Multi-factor Authentication:
Best Practices for Securing the Modern Digital Enterprise

WHITE PAPER

TABLE OF CONTENTS

INTRODUCTION

MFA has been increasing in popularity over the years, to the point where it is a rare person who does not use some form of MFA in their daily life. There are many different kinds of MFA, however, and many different scenarios where it may be used. Companies looking to adopt MFA to better serve their employees, partners and customers must ensure they choose an authentication model that provides the greatest security, usability and cost-effectiveness possible.

> **Based on our studies, your account is more than 99.9% less likely to be compromised if you use MFA.[1]**
>
> - ALEX WEINERT, GROUP PROGRAM MANAGER FOR IDENTITY SECURITY & PROTECTION AT MICROSOFT

The goal of this paper is to help you deepen your understanding of authentication models by describing the different categories of MFA and the MFA methods within those categories. Along with these descriptions, you'll find details about the strengths and weaknesses of both the categories and the individual methods, and, when appropriate, a description of the scenarios where the method or category may be most useful. The paper concludes with a set of recommendations and best practices.

We hope that you find this information useful as you pick the MFA methods best suited to your specific needs.

1 Catalin Cimpanu, "Microsoft: Using multi-factor authentication blocks 99.9% of account hacks," ZDNet, August 27, 2019, https://www.zdnet.com/article/microsoft-using-multi-factor-authentication-blocks-99-9-of-account-hacks/

MFA DESCRIPTION AND RATIONALE

Traditionally, authentication mechanisms or factors have been categorized as belonging to one of three groups:

1. Something you know (for example, a password or a PIN).
2. Something you have (for example, a mobile phone or a token).
3. Something you are (for example, a fingerprint or other biometric data).

These three authentication categories are the most frequently used. You may also encounter additional classifications such as something you are doing, somewhere you are, or time, but keep in mind that these are essentially elements or sub-factors of the existing three main categories. For example, location or somewhere you are (GPS) is tied to a device or wearable in your presence, which ultimately is something you have.

When authentication factors were first introduced, they added an extra level of assurance that the user was who they said they were. But no factor is foolproof on its own, because each type of factor (and authentication mechanism within that factor) has its own specific strengths and weaknesses. To keep bad actors from exploiting those weaknesses, companies began adopting two-factor authentication (2FA), where a user is required to provide two different factors to authenticate. The goal here was to force an attacker to have to compromise two different channels to take over an account, and this mechanism provided a massive increase in security over previous authentication methods.

Unfortunately attackers did not stand still, and invented a variety of ways to compromise multiple factors. Before we get into how to thwart those attacks, however, let's put these newer risks into context. 2FA—any 2FA, even password+SMS—is so much more secure than username/password that if you have not implemented it, you should stop reading this paper and do so immediately. Using 2FA versus single factor authentication (SFA) is table stakes for authentication in today's environment, as it materially increases the attacker's effort and its rate of compromise is far lower than SFA/password-based authentication alone.

But once you have a baseline level of protection in place, it's time to turn your thinking from 2FA to MFA. Technically MFA just means "multi-factor authentication," and in theory it could simply be the same 2FA that you are already using. Or, theoretically MFA could involve three, four or really any number of factors, chosen from the three basic categories.

In best practice, though, MFA goes beyond 2FA by requiring a user to authenticate via two or more authentication factors from different categories (e.g., a "something you know" combined with a "something you have"), as shown in Figure 1. The goal of having two or more authentication factors from different categories is to reduce the likelihood of an impostor gaining access.

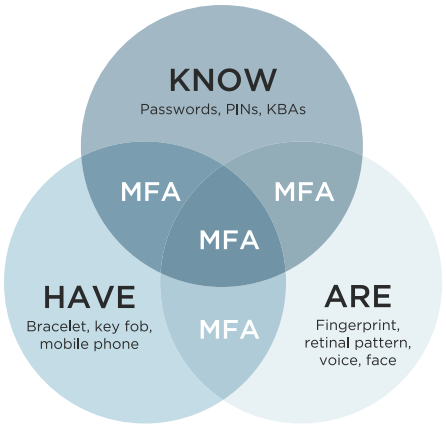

Figure 1: MFA requires users to identify themselves via two or more categories of authentication.

