2FA SECURITY ISSUES

Static passwords have had their heyday—a different approach is needed when it comes to improving user security

2FA Security Issue

- What is 2FA
- Why is 2FA Important?
- What are the types of 2FA?
- What Threats Does 2FA Address?
- Vulnerabilities in 2FA
- Risks that 2FA aim to Mitigate
- Better alternatives to 2FA SMS

What is 2FA

Two-factor authentication (2FA) is a specific type of multi-factor authentication (MFA) that strengthens access security by requiring two methods (also referred to as authentication factors) to verify your identity. These factors can include something you know—like a username and password—plus something you have—like a smartphone app—to approve authentication requests.2FA protects against phishing, social engineering and password brute-force attacks and secures your logins from attackers exploiting weak or stolen credentials.

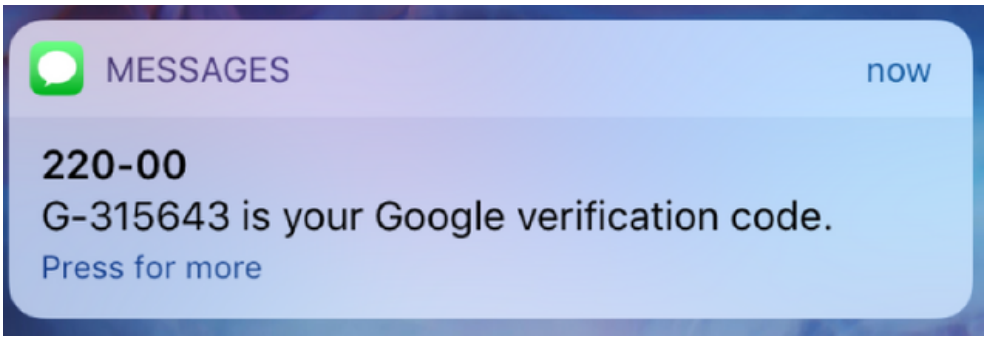
Why is 2FA Important?

Two-factor authentication (2FA) is the foundational element of a zero trust security model. In order to protect sensitive data, you must verify that the users trying to access that data are who they say they are. 2FA is an effective way to protect against many security threats that target user passwords and accounts, such as phishing, brute-force attacks, credential exploitation and more.

Let's say you use a username and password to complete primary authentication to an application. That information is sent over the Internet (your primary network). You'll want to use a different (out-of-band) channel to complete your second factor. Approving a push notification sent over your mobile network is an example of out-of-band authentication.

What are the types of 2FA?

SMS 2FA



SMS two-factor authentication validates the identity of a user by texting a security code to their mobile device. The user then enters the code into the website or application to which they're authenticating.

U2F Tokens

U2F tokens secure two-factor authentication by using a physical USB port to validate the location and identity of a user attempting to login. To use a U2F token, a user inserts the token into their device and presses the button located on the top of the device. Once the token is activated, the user enters their PIN and gains access to their accounts.

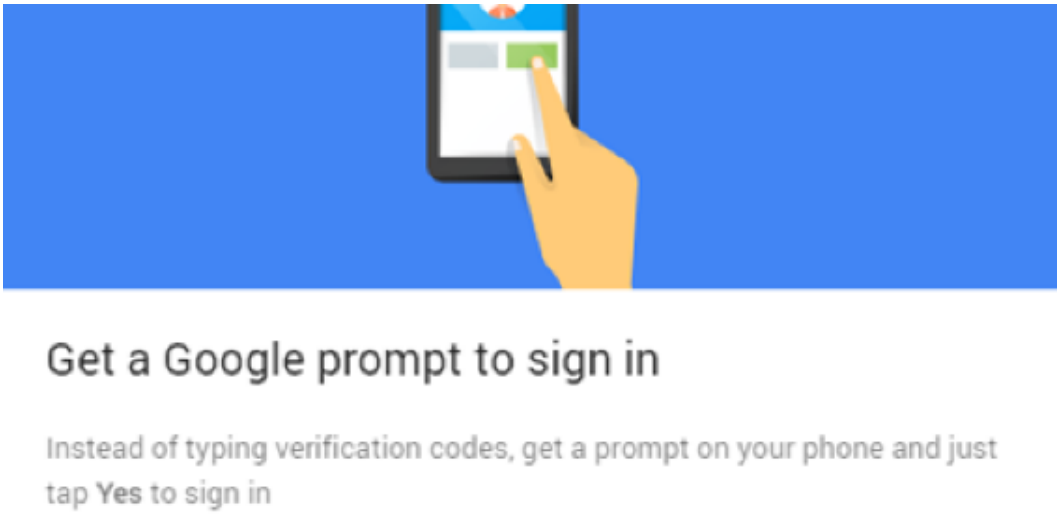
What are the types of 2FA?

