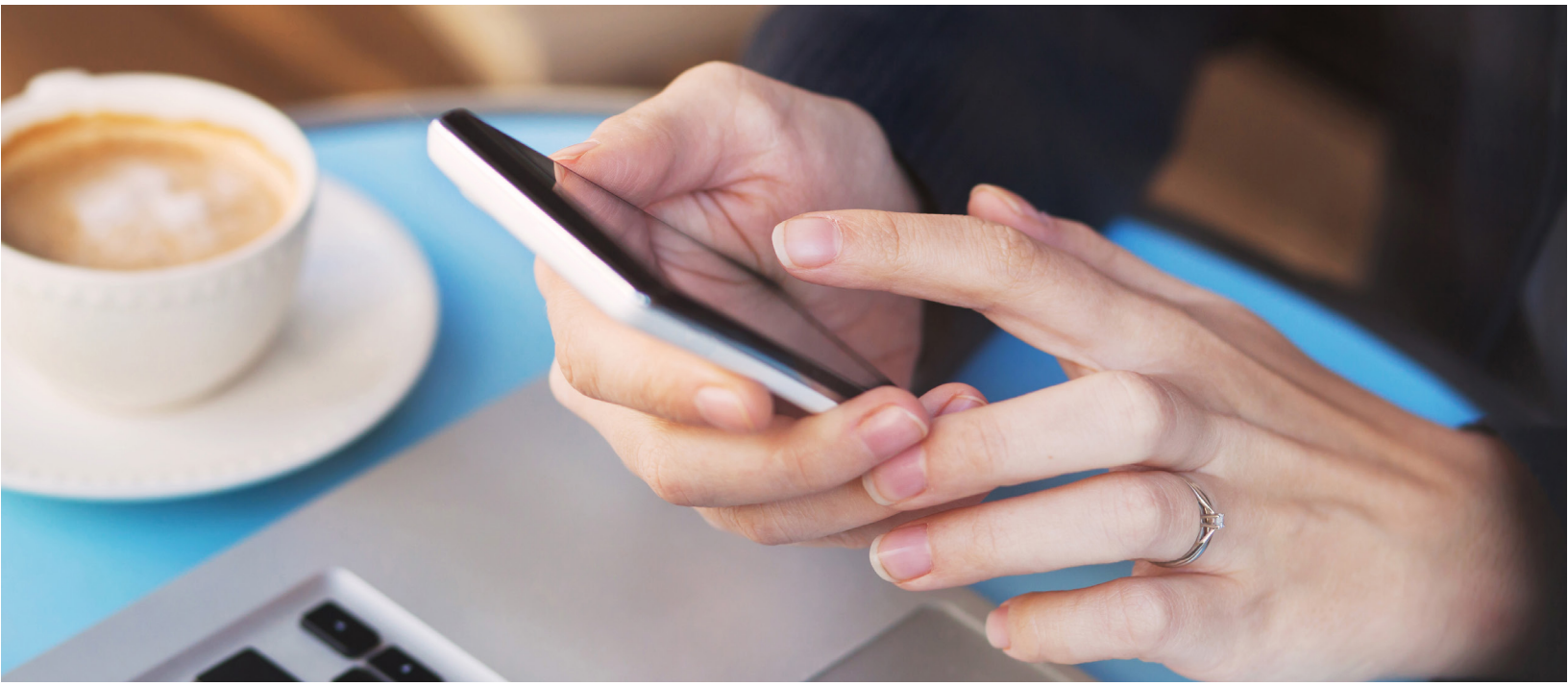
MULTI-FACTOR AUTHENTICATION: BEST PRACTICES FOR SECURING THE MODERN DIGITAL ENTERPRISE

Move from passwords or traditional 2FA to contextual MFA to optimize for cost, usability and security



WHITE PAPER

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Compromised credentials continue to be a top risk for breach with enterprises. It is not really that surprising when you consider the number of enterprises that still count on passwords as the single factor to authenticate users. Still others have moved to traditional two-factor authentication (2FA), but user experience and adoption can be poor with hard-tokens, not to mention very expensive. Enterprises must move to a more secure, more usable and more cost-effective model for authentication. This paper explores and recommends step-up MFA as a superior way to authenticate users.

