Preventing Account Takeovers on Centralized Cryptocurrency Exchanges

Recommended Practices

February 5, 2025

Prepared by: **Shaun Mirani, Kelly Kaoudis, and Evan Sultanik**

About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 100+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence with regard to blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Ethereum 2.0, MakerDAO, Matic, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

Trail of Bits, Inc.
228 Park Ave S #80688
New York, NY 10003
https://www.trailofbits.com
info@trailofbits.com

Notices and Remarks

Copyright and Distribution

© 2024 by Trail of Bits, Inc.

All rights reserved. Trail of Bits hereby asserts its right to be identified as the creator of this document in the United Kingdom.

This document is considered by Trail of Bits to be public information. Material within this document may not be reproduced or distributed in part or in whole without the express written permission of Trail of Bits.

The sole canonical source for Trail of Bits publications is the Trail of Bits Publications page. Documents accessed through any source other than that page may have been modified and should not be considered authentic.

Reference to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply endorsement, recommendation, or favoring by Trail of Bits. Where illustrative, we use certain businesses and services to show examples of particular technologies only. Trail of Bits has not partnered with any businesses, services, or apps mentioned in this document in order to produce it.

Table of Contents

Overview

Defending a complex system like a centralized cryptocurrency exchange (CEX) platform against a catastrophic outcome like user account takeover (ATO) requires the CEX to appropriately implement and maintain many intertwined risk mitigations. This document is directed toward stakeholders who might be involved in or affected by ATO, enumerating the types of vulnerabilities that an attacker can leverage to compromise platform user accounts. A security vulnerability is any trust assumption involving people, processes, or technology that can be violated to exploit a system[1]. Therefore, we not only propose software-level mitigations but also recommend supporting these mitigations with policy and user-facing guidance practices that reduce ATO risk.

ATO mitigations can be difficult to correctly implement in a user-friendly way. For example, multi-factor authentication (MFA), sometimes called two-factor or two-step authentication, is a common security control that helps platforms mitigate the risks of attacks such as password cracking or credential stuffing. But, as expanded on later in this document, MFA has technical and process components that are equally critical to get right. Successfully implementing MFA also requires clear and appropriate user documentation so legitimate users can appropriately choose, store, and use their second login factor. User MFA should not be bypassable by an external attacker exploiting, for example, a platform's password reset flow or through exploiting an alternative login means such as a social login. Insiders should also be unable to bypass user MFA by abusing account recovery flows.

While not comprehensive, this white paper is intended to be a thorough reference. Since it is written as a reference, it does not need to be read sequentially cover-to-cover. We provide executives with a high-level overview of the vulnerabilities and entities involved in user account takeover. We describe recommended security controls that help mitigate ATO that they can bring to lead engineers and technical product managers to check for and prioritize if not yet implemented. Software engineers, security engineers, and other technical employees should understand the risks of not prioritizing or appropriately integrating, maintaining, and documenting each overlapping security control that helps mitigate ATO.

[1] NIST SP 800-154 Section 2.1.1, lines 223-226, "Vulnerability"

Account Takeover: the Basics

When an attacker takes unauthorized ownership of a given account on a particular platform or service, we term this account takeover (ATO).

Common Attack Vectors

ATO is an end condition that results from attack vectors that mainly fall under the 2023 OWASP API Security Top 10 category of Broken Authentication (which is closely related to the 2021 Top 10 categories of Identification and Authentication Failures and also Broken Access Control). In this section, we'll provide a high-level overview of some common attack vectors that can be employed to achieve account takeover. Some of these vectors may need to be chained (combined) to yield ATO. For example, ATO on a given platform may not only require leveraging credentials from a data breach, but might also involve SIM swapping in order to bypass the enabled multi factor authentication (MFA) on the account in question.

Password compromise

When MFA is not supported or enabled, a user's password is the only piece of knowledge regulating access to their account. Their login ID, whether it is a username or email address, is often predictable.

If a user's password is weak and can be determined by brute-forcing the character space, by enumerating a dictionary of common passwords, or by guessing passwords that have personal meaning to the user, it is relatively straightforward to compromise their account in a targeted attack. Such targeted attacks are of more concern to VIP or high-value account holders, but even typical users can get caught up in widespread password compromises.

If the user has reused the same password for an account on another platform, and that second platform suffers a data breach in which account passwords are exposed, an attacker with access to the contents of the breach may be able to determine the password and compromise the user's account on the first platform. When performed against many accounts en masse, this is called credential stuffing. Credential stuffing is a common account takeover vector since it can be done cheaply, in bulk, and with little effort.

SIM swapping

An account takeover vector especially of concern for users with high-value accounts (and other VIPs) is SIM swapping, where an attacker gains the ability to intercept text messages and phone calls sent to the target user.

A SIM swap attack starts with the attacker conducting reconnaissance or social engineering of the user to find out enough of their personal information to impersonate them. The attacker then contacts the user's mobile carrier, pretending to be the user. The attacker requests to move the user's phone number to a SIM card the attacker controls, using the personal details gathered to convince the carrier that the request is legitimate. If the attacker succeeds, they will then have access to the user's SMS messages and phone calls.

This vector is especially problematic for platforms that rely on text message/phone call verification (e.g., for MFA and account recovery flows) and do not attempt to detect SIM swaps. Due to the effort required to gather information about a victim and impersonate them, SIM swaps are rarer than other vectors and are likely to be targeted, often to VIP or high-value accounts.