Generally, combining multiple authentication factors from different categories results in a higher level of assurance that the individual attempting to authenticate is actually the individual in question—because even if one of the factors has been compromised, the chances of the other factor also being compromised are low.

Authentication mechanisms can also be distinguished by whether they use the same channel where the user accesses the application or a separate channel (out-of-band) that's dedicated for authentication. At Ping we do not believe that one is more effective than the other, as both human and technology considerations affect the overall security of each.

Mobile Authentication Options

Given the ubiquity of mobile phones and the massive increase in the number of native mobile applications, it is not surprising that more and more authentication options revolve around mobile phones. Mobile phones provide platform-level authentication capabilities built into the phone itself, as well as being an excellent platform on which to run authentication applications.

That is not to say that mobile authentication methods are always the preferred choice. Not everyone has a mobile phone and there are circumstances where the use of mobile phones is not permitted, such as call centers or manufacturing scenarios. Also, some locations have Internet service but not cell service, such as some rural areas or inside large buildings and other structures. If you expect any of these scenarios to apply to your users, you should make sure you provide a multi-factor authentication option that does not depend solely on mobile authentication.

With all this is mind, let's talk about the different authentication categories and methods within those categories.

AUTHENTICATION CATEGORIES: STRENGTHS & WEAKNESSES

Something You Know: Knowledge Factors

Something you know—typically a secret password you initially created and linked to your identity—has been the default method for authenticating to systems since the beginning of secured computers and applications. These secrets can be stored as part of a user's record and quickly compared against a value the user enters during authentication, which is a key reason for their tremendous popularity.

Back before the days of the Internet this was more than secure enough, since applications and systems were only available via direct access to a computer, or perhaps via dial-up. The passwords were also relatively usable, since many systems prescribed a short minimum-password length and limited maximum length to eight characters.

But even then, problems existed. If you forgot your login, how would you ever be able to request a password reset via email? As a result, users made (and still make) lots of password-reset calls to help desks.

The growth of the Internet and the myriad systems that users both want to access and need passwords for showed the limitations of passwords from both a security and usability standpoint. But until we perfect an easy-to-use, low-cost authentication mechanism that's impossible to lose or forget, we expect passwords and other knowledge-based factors to be around for a very long time. So let's take a closer look at the security and usability issues of passwords.

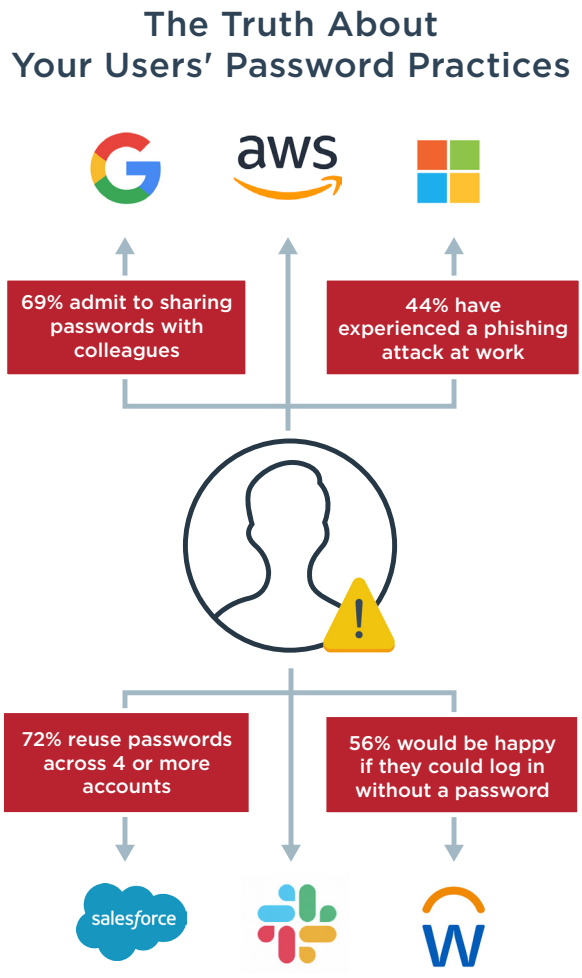

Figure 2: If you're relying on passwords to make sure your users are who they say they are, you're putting your enterprise at undue risk.