Step-up multi-factor authentication (MFA) is a dynamic authentication model where the user—either a customer or an employee—is required to perform additional authentication operations, as needed, based on policy. Some typical examples of step-up MFA include:

- A customer, having signed on with a password to a banking site, wants to transfer money. The bank sends an SMS to the customer's previously registered phone number to establish the required additional assurance.
- An executive, trying to purchase a birthday gift for her child while traveling in Africa, is prompted to authenticate to her iPhone with her fingerprint to approve the transaction.
- A parent of a teenager receives a notification to approve a new channel that's been added to the family's cable TV package.
- A customer, signing on to an e-commerce site from her iPad at home, sees no visible authentication step until she has to change her settings.
- An employee is attempting to access a native SaaS application from the office. Because he's on the corporate network, he's not asked to perform any additional authentication.
- A customer wants to link a smart thermostat she purchased to her home automation platform account. She uses an application on her Android phone to provision credentials to the thermostat, and she's notified that the thermostat is being installed and asked to approve this sensitive operation.

This white paper proposes best practices for customer and enterprise deployments of step-up MFA. It explores a risk-based approach that combines dynamic step-up authentication with passive contextual mechanisms, such as geolocation and time of day. A risk-based approach provides a holistic assessment of users, their computing environment and the nature of the transaction they're attempting to perform, with the end goal of applying proportionate authentication and authorization.

Here are some of the advantages of a risk-based, step-up MFA approach:

- It creates an optimal user experience by demanding the minimum acceptable level of authentication for a given operation.
- If there is a cost-per-use of higher assurance mechanisms, risk-based models can be cost effective since more expensive options are used only when needed.
- It improves fraud detection relative to traditional binary rule sets.
- It creates a flexible and future-proof architecture than can adapt to emerging technologies and data assets.

In this white paper, you'll learn about:

- Authentication in depth, including its vocabulary, mechanisms and signals.
- Choosing the right MFA mechanisms for your environment.
- Applying a risk-based model to step-up MFA.
- Best practices in step-up MFA, including risk analysis, choice of authentication factors, privacy, lock-out, registration, user opt-in, suspension and bypass, self-service, native applications, initial authentication and multiple touch points/channels.

AN OVERVIEW OF AUTHENTICATION

Traditionally, authentication mechanisms have been categorized as either:

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).

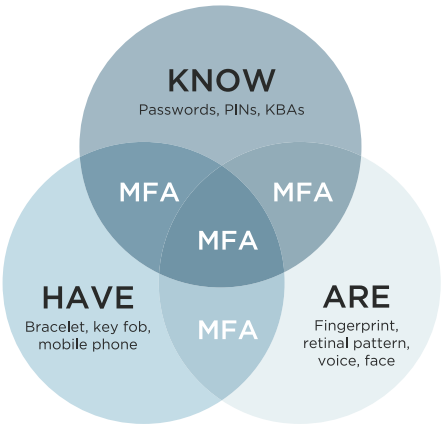


Figure 1: MFA requires users to identify themselves via two or more categories of authentication.

In theory, MFA goes beyond 2FA by requiring a user to authenticate via two or more authentication factors (e.g., a "something you know" combined with a "something you have"), as shown in Figure 1. In practice, however, there's still value in multiple factors of the same type, as long as compromising one factor doesn't mean compromising the other.

Generally, combining multiple authentication factors results in a higher Level of Assurance (LoA) 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 that's dedicated for authentication.

Before we explore authentication mechanisms in detail, let's first define the vocabulary we'll be using throughout this white paper.

AUTHENTICATION VOCABULARY

The world of authentication has many different (and sometimes contradictory) terms to describe authentication models. This white paper uses the following definitions:

- **Authentication** is the process of verifying that a claimed identity is genuine and based on valid credentials.
- A **credential** is something the user has access to (either "has" or "knows") that can be used in an authentication protocol. Before a credential can be used to authenticate the user, it must have previously been associated or bound to that user.
- **Identification** is the process by which information about a person is gathered and used to provide some level of assurance that the person is who they claim to be.
- **Identity proofing** is a part of the registration process that verifies a customer's identity before he/she is issued accounts and credentials.
- **A Level of Assurance or LoA** describes the degree of certainty that an individual is who he/she claims to be when presented with a digital credential. LoA is determined by the quality of the identity vetting, proofing and credentialing phase, and by the quality of the actual authentication process, including the quality/type of the authentication credential and robustness of the authentication mechanism. LoA models typically define about four different categories, each with defined requirements for identity proofing and the particulars of the authentication mechanism(s).
- **Multi-factor authentication or MFA** refers to the use of two or more credentials in the authentication of the user. Generally, the use of multiple factors/credentials results in a higher LoA about the user. Two-factor (2FA) is an example of MFA where two different credentials are used.
- **Registration** is the process by which the user is linked to his/her credential and identity record, and a corresponding credential is issued to the user.

AUTHENTICATION MECHANISMS

Now, let's explore different authentication mechanisms that might be part of a step-up MFA architecture. Some authentication mechanisms demand an explicit operation from the user, while others rely on passive collection (and so offer improved usability).