TOTP 2FA

The Time-Based One Time Password (TOTP) 2FA method generates a key locally on the device a user is attempting to access. The security key is generally a QR code that the user scans with their mobile device to generate a series of numbers. The user then enters those numbers into the website or application to gain access. The passcodes generated by authenticators expire after a certain period of time, and a new one will be generated the next time a user logs in to an account. TOTP is part of the Open Authentication (OAUTH) security architecture.

What are the types of 2FA?

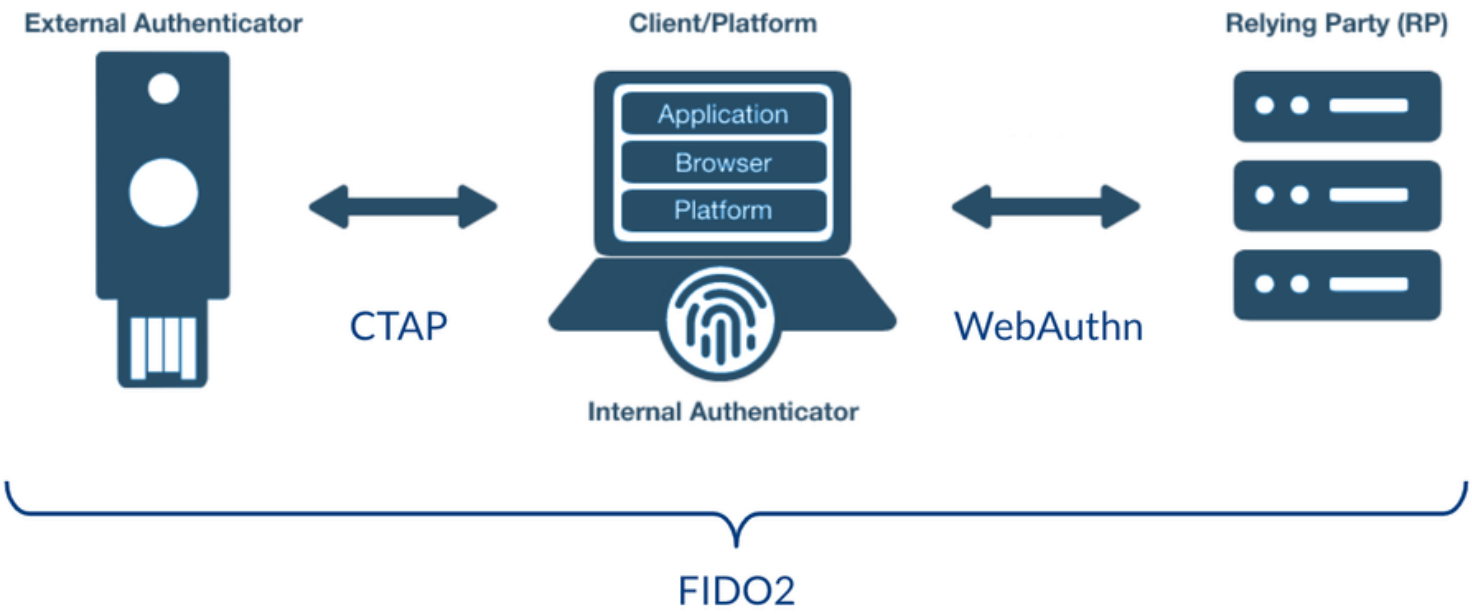
Push-Based 2FA



Push-based 2FA improves on SMS and TOTP 2FA by adding additional layers of security, while improving ease of use for end users. Push-based 2FA confirms a user's identity with multiple factors of authentication that other methods cannot. Duo Security is the leading provider of push-based 2FA.

What are the types of 2FA?

WebAuthn



Created by the FIDO (Fast IDentity Online) Alliance and W3C, the Web Authentication API is a specification that enables strong, public key cryptography registration and authentication. WebAuthn (Web Authentication API) allows third parties like Duo to tap into built-in biometric authenticators on laptops and smartphones, letting users authenticate quickly and with the tools they already have at their fingertips.

What Threats Does 2FA Address?