Passwords

Password Security

Passwords are the most common knowledge-based factor—and they're notoriously risky. Often it isn't the passwords themselves that are the problem, though, but rather users' password practices. Long and randomly generated single-use passwords are extremely secure, but they're also hard to remember. This is where the poor practices come into play, such as the use of easily guessed passwords like "123456" or "letmein."

Poor practices also include the use of the same password or other knowledge-based information on multiple sites, which opens up even more risk. Even if a site has excellent security, that security is only as good as the weakest system when credentials used across other sites are compromised.

Stolen credentials often end up on the dark web for resale, where they can be obtained by attackers. This has led to the rise of account takeovers via botnet attacks, where stolen lists of usernames and passwords are replayed against websites looking for matches. The attackers don't have to be successful very often since even a 2% success rate means that in a system with 1,000 users, an attacker may be able to compromise 20 accounts. In order to improve password security, best practice is to check new or updated passwords against a list of known stolen or pwned passwords.

Also of importance is how a company manages passwords. Even the hardest-to-guess password can be vulnerable if it isn't hashed with a private seed, stored centrally and concealed from the system where the user is logging in. When passwords are transferred or stored in the clear, they're at increased risk of being stolen and used for account takeover. This doesn't just apply to passwords either. Any knowledge-based factor that an application provider stores is at greater risk of theft.

Password Usability

In terms of usability, passwords are a mixed bag. Simple, easy-to-remember passwords are usable, but in addition to the credential-stuffing attacks mentioned above, they are vulnerable to brute force dictionary or common password list attacks. Long passwords are more secure but become less usable, especially if they must be updated frequently. This lack of usability can result in increased password resets, support calls, password sharing among sites and the other usability and user friction issues we mentioned at the beginning of this paper.

Password Cost

From a cost standpoint, passwords are cheap to implement since most systems will store them by default or place them in a user directory. From an implementation standpoint, they also are rated high as they are a go-to authentication method with pervasive support across the entire software infrastructure.

PIN

A PIN is a shorter version of a knowledge factor. They are typically numeric only and 4 to 6 digits long. Obviously a 4- or 6-digit PIN is much more susceptible to brute force than passwords, as there are only 10,000 or 1,000,000 possible combinations, respectively. As such, a PIN would be a poor choice as the sole factor protecting a website and should never be stored in a central location. But if the PIN is used solely to unlock a mobile application, it would take a long time for someone to brute force the PIN. If a lockout happens after a certain number of bad entries, or increased delays occur between each subsequent entry, security is improved even more.

From an implementation standpoint, PINs are straightforward and inexpensive to implement. Since most users have no issues remembering a 4- or 6-digit PIN, they also score high on usability. In general, PINs are an excellent local-only stored first factor in a multi-factor scenario.

KBA

Another common way to confirm an individual's identity is with knowledge-based authentication (KBA), which requires consumers to provide answers to questions that, theoretically, an attacker would not know. KBA can be used in any scenario a password can, although KBA is most frequently used as a step-up factor for either account recovery or when a device or browser is being used for the first time.

There are two kinds of KBA: shared answers and dynamic KBA. With shared answers, the organization provides a list of questions, and the user provides the answer as part of a registration process. When challenged with the question, the user must provide the correct answer. Typically, if the user enters the wrong answer, they are asked a different question, as repeatedly being asked the same question could make it too easy for an attacker to guess the answer.

Dynamic KBA, on the other hand, challenges the user with questions that have not been pre-shared. Instead, the organization running the KBA or other record sources have gathered this information. The challenge with this method is it is difficult to find questions that are both not public knowledge and reasonably easy to answer. For example, asking someone to remember their exact mortgage payment two refinancings ago is unlikely to result in a positive customer experience.

KBA Security

From a security perspective, KBA is a mixed bag. Many KBA questions are based on information that criminals can easily find on social media sites or through other public sources. And since KBA answers are centrally stored, they must be encrypted and never transmitted in the clear during authentication.

KBA Usability

From a usability standpoint, it's common for a consumer to fail their own KBA quiz, resulting in a negative customer experience. On the plus side, KBA does not require possession of a device or other factor, can be used in both mobile and desktop scenarios, and is based on something the user already knows rather than a random secret. As such, KBA can have its place in scenarios like account recovery.

KBA Cost

Since KBAs are stored and managed in a manner similar to passwords, their cost is similar. The only major differences in cost are in the writing of the software to get the KBA answers from a user at registration and to challenge the user with KBA questions when necessary.

