least one special character.`
- `It should not be the username.`
- `It should be changed every 60 days.`

Our new employee, Mark, who got an error when creating the email with the password password123, now picks the following password **Inlanefreight01!** and successfully registers his account. **Although this password complies with company policies**, it is not secure and **easily guessable** because it uses the company name as part of the password. We learned in the "**Password Mutations**" section that this is a common practice of employees, and attackers are aware of this.

Once this password reaches the expiration time, **Mark can change 01 to 02**, and his **password complies with the company password policy**, but **the password is nearly the same**. Because of this, security professionals have an open discussion about password expiration and when to require a user to change their password.

Based on this example, we must include, as part of our password policies, **some blacklisted words**, which include, but are not limited to:

- `Company's name`
- `Common words associated with the company`
- `Names of months`
- `Names of seasons`
- `Variations on the word welcome and password`
- `Common and guessable words such as password, 123456, and abcde`

Enforcing Password Policy

A password policy is a guide that defines **how** we should **create**, **manipulate** and **store passwords** in the organization. **To apply** this guide, **we need to enforce it**, using the technology at our disposal or acquiring what needs to make this work. Most applications and identity managers provide methods to apply our password policy.

For example, if we use Active Directory for authentication, we need to configure an **Active Directory Password Policy GPO**, to enforce our users to comply with our password policy.

Once the technical aspect is covered, we need to communicate the policy to the company and create processes and procedures to guarantee that our password policy is applied everywhere.

Creating a Good password

Creating a good password can be easy. Let's use **PasswordMonster**, a website that helps us test how strong our passwords are, and **1Password Password Generator**, another website to generate secure passwords.

CjDC2x[U was the password generated by the tool, and it is a good password. It would take a long time to crack and would likely not be guessed or obtained in a password spraying attack, but it is tough to remember.

We can create good passwords with ordinary words, phrases, and even songs that we like. Here is an example of a good password ***This is my secure password*** or ***The name of my dog is Poppy***. We can combine those passwords with special characters to make them more complex, like **()** The name of my dog is Poppy!. Although hard to guess, we should keep in mind that attackers can use OSINT to learn about us, and we should keep this in mind when creating passwords.

With this method, we can create and memorize 3, 4, or more passwords, but as the list increases, it will be difficult to remember all of our passwords. In the next section, we will discuss using a Password Manager to help us create and maintain the large number of passwords we have.

Password Managers

According to this **NordPass study**, the **average person has 100 passwords**, which is one of the reasons that most people reuse passwords or create simple passwords**.**

A **password manager** is an application that **allows users to store their passwords** and secrets ***in an encrypted database.***

In addition to keeping our passwords and sensitive data safe, they also have features to **generate** and **manage** **robust and unique passwords**, **2FA**, **fill web forms**, **browser integration**, **synchronization between multiple devices**, **security alerts**, among other features.

How Does a Password Manager Work?

The implementation of password managers varies depending on the manufacturer, but most work with a master password to encrypt the database.

The **encryption** and **authentication** work using different **Cryptographic hash functions** and **key derivations functions**, **to prevent unauthorized access to our encrypted password database** and **its content**. The way this works depends on the manufacturer and if the password manager is offline or online.

Let's break down the common password managers and how they work.

Online Password Managers

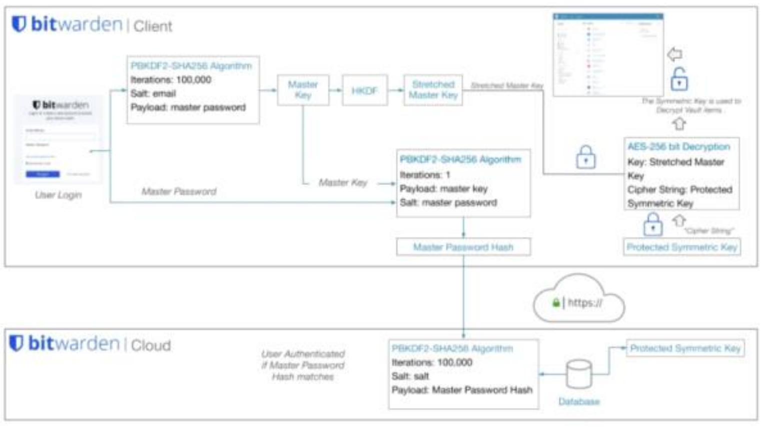
One of the key elements when deciding on a password manager is convenience. A typical person has 3 or 4 devices and uses them to log in to different websites, applications, etc. An online password manager allows the user to synchronize its encrypted password database between multiples devices, most of them provide:

- A mobile application.
- A browser add-on.
- Some other features that we'll discuss later in this section.