PASSWORDS

A password is a shared secret known by the user and presented to the server to authenticate the user. Passwords are the default authentication mechanism on the web today. However, poor usability and vulnerability to large-scale breaches and phishing attacks make passwords an unacceptable authentication mechanism in isolation.

HARDWARE TOKENS

These 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. 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.

A new form factor for hardware tokens has emerged. Small 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 sends it to the server as if the user had entered it by hand.

SOFT TOKENS

These are software-based security token applications, typically running on a smartphone, that generate an OTP for signing on. Software tokens have some significant advantages over hardware tokens. Users are less likely to forget their phones at home than lose a single-use hardware token. When they do lose a phone, users 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.

MOBILE AUTHENTICATION

Soft tokens leverage mobile phones' ability to generate an OTP and, possibly, their communication network. A user can demonstrate possession of his/her phone (previously bound to that user) by receiving a message sent to that device. An OTP can be sent to the phone by SMS (which is then entered by the user into a sign-on screen), or an app can receive a prompt for authentication via the mobile OS notification services, or the phone can be called. A mobile app provides useful context to the user to explain why he/she is being asked to authenticate and what he/she may be implicitly or explicitly authorizing. There's a big difference between "sign on to your bank account" and "empty your bank account."

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:

- It was never designed for security.
- It relies on operator practices around number porting, among other things.
- It doesn't protect against phishing, although it does force attackers to implement a real-time attack.
- It doesn't have the sort of delivery guarantee that authentication demands—a delay in delivery of minutes can effectively lock the customer out.

BIOMETRIC AUTHENTICATION

Biometric authentication methods include retina, iris, fingerprint and finger vein scans, facial and voice recognition, and hand or even earlobe geometry. Mobile devices can enable a preferred model for biometrics; the template can be stored on the device rather than the server. The latest phones are adding hardware support for biometrics, such as TouchID on the iPhone. Biometric factors may demand an explicit operation by the user (e.g., swiping a fingerprint) or may be implicit (e.g., analyzing the user's voice as they interact with the help desk).

The FIDO Alliance is defining a standardized architecture by which a user's local authentication to the device (e.g., laptop, phone) can be communicated to a server. When that local authentication is biometric (e.g., a scan of the user's fingerprint by a phone sensor or a facial scan), then the FIDO model's advantage is that the biometric template doesn't have to be stored on the server, with attendant privacy risk.

DEVICE IDENTIFICATION

Device identification is a process that establishes a device fingerprint somewhat unique to that device. Over time, this fingerprint allows the authentication server to recognize that device and determine when the user associated with it attempts to authenticate from a different device, which could indicate fraudulent activity. Device identification solutions create a fingerprint based on such characteristics as geolocation, OS version and browser. The simplest mechanism for device identification is a long-lived cookie that is set in the mobile browser by the authentication server. Device identification applications are best suited for organizations that have large populations of users accessing sensitive information from the Internet.

CONTEXTUAL AUTHENTICATION

This process uses contextual information, such as geolocation, IP address, time of day and device identifiers to determine whether a user's identity is authentic or not. Typically, a user's current context is compared to previously recorded context in order to spot inconsistencies and identify potential fraud. These checks are invisible to the authorized user so there are no usability issues, but they can create a significant barrier to an attacker. As an example, Visa's mobile location confirmation mechanisms determine the location of the user's mobile phone to verify that the user is physically near the location where the credit card is being used. The chances of a fraudulent transaction are higher if the transaction takes place in a different location from the phone. This is an example of using the context of the application channel, as compared to the context of a separate authentication channel, to spot potential fraud.

There are many other authentication mechanisms, including X.509-based certificates, that are more suitable to the enterprise space than customers given their deployment and implementation issues.

AUTHENTICATION SIGNALS

Contextual authentication presumes passively collecting (in theory) a variety of different signals about users and their context. These authentication signals might include their location (both physical and network), their computing environment and the resources they're trying to access.

Signals can be collected by:

- The web pages where they authenticate.
- The mobile devices used for MFA.
- Other network hardware.
- The application (or gateways in front).
- Other sensors in proximity to the user (e.g., wearables, smart watches, etc).

Once collected and aggregated, the risk and policy infrastructure can analyze these signals to look for anomalous patterns that might indicate an attack or fraudulent behavior. This analysis can be:

- **Contextual**, comparing a given signal value to a prescribed list of allowed or disallowed values (e.g., not allowing sign-on for any IP address coming from Uzbekistan).
- **Behavioral**, comparing a given signal value to the expected value based on a previously established pattern (e.g., an employee often travels to Uzbekistan on legitimate business, and therefore is allowed to sign on with MFA, whereas any other employee is prohibited from signing on from Uzbekistan).
- **Correlative**, comparing a given signal value to a different collected signal value and looking for inconsistencies between the two (e.g., according to the laptop IP, an employee is in the United States, but according to their mobile phone, this employee is in Canada).