Something You Have: Possession Factors

A possession factor can work great—until you lose or forget it. Because this is a common occurrence, any system that uses a possession factor needs to have a fallback plan. Case in point: A large consulting provider (and Ping customer) reports that on any given day, somewhere between several hundred to more than a thousand employees either lose their phones or forget to bring them to work. Since the enterprise has stringent security requirements for access to sensitive applications, this results in hundreds if not thousands of daily phone calls to the helpdesk to manually validate the employee and issue temporary credentials.

Possession Factor Security

From a security standpoint, possession factors can be extremely effective in preventing remote attacks since the possession factor cannot be spoofed remotely—with the possible exception of SMS, which we will cover in a section below.

Possession Factor Usability

From a usability standpoint, possession factors are rated high. Responding to a push notification on your phone, plugging a FIDO authenticator into a USB port or clicking on a temporary link in an email are all relatively convenient. Which one is the best choice depends on the particular scenario.

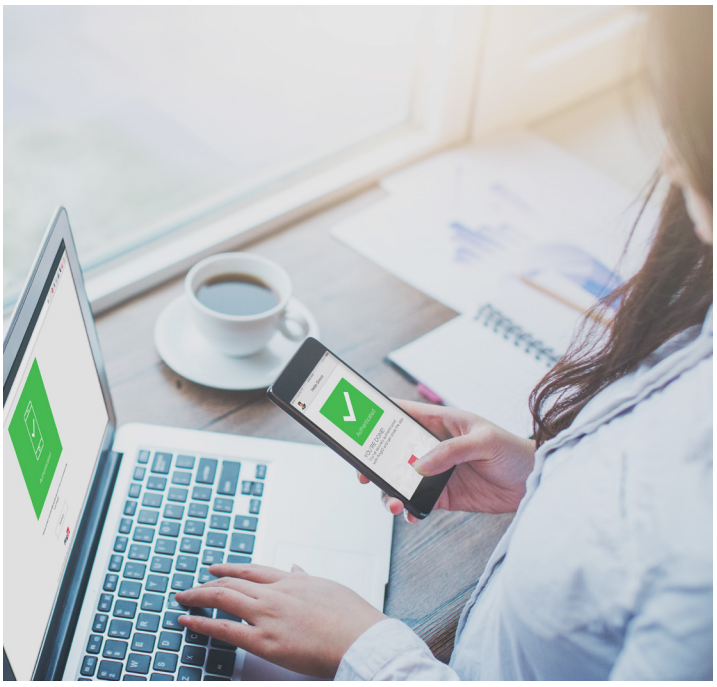

Figure 3: Possession factors like mobile phones allow users to authenticate in multiple ways, including via a mobile app or through pop-up notifications.

Possession Factor Cost

From a cost perspective, possession factors vary widely. There is no real cost if the user already has the device, or if the possession factor is software-based. Dedicated cards or tokens can vary in price from a couple of dollars to as high as $50 for high-end authenticators with built-in biometrics. However, these high-end devices can offer great usability, portability and resistance to phishing attacks. For example, biometric-enabled FIDO authenticators such as Feitian BioPass or YubiKey Bio store your identity and a biometric in the device to confirm your identity at login, allowing true one-touch login.

RSA and OATH Hardware Token

RSA and OATH tokens are small hardware devices that the owner carries to authorize access to a network service. The device may be in the form of a smart card, or it may be embedded in an easily carried object such as a key fob or USB drive. The device itself contains an algorithm (a clock or a counter) and a seed record used to calculate the pseudorandom number, and users enter this number to prove that they have the token. The server that's authenticating the user must also have a copy of each key fob's seed record, the algorithm used and the correct time.

Some hardware tokens are equipped with a USB interface, and these tokens are inserted into the PC's USB slot. When the user needs to authenticate, they press a key on the device, which generates a one-time passcode (OTP) and emulates a keyboard to send the passcode to the server, as if the user had entered it by hand.

Hard FIDO Authentication Tokens

Hard FIDO tokens are also small hardware devices. They can interface with your computer via USB, Near Field Communication (NFC) or Bluetooth Low Energy (BLE). Highly effective against phishing attacks, FIDO authentication requires that the user register the authenticator for each website that they want to authenticate with. The authenticator generates a unique public/private key pair for a specific website and returns the public key to that website. Authentication is only allowed over TLS, and the key is bound to the website's domain. Therefore, if the user is subsequently phished to a fake website, the authentication request will fail since the attacker is not coming from the registered website domain.