- **Stolen Passwords**
- **Phishing Attempts**
- **Social Engineering**
- **Brute-Force Attacks**
- **Broken Logic**
- **Key Logging**

Vulnerabilities in Two-Factor authentication

Bypassing two-factor authentication

At times, the implementation of two-factor authentication is flawed to the point where it can be bypassed entirely.

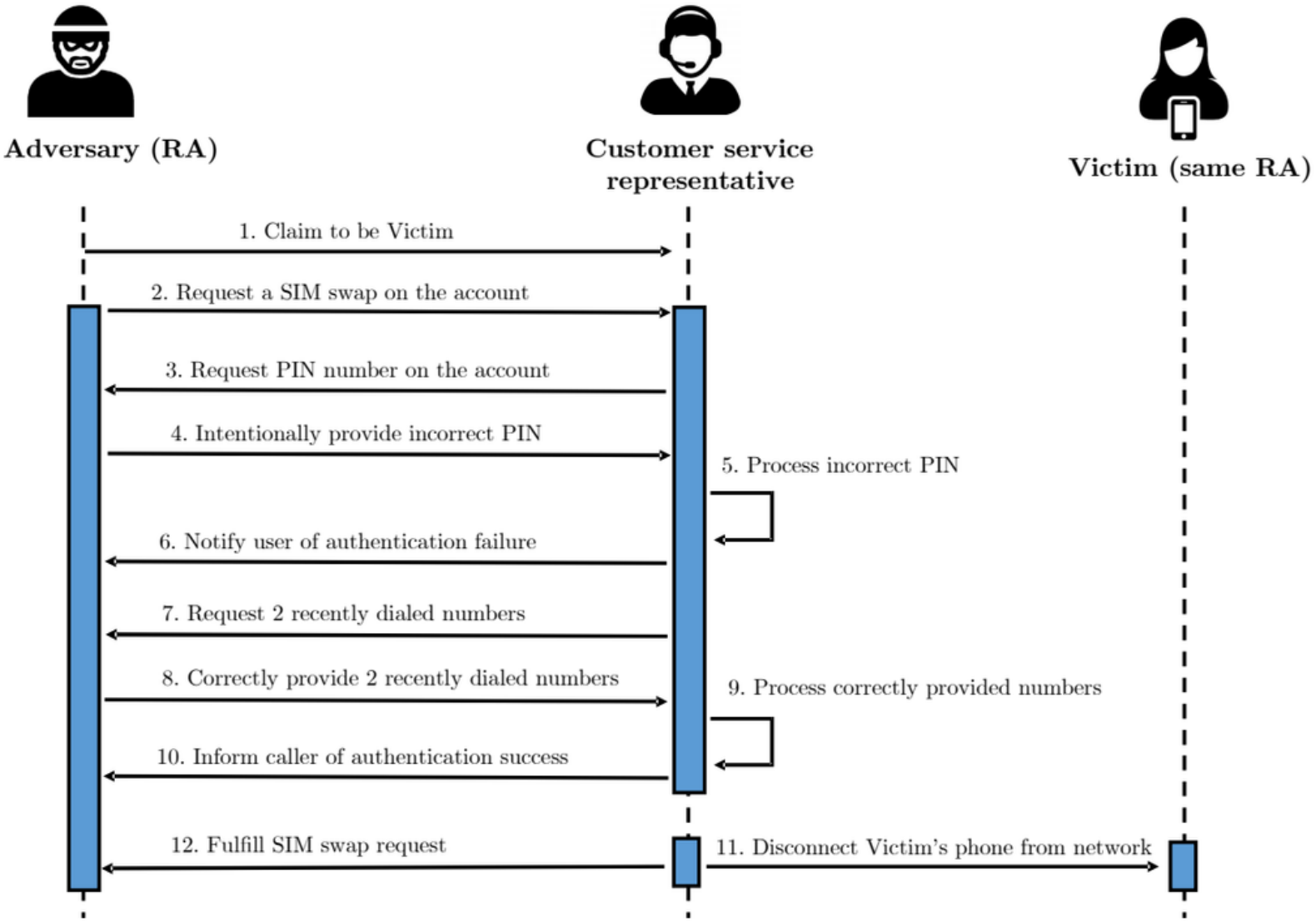
If the user is first prompted to enter a password, and then prompted to enter a verification code on a separate page, the user is effectively in a “logged in” state before they have entered the verification code. In this case, it is worth testing to see if you can directly skip to “logged-in only” pages after completing the first authentication step. Occasionally, you will find that a website doesn’t actually check whether or not you completed the second step before loading the page.