CHOOSING THE RIGHT STEP-UP MFA MECHANISMS

How do you choose the right step-up MFA mechanism for your environment? Consider these variables when making your choice:

- **Strength:** How many false positives? How many false negatives?
- **IT benefits:** Is the authentication method easy to deploy? Will it require additional IT resources? Can it work across multiple channels (e.g., online, telephony, etc.)?
- **User benefits:** Is the authentication method easy to use? Will end users accept the new process? Can users be expected to have a device capable of supporting a particular mechanism? Does it put undue burden on users? Will users be concerned about privacy?
- **Industry-specific benefits:** Are there aspects of the authentication method that make it better suited for certain industries or functional areas? For example, if employees have to wear gloves to do their jobs, biometrics are not the best choice.
- **Initial purchase cost:** Is there a cost per user that will grow every time a new user is added? What is the replacement cost, both for the device and its associated administrative burden?
- **Deployment cost:** What are the costs associated with deploying the authentication mechanism? Is client hardware or software required? If so, how is that distributed to customers and what are the associated costs?

APPLYING A RISK-BASED MODEL TO STEP-UP MFA

The premise of risk-based step-up authentication 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 and, if necessary, to require the user be authenticated with an additional factor if the result of the calculation is below some threshold.

In this model, step-up authentication is typically triggered by atypical and anomalous context (e.g., signing on from Uzbekistan) or behavior (e.g., an attempt to complete a transaction over $100,000).

As shown in Figure 2 on the next page, to be granted access to some resource, a user authenticates with some factor—a password, for example. 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.

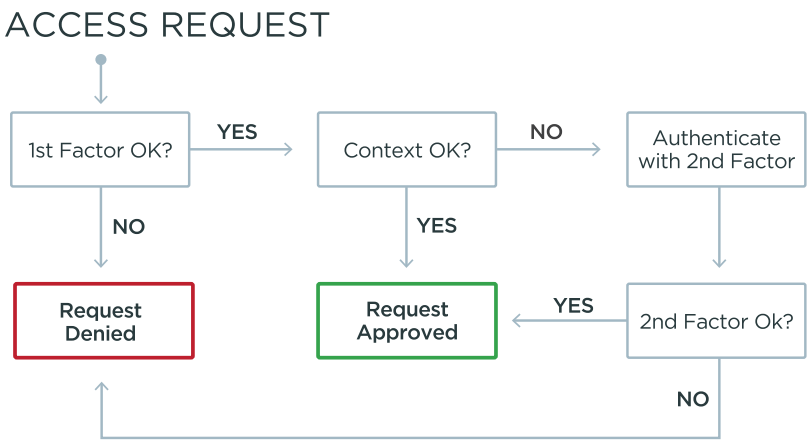


Figure 2: Risk-based step-up MFA is triggered by atypical and anomalous context or behavior. It's only when the context collected via the first authentication factor indicates something unexpected that a second factor of authentication is requested before access is granted.

A key advantage of risk-based step-up MFA is improved usability. A user is asked to authenticate with the additional factor only when necessary, as determined by the passive context checks and not by default.

Figure 3 shows screenshots of a banking mobile app using a form of step-up MFA. Some aspects of the app, like the branch phone numbers, are available even to an unauthenticated user. But when the user tries to access a more sensitive aspect, such as money transfer, they are prompted for authentication.