Because FIDO keys are accessed through the browser, they depend on the user's choice of browser to support the key on whatever platform the user is currently on. Until recently, the availability of support for FIDO security keys was not consistent across browsers and operating systems. With the release of iOS 13.3, FIDO keys are supported across all major operating systems and browsers, including Chrome, Safari and Edge on Mac, Windows, Android and iOS.

The cost of FIDO authenticators has traditionally been a barrier, and enterprises typically only require and distribute FIDO authenticators for select groups of people and for applications with the highest security needs. But major mobile platforms from Android and Apple are building support for soft FIDO authenticators, which are software-based and can leverage the phone's biometric capabilities such as fingerprint or facial recognition. Because there is no additional cost to use these authenticators, mass adoption of FIDO authentication is much more feasible economically. Soft FIDO authenticators are covered below in the section on biometric authentication.

One-time Passcodes (OTP)

One-time passcodes are the most popular additional security factor today, in part because they can be delivered in a wide variety of ways to meet user needs. This possession factor enables the user to receive the OTP and enter it into an application, proving that the user owns or controls the device or method of OTP delivery. OTPs are time limited, and servers can restrict the number of instances a user can attempt to enter the correct OTP. This makes OTP an effective defense against the online credential stuffing attacks used to compromise passwords.

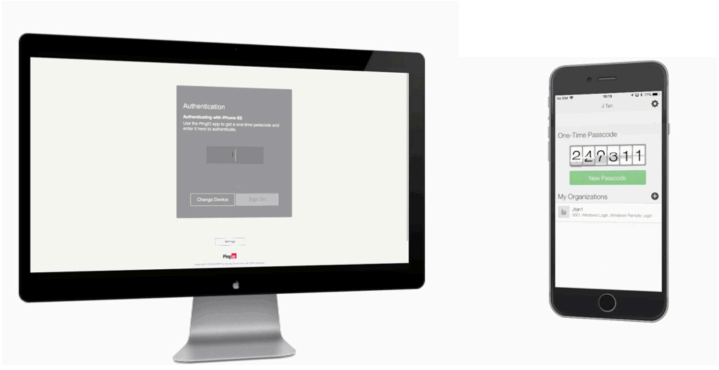

Figure 4: One-time passcodes are popular security factors

One-time passcodes are still vulnerable to phishing attacks, however. Phishing proxies such as Modlishka allow attackers to set up phishing sites using domain names that are similar to real websites. These sites look like the real site since they are actually proxying traffic to and from the legitimate site, and can inspect and change any information they wish in real time. A user who has been convinced that a phishing site is a legitimate site will enter the OTP into the phishing site, which can then use the OTP code to log in to a legitimate site in real time.

The security of one-time passcodes depends on the security surrounding the device or delivery method for OTP. We will cover security considerations of the different OTP methods in the individual sections below, starting with soft tokens.

OTP Application/ Soft Tokens

Soft tokens are a software-only variant of the RSA/OATH tokens. They use the same interface as the hard tokens, so a single server-side implementation can leverage both hard and soft tokens. The software provides a rolling series of OTPs and can run as a mobile or desktop application.

As far as security goes, in practice soft tokens are less vulnerable to loss than hard tokens. Mobile users are more likely to lose a single-use hardware token than they are to forget or lose their phones, and when they do lose a phone, they are more likely to report the loss and the soft token can be disabled. Soft tokens are also easier and less expensive to distribute than hardware tokens, which need to be shipped.

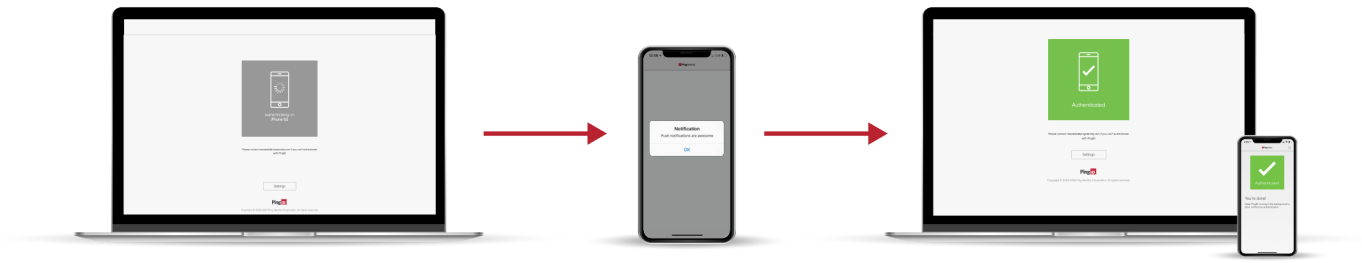

Figure 5: Push notifications can deliver great digital customer experiences.