Two-factor authentication tokens

Verification codes are usually read by the user from a physical device of some kind. Many high-security websites now provide users with a dedicated device for this purpose, such as the RSA token or keypad device that you might use to access your online banking or work laptop. In addition to being purpose-built for security, these dedicated devices also have the advantage of generating the verification code directly. It is also common for websites to use a dedicated mobile app, such as Google Authenticator, for the same reason.

On the other hand, some websites send verification codes to a user's mobile phone as a text message. While this is technically still verifying the factor of "something you have", it is open to abuse. Firstly, the code is being transmitted via SMS rather than being generated by the device itself. This creates the potential for the code to be intercepted. There is also a risk of SIM swapping, whereby an attacker fraudulently obtains a SIM card with the victim's phone number. The attacker would then receive all SMS messages sent to the victim, including the one containing their verification code.

SIM Swapping Attack



Brute-forcing 2FA verification codes

As with passwords, websites need to take steps to prevent brute-forcing of the 2FA verification code. This is especially important because the code is often a simple 4 or 6-digit number. Without adequate brute-force protection, cracking such a code is trivial.

Some websites attempt to prevent this by automatically logging a user out if they enter a certain number of incorrect verification codes. This is ineffective in practice because an advanced attacker can even automate this multi-step process by creating macros for Burp Intruder. The Turbo Intruder extension can also be used for this purpose.

https://github.com/PortSwigger/turbo-intruder

https://portswigger.net/burp/documentation/desktop/options/sessions#macros

2FA broken logic

Sometimes flawed logic in two-factor authentication means that after a user has completed the initial login step, the website doesn't adequately verify that the same user is completing the second step.

1. POST /login request, the verify parameter is used to determine which user's account is being accessed.
2. Log out of your account.
3. GET /login and change the value of the verify parameter to victim and send the request.
4. Go to the login page and enter your username and password. Then, submit an invalid 2FA code.
5. POST /login
6. set the verify parameter to victim and add a payload position to the mfa-code parameter. Brute-force the verification code.

Risks that 2FA aim to Mitigate

- Implementing 2FA is one of the simplest and most effective actions that a company can take to improve the security of the deployment to ensure only authorized users can access the secure data.
- Two-factor authentication, used in conjunction with a username, can prevent unauthorized access and credential leakage by ensuring that only a user, who can be validated against a second authentication factor, will be authorized to access the online resource
- It incorporates logical and physical security which helps in filling the gaps in both security domains and reduces risk. In addition, it can reduce brand, reputation, and customer relationship damage resulting from identity theft fraud.
- The benefits encompass bringing improved security, productivity, and flexibility in the workplace, fraud reduction, and having secure online relationships.
- Two out of three [attacks] focus on credentials at some point in the attack. Trying to get valid credentials is part of many styles of attacks and patterns.
- Two out of three [attacks] focus on credentials at some point in the attack. Trying to get valid credentials is part of many styles of attacks and patterns.
- brute force and dictionary attacks, in which perpetrators use automated software to generate massive amounts of username/password combinations in an attempt to guess a user's credentials.
- social engineering attacks, e.g., phishing and spear-phishing, which attempt to dupe a user into revealing sensitive data, including their username and password

Better alternatives to 2FA SMS

While it's best to skip 2FA if SMS is the only option, this doesn't solve the reason for adding 2FA in the first place. To prevent brute force and other attacks targeting password-only authentication, some form of 2FA is needed.

Hardware authentication

Hardware authentication relies on a dedicated physical device to grant access. Along with their password, users will also have to input a random token code generated by the device. Logins will fail without the code. Providers of hardware authentication include RSA SecurID and Thales SafeNet.

Software authentication

Software authentication is essentially the same principle as hardware authentication. But instead of requiring a physical device, token codes are generated with a mobile application. The most popular authentication app is Google Authenticator, but there are many options. For example, RSA now offers their SecurID authenticator as an app.

IP-based authentication

This method checks the user's IP address when logging in. You can block access to specific IP addresses suspected to be malicious, or simply only allow logins from known IP addresses and ranges. IP-based authentication can be used in conjunction with other forms to add another layer of protection.

Resources

- https://blog.sucuri.net/2020/01/why-2fa-sms-is-a-bad-idea.html
- https://duo.com/product/multi-factor-authentication-mfa/two-factor-authentication-2fa
- https://portswigger.net/web-security/authentication
- https://www.sathwikat.me/describe-the-risks-that-two-factor-authentication-mechanisms-aim-to-mitigate/
- https://devsecopsguides.com