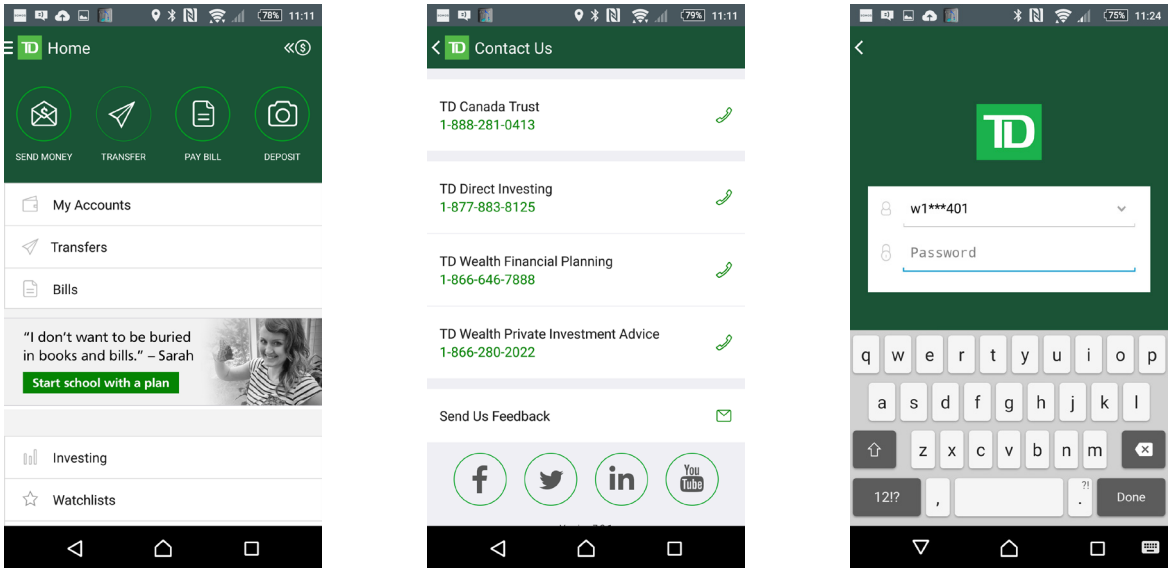


Figure 3: Only sensitive aspects of a banking app, such as a money transfer, prompt MFA using the risk-based step-up model

Keep in mind that one UK bank rejected the step-up model. Instead, this bank chose to authenticate the customer with the highest possible LoA mechanism available to that customer at the beginning of their session, regardless of the next app operation they might attempt to perform. The bank felt that asking for a step-up authentication later, only when warranted by a change in the risk profile, might confuse their customers and opted instead for a simpler model.

BEST PRACTICES FOR STEP-UP MFA

Ping Identity has compiled the following step-up MFA best practices from industry trends, interviews with customers who've deployed step-up, and our own experience and analysis.

RISK ANALYSIS

A risk-based authentication model presumes that the risk associated with different application resources and operations has been determined. The United States Office of Management and Budget (OMB) Memorandum M-04-04, "E-Authentication Guidance for Federal Agencies," defines a model for assessing risk that may be applicable in the consumer space:

The risk from an authentication error is a function of two factors:

1. Potential harm or impact
2. The likelihood of such harm or impact

Categories of harm and impact include:

- Inconvenience, distress, or damage to standing or reputation
- Financial loss or agency liability
- Harm to agency programs or public interests
- Unauthorized release of sensitive information
- Personal safety
- Civil or criminal violations

The risk assessment should be performed by the marketing, security and compliance teams collaborating on the level of risk they're willing to accept. Once resources have been categorized based on risk, the requisite LoA for each risk category can be decided. Authentication factors and models can then be chosen according to the level of LoA they can satisfy.

Let's look at an example of a risk analysis. A UK bank implemented a "what you can do when…" model. The bank applied this logic: "When authenticating with certain mechanism(s), a user can perform the following operations." The bank assigned strengths (ranging from 0–40) for the different authentication mechanisms they provided their customers. For instance:

- A physical card reader combined with a user PIN that generates an OTP was assigned a strength of 40.
- A mobile application that can generate OTPs was assigned a strength of 35.
- A password was assigned a strength of 20.

For each strength, there's a corresponding list of allowed operations (e.g., check balance, transfer funds, etc.) that can be performed once the customer has authenticated with that mechanism. Other methodologies for performing a risk analysis exist, but the fundamental requirement is mapping possible authentication mechanisms to different application resources.

CHOICE OF AUTHENTICATION FACTORS

The "one size fits all" approach doesn't work when choosing the appropriate authentication factors. A small user base that accesses highly sensitive resources may not require the same authentication factors as a large user base that accesses resources with less risk. Section 3 of this white paper, "Choosing the Right Step-Up MFA Mechanisms," may help guide your selection of appropriate authentication factors.

Organizations must balance usability, cost and security in order to enhance the user experience without alienating their user base. Different authentication factors can vary significantly in their user experience—from invasive to completely unobtrusive. A risk-based model ensures that the user is confronted with an explicit authentication UX only when necessary, with passive contextual authentication becoming the default.

A flexible MFA solution allows for easy switching between supported modes. For example, if a mobile phone is offline or if the user is roaming, the fallback is to a generated OTP. User adoption, particularly among customers, is enhanced if you provide multiple options for step-up mechanisms. Some users may not have phones for mobile-based mechanisms. Users with disabilities may rule out other mechanisms. And some users are simply resistant to new technologies.