Push Notification

Push notification provides a second factor by having an authentication service push a notification message to an application installed on the user's smartphone. The capabilities of the applications can vary widely.

While easy to use, you should take some security considerations into account: primarily, training users to not just automatically hit approve whenever a request comes in. Users are often so bombarded with notification requests that they simply approve every one of them. This trait works to the advantage of hackers who have compromised the first factor during a login process, because they know that some percentage of users will submit the second factor on the attacker's behalf when they receive a push notification.

OTP via SMS

SMS OTP is delivered via SMS to a user's mobile phone. While the SMS OTP option has the advantage of not requiring a user to own a modern smartphone that supports mobile applications, it has several disadvantages around number porting and SIM swapping. This has been prevalent enough that NIST has deprecated SMS usage. So while it's a good option to offer consumer-facing OTP via SMS for less security-critical access, organizations should consider whether its security is sufficient for higher-value enterprise logins.

OTP via Voice

One method of OTP delivery is via phone call to a number already associated with a user. This method is highly available as all it requires

is a phone. It will work when no mobile phone is present, or via land line when no cell coverage is present. But from a security perspective, it is vulnerable to SIM swapping if on a mobile device, or potential abuse if a phone is shared among multiple people.

OTP via Email

OTP delivered via email is a viable second factor. It loses usability points since it requires the user to switch to their email application from whatever application they were authenticating to, and either remember the OTP code or copy and paste it into the authenticating application. Because of these limitations, email-based OTP is typically used for resetting forgotten passwords, where the user can prove they own the email account by responding to a time-limited link within the email.

From a security standpoint, email is only as secure as the credentials that are used to gain access to it. You should never allow email to be the backup mechanism for multiple MFA factors. Say, for example, you were to allow email OTP to be the fallback for a forgotten password, as well as the fallback option for a lost device, with the email protected by only a single factor. In this case you wouldn't have true multi-factor authentication, since the attacker simply needs to compromise one thing (the email account) to break into the user's account.

Something You Are: Biometric Factors

Biometric factors are popular as an MFA option and are far and away the most popular passwordless option, in part because they provide unbeatable ease of use. Fingerprint readers are now standard on almost every smartphone and laptop. Windows Hello offers integration with biometric devices, while newer devices such as the iPhone X and the Microsoft Surface Book 2 provide built-in facial recognition features. These platform-provided capabilities can be easily utilized as part of an authentication flow.

Biometric Security

From a security perspective, biometrics score high. Fingerprint readers, for instance, have a negligible false identification error rate. Facial recognition technology has also made enormous strides in accuracy, including the ability to detect sign-on attempts with photographs, masks or video replay.

Biometrics have some weaknesses, however. From a security standpoint, if the biometric is tied to a device, for example, then it is subject to the same forgotten-device issues that a possession factor is. Storing the biometric on a central server eliminates that problem, but you must take care with this type of data as it is considered personally identifiable information (PII) under regulations like GDPR and CCPA. In the United States, Texas, Illinois and Washington have passed laws concerning the collection and sharing of biometric information, and several other states have proposed similar regulations.

