WHITEPAPER

Verification beyond SMS

OVERVIEW

In response to remote work, organizations are looking to enhance existing measures — or roll out additional ones — to fight fraud and protect user accounts. Today, SMS is the most commonly used channel to add an extra layer of security. And for good reason. Adoption and implementation are easy, and it's useful for protecting the primary areas where fraud happens: account signups, logins, transactions, and password resets. But how do you know if SMS is right for your business? Choosing the right verification method comes with trade-offs between user experience, cost, deployment time, as well as security. How you prioritize these will depend on your organization and how much friction your users are willing to accept in order to protect their accounts. In this white paper, we examine the pros and cons of verifying users via SMS and when it's worth considering user verification beyond SMS.

↘ SMS as an Authentication Factor

We're all familiar with SMS — if you have ever received a text message, you know how easy a form of communication it is. In fact, 96% of Americans can receive SMS messages and that's why, for years now, SMS has also been considered a great option as a second form of identity verification — commonly known as SMS OTP (one-time passcode).

For example — log into your banking app on a new device, receive a text sent to confirm your identity. Or, log into Facebook from a new location — receive a text to verify it was really you attempting to log in.

Easy, familiar, and perceived to be "secure enough," SMS OTP has extended to the workplace for companies that have implemented two-factor authentication (2FA) — log into your work email, Slack or other apps, and just provide an SMS OTP to complete the login.

The second factor is useless if users don't enable it, and SMS has the distinct advantage of easy user onboarding: no additional app install required. This is why many companies continue to offer SMS as an option: SMS 2FA is still better than no 2FA at all.

While SMS OTP may be the quickest way to get up and running with 2FA, there are more secure channels.

Common issues with using SMS OTP as a 2FA factor

1. SIM Swapping/SIM Hacking

The SIM card in your phone essentially tells your phone which wireless carrier to connect to, and what phone number to connect with. In a SIM swap/SIM hack attack, a threat actor can use social engineering or bribery to convince a carrier that they are, in fact, you.

Ultimately, your phone number is then assigned to a new SIM card on a different phone. In a SIM swap/SIM hack, threat actors do not need access to any of your physical devices to gain access to your accounts. Once your number has been switched to a device in their possession, they can receive all SMS OTP messages tied to your online accounts.

Who should be concerned: companies with high profile or high value accounts like celebrities on Twitter or individuals with large cryptocurrency balances on a banking app.

2. Synced devices

Nowadays, phone numbers are connected to personal information, such as banking and wealth management apps. A lost phone is a gateway to someone's finances.

In general, 2FA is considered a combination of two pieces of evidence which prove that you are who you say you are — a knowledge factor (something you know), an Inherence factor (something you are), or a possession factor (something you have). Using password and an SMS OTP as a factor is a combination of knowledge and possession factors.

Because SMS messages aren't tied to a single device and can be synced across multiple devices, the "possession" factor can have many access points, which means more access points for an attacker. This is considered insecure when text messages are synced with users' computers, leaving the door open to access if people log in on a shared device.

3. Taking over your phone account portal

Some wireless providers allow users to view text messages via your online account, within their web portal. If your account for the web portal itself isn't protected with a strong password or second factor, a threat actor can monitor a user's account for an SMS OTP message that they initiated for a particular app and consequently, gives the threat actor access to those accounts.

4. Social engineering & phishing

Unfortunately, SMS OTP is not the only form of authentication susceptible to social engineering phishing attacks. Less secure factors like passwords and security questions are equally susceptible. In a social engineering attack, a threat actor posing as an employee from a trusted brand convinces a customer to hand over their account credentials, and in many cases, the SMS OTP sent to their device as well.

For example, a customer receives a call from their "bank" telling them that they need immediate access to their account for security purposes. In this case, the customer unknowingly gives a threat actor their username/password combination, as well as the SMS OTP code, which gets sent to their phone during the login process.

Phishing attacks aren't just specific to calls or emails. Users can receive a phishing text message with a link to an imposter site, and if they inadvertently type a username/password combination into the malicious website, the threat actor could then use this information to take over the real account.

This is not a full list of the issues with using SMS OTP as a factor, but should give a sense of why it's wise to consider using stronger factors to protect your users and their data.

What can be used as an alternative to SMS?

While SMS-based verification may still be the right option for some businesses, you can still offer multiple factor types to validate a user's identity. Ideally, you can enable multiple factor types for a single user, with some factors as required and some as optional. Here are some recommendations on more secure factors to enable for your users:

1. Mobile authenticator apps (TOTP)

Mobile authenticator apps like Authy, Duo, or Google Authenticator support OTPs within the app. When a user enters their credentials into a web app, they are then prompted to enter the generated TOTP. This authentication method is even stronger if the mobile authenticator app supports biometrics, like FaceID on iOS or fingerprint on Android.