Consider the number of different MFA mechanisms Google supports, as shown in Figure 4. These mechanisms account for different preferences and constraints of their users, plus serve as backup when the primary mode is unavailable.

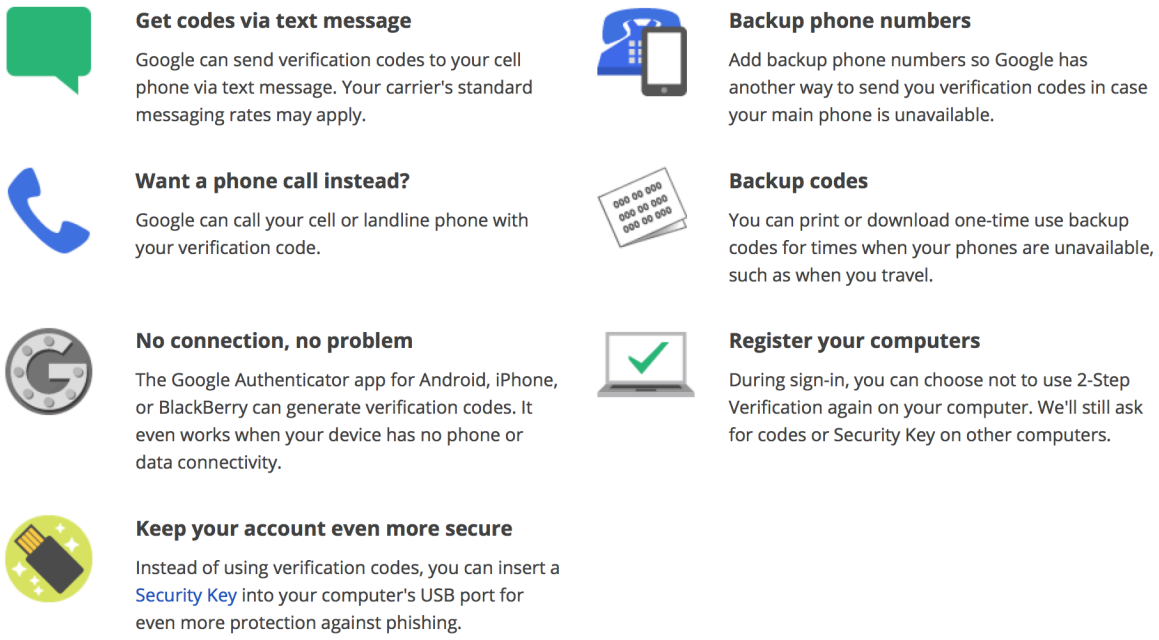


Figure 4: Google supports a wide variety of primary and backup MFA mechanisms to account for their users' preferences and constraints.

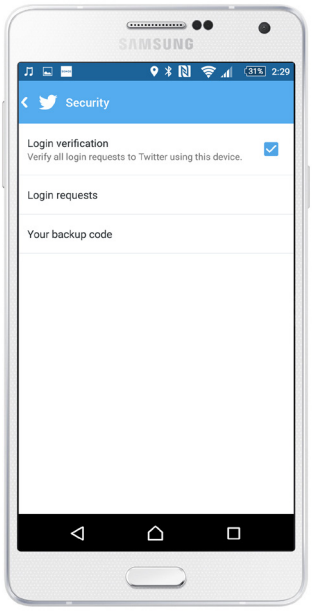


Figure 5: Twitter integrates MFA capabilities into its mobile app.

Hardware tokens may provide enhanced security relative to mobile-based soft token models, but the associated higher costs and poor usability make them less acceptable.

For a consumer mobile app, MFA capabilities should be integrated with any existing native application, rather than requiring the customer to download a separate authentication application. As shown in Figure 5, Twitter app users can turn on what's called "2-step verification" from within the app. Through the main Twitter application, users are prompted to approve certain operations—for example, signing on to their account from a previously unknown machine.

This presumes that a mobile app-based MFA solution offers an SDK, which allows its functionality to be integrated with existing business applications.

PRIVACY

Different authentication mechanisms demand different amounts of potentially sensitive user information to be collected. For instance, an SMS OTP model requires users to provide their phone number. If personally identifiable information is to be collected, its intended use should be explained to the user. Device identification solutions effectively create a fingerprint of a device. Used inappropriately, this fingerprint might allow a single customer to be tracked across multiple applications. Privacy laws in Europe restrict information that can be collected on users. Consequently, in some regions, companies are required to allow customers to opt out of device identification applications.

Protecting user privacy depends on providing opt-in MFA models and multiple options for the MFA mechanism. Giving users flexibility and control over their personal information is critical in the consumer space; it's also relevant at the enterprise level. Accustomed to heightened consumer privacy controls, employees no longer passively consent to outdated IT security policies.