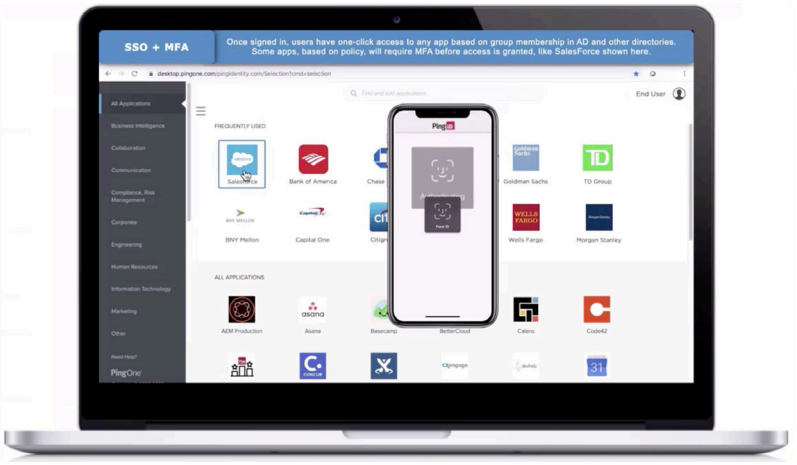

Figure 6: Facial biometrics have been a mainstream authentication option since Apple introduced the Face ID feature in 2017.

Biometric Usability

Usability issues with biometrics also vary widely, depending on the use case. For example, a fingerprint reader would be a poor choice for verifying patients at a flu clinic. Similarly, facial recognition isn't reliable if the lighting cannot be controlled. One study of facial recognition-equipped ATMs found that accuracy fell dramatically in the afternoon for machines facing windows with western exposure, because the reflection of the setting

sun on those windows completely washed out the facial recognition images.

Biometric Cost

Biometrics vary widely from a cost perspective as well. If the platform natively provides the capability, then you will have no added hardware cost in your authentication flows. But if add-on fingerprint readers are required, you can expect to pay $30 to $40 each, and $75 to $150 for add-on cameras that support Windows Hello.

Biometrics and the Platform FIDO Authenticator

As mentioned in the previous section on FIDO authenticators, FIDO provides the best protection by far against phishing attacks, since they will not accept an authentication request from a domain other than previously registered ones. FIDO tokens have not seen mass adoption because of their expense—each one costs roughly between $20 and $50—but the arrival of soft FIDO authenticators is likely to change this.

- Android Version 7 implemented a certified platform FIDO authenticator in the operating system itself. The platform authenticator can be used to access applications via the Chrome browser on the phone. The user registers the FIDO authenticator with websites that support the WebAuthn standard via the browser. When the website requests authentication, the user authenticates to the phone the same way they always do: with PIN, fingerprint or facial recognition. This provides a powerful combination of security and ease of use.
- Apple supports using external FIDO authenticators (hard FIDO tokens) to authenticate via WebAuthn to browser-based applications running on the Mac and iPhone over Bluetooth Low Energy or USB. Safari 14, the version of Apple's browser shipped with iOS 14 and macOS Big Sur, introduced a FIDO platform authenticator that will let you use Face ID or Touch ID on both Macs and iPhones to register your iPhone as a FIDO security key. This allows users to log into websites that support the WebAuthn standard.

These changes promise to make FIDO a globally available, high-security phishing-resistant authentication option for employee usage. And for the first time, it makes it practical for consumer-facing websites to offer FIDO authentication as an option.

CHOOSING THE RIGHT MFA MECHANISMS

Choosing the MFA mechanisms that are best for your particular situation requires balancing security, usability and cost. The table below attempts to summarize the strengths and weaknesses of different MFA mechanisms.

Mechanism	Security	Usability	Cost	Best Application
Passwords	Low to Medium	Medium	Low to No	First factor in standard 2FA
PIN	Low	Good	Low to No	Local-only unlock of computer or device
KBA	Low to Medium	Poor to Medium	Medium	Step-up factor for previously unseen device
RSA/OATH Token	High	Low	High	Possession factor for high-security application, especially with no Internet
FIDO Authenticator	Very High	Medium	High	When you need phishing-resistant authentication that can roam between computers
OTP	High (except SMS)	Medium	Low	Good 2nd factor choice when users do not have dedicated authentication app
Push Notification	Medium to High	High	Low to Medium	Good choice if you can leverage mobile authentication app
QR Code	High	High	Low	For pairing device with account
Platform FIDO Authenticator	Very High	Very High	Low	Best combination of security and usability, becoming universally available
Biometric Authentication	Very High	Generally High	High if not provided by platform	Support is app or OS specific. Good for system login or specific apps

CONTEXTUAL FACTORS AND RISK-BASED AUTHENTICATION

We have been talking a lot about authentication categories and the different mechanisms within those categories. Intelligent MFA, however, is more than just a collection of factors and methods. By adding contextual factors to MFA, we can evolve from treating every authentication the same to being able to better understand the risk around each specific authentication and to tailor the authentication flow to that risk. The premise is to dynamically assess the risk of a given operation based on:

- The user's current authentication status
- The risk associated with the resource in question
- The context of the request

This allows us to provide better usability and greater user convenience by skipping additional authentication factors when the risk is low, e.g., when a user is logging in on a managed device from a frequently used IP address at the same time of day they typically sign on.