All password manager vendors have their way of managing their security implementation, and they usually provide a technical document that describes how it works. You can check **Bitwarden**, **1Password** and **LastPass** documentation as a reference, but there are many others. Let's talk about how this generally works.

A common implementation for online password managers is deriving keys based on the master password. Its purpose is to provide a **Zero Knowledge Encryption**, which means that no one, except you (not even the service provider), can access your secured data. To achieve this, they commonly derive the master password. Let us use Bitwarden's technical implementation for password derivation to explain how it works:

1. **Master Key:** created by some function to turn the master password into a hash.
2. **Master Password Hash:** created by some function to turn the master password with a combination of the master key into a hash to authenticate to the cloud.
3. **Decryption Key:** created by some function using the master key to form a Symmetric Key to Decrypt Vault items.



This is a simple way to illustrate how password managers work, but the common implementation is more complex. You can check the technical documents above or watch the How Password Managers Work - Computerphile video.

Most popular online password managers are:

- 1Password
- Bitwarden
- Dashlane
- Keeper
- Lastpass
- NordPass
- RoboForm

Local Password Managers

Some companies and individuals prefer to manage their security for different reasons and not rely on services provided by third parties. Local password managers offer this option by storing the database locally and putting the responsibility on th e user to protect their content and the location where it is stored. **Dashlane** wrote a **blog post Password Manager Storage: Cloud vs. Local** which can help you discover the pros and cons of each storage. The blog post states, "At first it might seem like this makes local storage more secure than cloud storage, but cybersecurity is not a simple discipline." You can use this blog to start your research and understand which method would better serve the different scenarios where you need to manage passwords.

Local password managers encrypt the database file using a master key. The master key can consist of one or multiple components: a master password, a key file, a username, password, etc. Usually, all parts of the master key are required to access the database.

Local password managers' encryption is similar to cloud implementations. The most noticeable difference is data transmission and authentication. To encrypt the database, local password managers focus on securing the local database using different cryptographic hash functions (depending on the manufacturer). They also use the key derivation function (random salt) to avoid precomputing keys and hinder dictionary and guessing attacks. Some offer memory protection and keylogger protection using a secure desktop, similar to Windows User Account Control (UAC).

The most popular local password managers are:

- **KeePass**
- **KWalletManager**
- **Pleasant Password Server**
- **Password Safe**

Features

Let's imagine we use Linux, Android, and Chrome OS. We access all of our applications and websites from any device. We want to synchronize all passwords and secure notes across all devices. We need extra protection with 2FA, and our budget is 1USD monthly. That information may help us identify the correct password manager for us.

When deciding on a cloud or local password manager, we need to understand its features, Wikipedia has a list of password managers (online and local) as well as some of their features. Here's a list of the most common features for password managers:

1. 2FA support.
2. Multi-platform (Android, iOS, Windows, Linux, Mac, etc.).
3. Browser Extension.
4. Login Autocomplete.
5. Import and export capabilities.
6. Password generation.

Alternatives:

Passwords are the most common way of authentication but not the only one. As we learn from this module, there are multiple ways to compromise a password, cracking, guessing, shoulder surfing, etc., but what if we don't need a password to log in? Is such a thing possible?

By default, most operating systems and applications do not support any alternative to a password. Still, administrators can u se 3rd party identity providers or applications to configure or enhance identity protection across their organizations. Some of the most common ways to secure identities beyond passwords are:

1. Multi-factor Authentication.
2. FIDO2 open authentication standard, which enables users to leverage common devices like Yubikey, to authenticate easily. For a more extended device list, you can see Microsoft FIDO2 security key providers.
3. One-Time Password (OTP).
4. Time-based one-time password (TOTP).
5. IP restriction.
6. Device Compliance. Examples: Endpoint Manager or Workspace ONE

Passwordless

Multiples companies like Microsoft, Auth0, Okta, Ping Identity, etc, are trying to promote the Passwordless strategy, to remove the password as the way of authentication.

Passwordless authentication is achieved when an authentication factor other than a password is used. A password is a knowledge factor, meaning it's something a user knows. The problem with relying on a knowledge factor alone is that it's vulnerable to theft, sharing, repeat use, misuse, and other risks. Passwordless authentication ultimately means no more passwords. Instead, it relies on a possession factor, something a user has, or an inherent factor, which a user is, to verify user identity with greater assurance.

As new technology and standards evolve, we need to investigate and understand the details of its implementation to understand if those alternatives will or not provide the security we need for the authentication process. You can read more about Passwordless authentication and different vendors' strategies:

1. Microsoft Passwordless
2. Auth0 Passwordless
3. Okta Passwordless
4. PingIdentity

There are many options when it comes to protecting passwords. Choosing the right one will come down to the individual or company's requirements. It is common for people and companies to use different password protection methods for various purposes.

À partir de l'adresse <https://academy.hackthebox.com/module/147/section/1333>

.