LOCKOUT

Locking out a user from application access inevitably creates a negative impression. It may have a negative financial impact, such as when a customer can't complete a purchase. It may also decrease productivity; when an employee is locked out of their work applications, he/she can't perform job duties for some time.

Lockout should be a last resort. There are better options. For example, if a user repeatedly enters an incorrect password, judicious use of MFA can guide the user through a password reset process instead of locking the user out of the application altogether. Additionally, the continuous authentication model makes the need to consider lockout less likely. The more authentication signals collected and analyzed, the less critical an atypical value might be (that may warrant a lockout on its own) for any one of them.

REGISTRATION

An authentication mechanism is only as strong as the registration process that issued the credentials. A thorough registration process that strongly binds credentials to an individual user is mandatory.

For mobile app-based systems, displaying a QR code is a powerful and useful mechanism for the authentication server that's already authenticated the user with the first factor (e.g., password). Figure 6 shows an example of a QR code that appears in the PingID registration process. The user uses the previously downloaded and installed application (either purpose-built for authentication or integrated with an existing application) to scan the QR code. Because the QR code references the customer identity, the application becomes bound to that account.

Figure 6: The PingID mobile app uses a QR code during the device registration process.

USER OPT-IN

Different user constituencies may make it impractical to mandate step-up MFA for all users. Some users may not have a phone capable of supporting mobile app-based authentication. Some users may be apprehensive of and slow to adopt new technology. Step-up MFA should be marketed as a means to protect customer data rather a company's information assets.

Educating customers on the value of MFA is important. Creating opt-in incentives may be more effective. Some customers may be convinced to opt into MFA through offers of enhanced services. For instance, a U.S. bank increases the daily and weekly transfer limits for customers who sign up for its MFA service. Google offers a bonus to users who opt in to step-up MFA through a security checkup, as shown in Figure 7.

Figure 7: Google creates incentives for users to opt in to step-up MFA.

As for the enterprise, where MFA may be mandatory, allow for a graduated transition, where employees can choose to defer registering for MFA a few times or for a short period.

SUSPENSION AND BYPASS

Develop secure and thorough exception processes and backup access methods for common user situations such as forgotten, lost and stolen MFA credentials.

Consider allowing users to opt out of step-up MFA if they're accessing the application from a known and trusted device from which they've previously performed the MFA step successfully. Or you can apply a risk-based approach to a trusted device model and not require users to explicitly opt out.

Google uses this trusted device model to minimize overt step-up prompts, as shown in Figure 8.

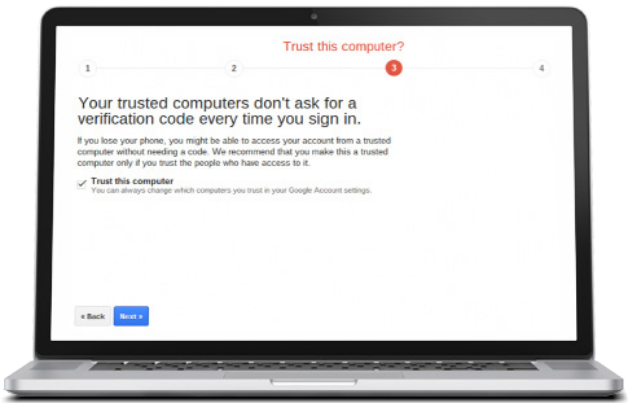


Figure 8: Google uses the trusted device model as a means of bypassing MFA.

For lost or stolen devices, implement a process that allows users to securely access the application or register a new device. Possible recovery mechanisms include:

- Sending a reset email to the registered address
- Knowledge-based answers (KBA)
- Printed recovery codes (which presumes users will safely store these and remember their location)

Non-secure recovery processes include asking questions that can easily be answered with a few Google or Facebook searches—for instance, where did you go to school?

SELF-SERVICE

Giving users the ability to manage their MFA mechanisms and devices themselves can provide important efficiencies. Self-service MFA mechanisms can be relevant at times of enrollment, recovery and revocation, and can minimize the need for costly administrative interventions. Self-service mechanisms can also give users more control over and visibility into their authentication information, which can enhance privacy overall.

NATIVE APPLICATIONS

The best practice for authenticating users of native applications is to use OAuth 2.0 or OpenID Connect 1.0. When the native app is first installed (or at some frequency after installation) the native application launches a browser window (not a web view) and loads the sign-on page at the corresponding server. After authentication, tokens are delivered back to the native app for subsequent use on calls to the relevant API.