Contextual factors also warn us when some aspect of an authentication is not normal; atypical and anomalous context might then trigger step-up authentication. Examples include users logging in for the first time from an unmanaged device, logging in from San Francisco an hour after they logged in from Paris, logging in from an IP address with a poor IP reputation, or attempting to complete a transaction over $100,000. In these cases, it might make sense to simply request additional factors to help increase our trust in the user.

As shown in Figure 7, to be granted access to some resource, a user authenticates with a factor such as a password. At the time of authentication, the system also collects and checks authentication signals. Only if those checks identify something unexpected and anomalous is the user asked to authenticate with the second factor before being granted access. And if the risk is too high, your authentication policy may decide to not allow access at all, or to only allow access with reduced privileges.

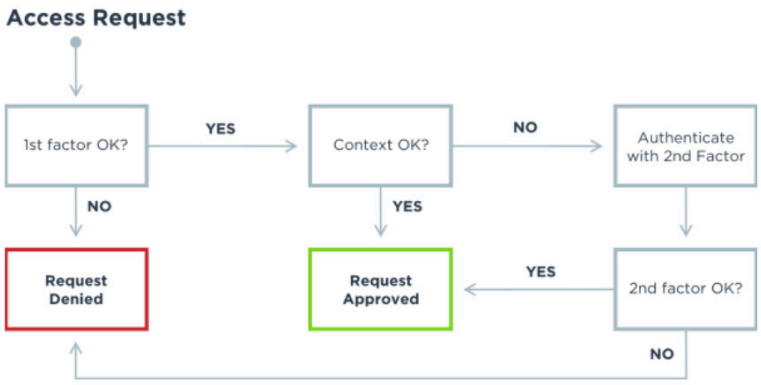

Figure 7: Risk-based step-up MFA is triggered by atypical and anomalous context or behavior. It's only when the context collected via the first authentication factor indicates something unexpected that a second factor of authentication is requested before access is granted.

As a final note, a good MFA solution provides you with a number of already-available contextual and risk factors, and allows you to integrate additional factors into your authentication policies and flows as your needs dictate.

CONCLUSION AND RECOMMENDATIONS

It is clear that enterprises must continue to evolve and improve their authentication of users, moving beyond the limitations of passwords and traditional 2FA. But as stated at the beginning of this paper, 2FA is the floor every organization should be using.

Based on our evaluation of current and emerging technologies, and discussions with customers, partners and other stakeholders, we make the following recommendations:

- Employ phishing-resistant authentication. With MFA in place, the weakest link in the security chain is not the technology, it's the user. Providing phishing-resistant authentication should be a top priority for organizations in order to improve their overall security posture, particularly for high-risk users and systems. We strongly recommend that organizations evaluate and adopt FIDO authentication, either with hard tokens or using the emerging soft platform authenticators.
- Use biometric authentication wherever possible, either in conjunction with FIDO or as provided by your MFA offering. Replacing the "something you know" with "something you are" both significantly reduces authentication friction and can eliminate the far too common password reset flows or calls to help desks to reset locked-out accounts.
- Offer authentication options. The perfect authentication choice is not nearly so perfect if it is not available, causing lost productivity in workforce scenarios and lost revenue in customer-facing scenarios. Your MFA solution should allow users to register multiple authentication tokens or authentication methods.
- Make sure that users cannot use the same recovery channel for lost or forgotten factors. If a user can reset a forgotten password via an email link and at the same time get a one-time code via email if they have forgotten their phone, then you have reduced your security to a single factor: the email account.
- Account for non-mobile scenarios. Provide options that allow MFA in cases where there is no cell service.
- Configure contextual policies where sensible, such as impossible travel or location by persona or group. Also consider using contextual authentication policies to skip requiring multiple authentication factors when the user is on a managed or known device, and other contextual risk factors are low.

We hope you have found this guide to be informative and helpful. For more information on Ping's multi-factor authentication offerings, please visit www.pingidentity.com.

#3001 | 09.20 | v05