Benefits of mobile authenticator apps over SMS OTP:

- Does not rely on your wireless carrier's reliability or security — the TOTP is tied to your phone, regardless of the phone number.
- Based on an open standard for authentication that can be used for both enterprise and consumer use cases and gives customers flexibility for which authenticator app they wish to use.
- TOTP codes expire quickly, often rotating every 30 seconds, to offer a higher level of security than SMS OTP.
- Location agnostic and available offline. The inputs to the TOTP algorithm include a secret key (synced at device registration) and the system time — both available offline — so neither cellular nor internet connectivity is required.

There are many different mobile authenticator apps on the market; some are a better fit for enterprise use cases than others.

2. Verification SDKs

One potential drawback of using authenticator apps is that companies cannot control the user experience during authentication. Asking users to leave the app they are using to toggle to the authenticator app, or responding to a push verification request from the authenticator app, means users cannot get a seamless in-app experience. This is where verification software development kits (SDKs) are indispensable.

Using proven security technologies, such as public-key cryptography, verification SDKs allow companies to embed the security and functionality of an authenticator app within their own mobile apps. These SDKs integrate with a verification API backend, allowing the app, SDK, and API to work together to turn the user's mobile device into a unique digital key that the user can login with.

Benefits of Verification SDKs

- A standards-based approach to secure passwordless authentication.
- Works for both web and mobile apps.
- Verifications over a secure data channel with encryption in-transit and at-rest.
- SILENT Push flow that is completely seamless to the user. Even if the user needs to be prompted, it's just one touch to Approve.
- Phishing-resistant factor type via a public and private key pair for each device that a user enrolls with.
- Step-by-step visibility on the status of the push, and if the user tapped Deny, that's a useful fraud signal that OTPs don't provide.
- Predictable costs globally, since there are no telco costs to deal with.
- Granular control over exactly which device to send the push.

- Better user privacy because no user PII is required, like a phone number or email address.
- Compliant with Europe's PSD2 SCA regulation's dynamic linking requirement for approving transactions.

3. FIDO2.0 (WebAuthn)

WebAuthn is a browser-based API that allows web applications to simplify and secure user authentication by using registered devices (phones, laptops, etc.) as factors. It uses public key cryptography to protect users from advanced phishing attacks.

As of 2019, the World Wide Web consortium announced WebAuthn as the new web standard for passwordless logins. Like Push Verifications, WebAuthn is also phishing-resistant.

WebAuthn factors can be on-device (platform), or off-device (roaming). Here are some details on both:

Off-device/roaming authenticators: These are WebAuthn-supported factors that are not built into the hardware (computer/phone). Examples include:

- Yubikey 5Ci
- Feitian BioPass
- HID Crescendo smart card

On-device authenticators/platform authenticators: These are WebAuthn-supported factors that are built into the hardware (computer/phone). Examples include:

- Windows Hello on Windows 10 1903 and later
- Touch ID on MacBook
- Fingerprint on Android 7.0+

Support for WebAuthn is dependent on developers updating the web app authentication process to support the WebAuthn API, browser support, OS support, and hardware support. This may seem overwhelming, but thankfully, many operating systems, devices and browsers already support WebAuthn. And, while consumer apps are still in the process of adopting this standard, if you're using an enterprise-grade authentication provider to secure access for the workforce, it's likely you'll be able to use WebAuthn with that provider.

Benefits of WebAuthn:

- A standards-based approach to secure passwordless authentication
- Phishing-resistant factor type via a public and private key pair for each WebAuthn on- or off-device authenticator that a user enrolls with
- Less friction for end users — use of biometrics means quick logins
- The same biometric you use to login/unlock the device can be used to access apps
- Multiple options for devices & security keys

Examples of browsers, hardware, and operating systems that support WebAuthn:

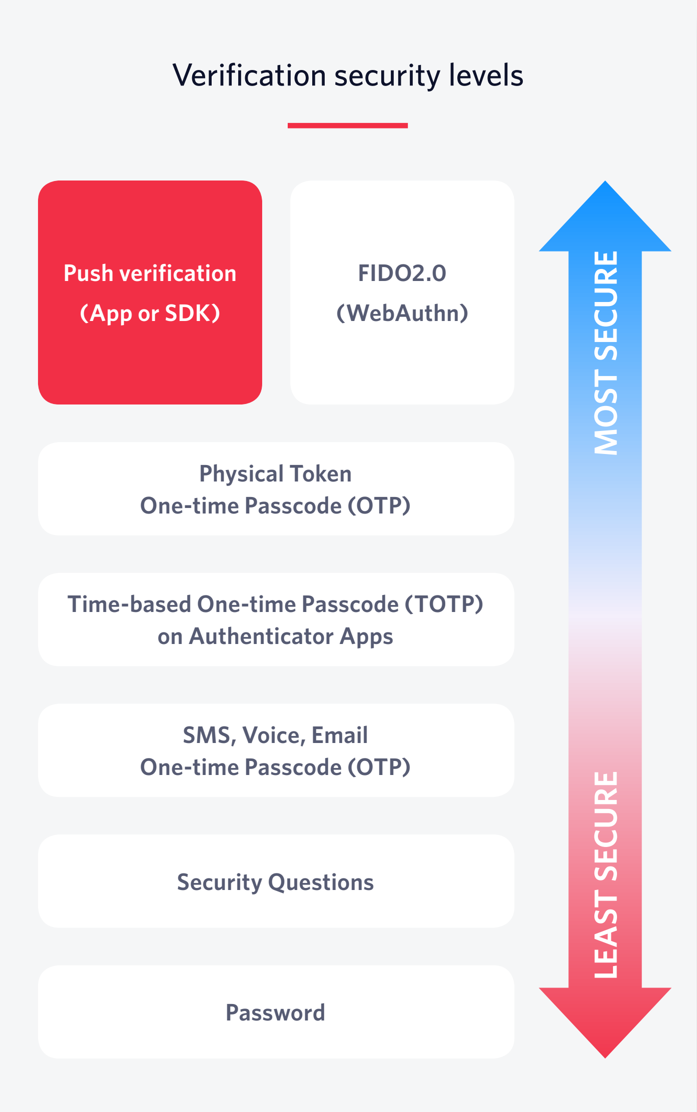
- Google Chrome on MacOS using Touch ID
- Google Chrome on Windows 10 using Windows Hello
- Microsoft Edge on Windows 10 using Windows Hello
- Firefox on Windows 10 using Windows Hello
- Google Chrome on Android 7.0+ using devices with fingerprint support
- Desktop apps on Windows and MacOS that use a WebAuthn compatible browser for login using Windows Hello and Touch ID, respectively
- Native mobile apps that use a WebAuthn compatible browser (ie Chrome) for login on Android 7.0+ using fingerprint support

The list above shows that while phones and computers are gradually becoming compliant with WebAuthn, there are still many users with non-compliant devices for whom WebAuthn is not a viable verification approach.

For example, a B2C company cannot ship physical authenticators to all its consumers, nor can it expect them to have compliant on-device authenticators. For this reason WebAuthn is better suited to enterprise use cases, where companies can influence the end user's device and ensure its compliance with WebAuthn.

While it's not an exhaustive list of available factors, the diagram below shows a few examples of common factors and their security levels. As you'd expect, using just a password has the lowest level of security, and the highest likelihood of account takeover and subsequent data breach.

Keep in mind that less secure methods like SMS OTPs also have less onboarding friction for the end user, which means that more people might enable the factor. However if many of your users already have your mobile app installed, in-app Push verification with an SDK might be just as easy and more secure.


Verification security levels

Push verification (App or SDK)
FIDO2.0 (WebAuthn)
MOST SECURE
Physical Token One-time Passcode (OTP)
Time-based One-time Passcode (TOTP) on Authenticator Apps
SMS, Voice, Email One-time Passcode (OTP)
Security Questions
Password
LEAST SECURE

Conclusion

Securing accounts with SMS OTP is better than nothing or using just a password. But the great news is that you have other options.

We recommend enabling more secure factors like embedded push, mobile app authenticators, and WebAuthn as optional factors for your users (at a minimum) — this gives them the flexibility in using more secure factors if available, and still allows for SMS OTP as backup. There are many great options for securing accounts, both in the enterprise and for consumers.

Twilio has been supporting organizations with user verification solutions for many years, working with customers like Stripe, Washington Mutual Business Services, Twitch, Deliveroo and other companies to properly secure customer accounts and financial transactions. Our Verify API abstracts away the complexities of building, scaling, and maintaining a global, multi-channel user verification solution that our customers can implement quickly and easily without ongoing security and maintenance costs.

Watch a demo (https://www.twilio.com/go/twilio-ittb-verify-push-download-1) of how Twilio Verify Push SDK balances cost, UX, and security.