At the time the token is issued, the provider may choose to enforce MFA. Based on the nature of the native application, policy may dictate that MFA is required even at the initial installation and setup. Because the authentication should happen in a browser window, and not within the application's UI, the same sign-on page (and step-up policy and architecture) put in place for the normal web application can be leveraged.

Through what are called "refresh tokens," OAuth allows for a long-lived session for native applications. This means that users may not be asked for an explicit sign-on for long periods of time. If long-lived refresh tokens are used, adding MFA to the sign-on process can help to more tightly bind that native application to the valid user, which can minimize the risk of the long-lived tokens.

It's also possible to implement a step-up MFA model for native applications. When the application presents a token on a request to a particular function, the API can assess the LoA associated with that token and, if insufficient, deny the request and indicate back to the application that a new token (with higher LoA) is required. The application passes this requirement back to the sign-on page, which triggers the appropriate step-up flow.

Additionally, the API can collect similar contextual information (e.g., IP address, time of day, app behavior) and feed that into the risk engine.

INITIAL AUTHENTICATION

While MFA mitigates some of their security issues, passwords are likely to remain the default first factor—the initial credential the user presents to the authentication server—for the foreseeable future. Current UX design typically prompts users to provide both on the same screen.

In the future, however, it may not be a password that the authentication server asks for as the initial credential. Instead, context or some other mechanism may be sufficient, which would negate the need for a password. It may make sense for organizations to consider a UX design that anticipates this future. Google is doing just that. As shown in Figure 9, Google separated the request for username and password from its sign-on page in May 2015.

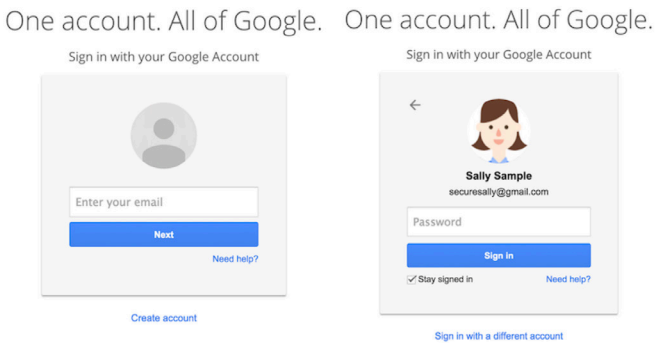


Figure 9: In anticipation of a future where a password may not be the initial credential that an authentication server requests, Google separated the username and password request from its sign-on page.

Distinguishing between how users identify themselves to Google (as shown on the left of Figure 9) and how they authenticate (as shown on the right) creates greater flexibility for future authentication schemes. When Google rolls out novel authentication methods beyond passwords, the first identification page won't need to change. Additionally, authentication logic sitting behind the first page can choose the appropriate authentication method for a given combination of user and context.

MULTIPLE TOUCH POINTS/CHANNELS

If a company offers multiple channels by which its customers can interact with application resources and data—for example, web, mobile apps, telephony and brick-and-mortar locations—the MFA choice may be different for different channels.

For example, a U.S. bank uses passive voice biometric authentication for its telephony channel. Similarly, Canadian bank Manulife recently announced an update to its interactive voice response (IVR) system with the deployment of natural language understanding (NLU) and passive voice biometric technologies.

CONCLUSION

It is clear that enterprises must continue to evolve and improve their authentication of users, moving beyond the limitations of passwords and traditional 2FA. MFA is better, but using contextual data to dynamically step-up authentication is the right approach for enterprises.

Today, contextual authentication is seen as complementary to active and explicit authentication factors. But in the future, Ping Identity expects contextual authentication to become the norm and explicit authentication to be used less frequently. A risk-based authentication architecture combines step-up MFA with passive contextual authentication for the optimal combination of cost-effectiveness, usability and security.

Ping Identity recommends these five high-level best practices for step-up MFA:

- Step-up MFA should be supplemented with passive contextual authentication.
- A risk-based approach, based on operation requests and contextual indicators, should be used to determine when to request step-up MFA.
- Step-up MFA should be optional for the majority of customer scenarios. Customers should be encouraged to opt into step-up MFA by educating them on its advantages and by creating optin incentives. Risks of customers opting out of step-up MFA should be mitigated through other mechanisms.
- Employees should be given a choice of and options for their authentication mechanisms. If employees opt out of enhanced active modes of authentication, companies should supplement their practices with passive modes.
- A variety of MFA options should be supported to address the needs of different user constituencies—for example, users with disabilities or users who are less tech savvy.

To learn more, visit www.pingidentity.com.

#3001 | 09.09 | v00b