Phishing

Phishing (under which we group vishing, smishing, etc.) is a notoriously difficult to mitigate class of attacks that often includes sender identity spoofing. Even certain forms of MFA, such as mobile authenticator apps and SMS verification codes, are susceptible to phishing. Phishing is relatively cheap to execute and can target many victim users simultaneously, for example, via malicious email or text message campaigns. Phishing messages do not always have to include bad grammar or implausible scenarios like a foreign prince who wants to split a sudden windfall with the message recipient.

For example, the attacker could trick a user into clicking a link that silently executes a malicious script or downloads a malicious payload, or into entering their credentials into a fraudulent website made to look like the real one; or they might send a text message claiming to be the user's work supervisor or romantic partner in order to request a sensitive or risky action like a funds transfer. Phishing attempts may even involve realistic-seeming LLM-fabricated content. For targets with better security who would yield a greater reward (e.g., the CEO of a corporation), the attacker may conduct targeted reconnaissance in order to more deeply personalize the phishing attempt so that it appears more legitimate.

Clientside malware

Suppose the attacker does manage to trick the victim user into clicking that malicious link. Rather than attacking the platform directly, such an attacker can instead infect the user's device with malware that could, for example, exfiltrate live session cookies or tokens. This vector is rare but is still potentially of concern if the user is a VIP, if the user's device is running out-of-date software, or if the user can be convinced to install software from untrustworthy sources.

Password reset flow bypass

Password reset is a necessary component of most platform accounts. Different flavors of password reset are more susceptible to bypass through use of external data breach dumps and similar credentials datasets. It is also possible for attackers to discover registered email addresses, phone numbers, or other data that can be used to reset platform accounts by abusing a password reset flow that lacks appropriate rate limiting and/or provides clues about the validity of the entered input in error messages.

Externally communicated identity verification bypass

Most platforms provide a password reset feature that anyone, including an external attacker, can initiate using a known username, phone number, or email address. The platform first verifies that the initiator is the account owner, often by sending a message with a one-time link or code to one or more of the communication methods associated with the account. If the initiator clicks the right link or enters the right code, they are allowed to set a new password for the account. Such functionality can often be easily exploited when an attacker has already achieved compromise of the target user's email account and/or phone number (e.g., via SIM swapping), since the attacker has control over the communication method to which the platform sends identity verification links or codes.

Security question bypass

An older, less secure method of verifying password reset requests that many platforms continue to use is having the initiator provide account-specific, previously shared answers to one or more security questions. Such questions often ask for concrete personal details that are hard to forget, like the user's mother's maiden name, the make of the user's first car, or the name of the user's first pet. If the user supplied the platform with factual, predictable answers to these questions, an attacker may be able to guess the answers via a dictionary attack or may be able to determine them through reconnaissance. This flow is easier to exploit than one based on email or SMS verification, which enables attacking the accounts of many users at once if, for example, a list of registered email addresses or usernames is leaked in a data breach.

CAPTCHA bypass

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a class of user liveness checks (e.g., rotate the donkey in the correct direction; select all images in the set that contain buckets; type in the numbers in the shown image in the right order) that platforms may leverage to help mitigate spammy actions like automated bot farming or credential stuffing. Actual humans are not always skilled at passing CAPTCHA checks on the first try, and some types of CAPTCHAs even violate accessibility guidelines. CAPTCHA bypass automation from scratch may require some programming skill, but it is also a common requirement of web scraping and legitimate automated web application QA, so a number of publicly available toolkits and services provide it.

Account recovery flow bypass

Many platforms support account recovery when an unintentional account lockout happens. This means if a user has lost access to the email or second factor associated with their account, they can regain access to their account by initiating a process outside the normal login flow. In contrast to automated password reset flows, these processes can depend on manual support staff work. If human beings are part of the flow for assessing the situation and validating the user's identity, an external attacker could use deception (social engineering) to gain unauthorized access to a user account. However, exploiting a manual account recovery flow could require significant attacker time and effort and would be usually directed at high-value or VIP users.

Application vulnerabilities

Application-specific vulnerability types that we also find relevant to ATO are briefly touched on below for completeness.

Authentication / authorization compromise

Bugs naturally afflict all software. By exploiting application-level vulnerabilities in authentication logic or in access controls (e.g., insufficient cookie or bearer token validation; lack of authentication attempt rate limiting; lack of appropriate session revocation logic), an attacker may be able to perform account takeover.

Vendor (supply chain) compromise

Attackers may be able to compromise a third-party vendor that a platform uses for security-critical actions, such as delivering verification codes to users and employees or for single sign-on (SSO). Platforms should carefully select endorsed third-party MFA vendors or SSO integrations and prioritize them by their level of vendor security support and general popularity. For example, an SSO vendor with a good reputation for prompt breach disclosure and quick security patch application should be prioritized over other potentially cheaper or seemingly desirable integration options.

Actor Categories

In this section, we'll establish types of actors who could directly or inadvertently threaten a given cryptocurrency platform and its users via actions leading to account takeover. For example, in a confused deputy attack such as cross-site request forgery, a normal user who is induced by a third party to take a malicious action against the system (such as initiating a password reset) would be both the victim and the direct attacker. Establishing the types of actors that could threaten the system is useful in determining which protections, if any, are necessary to mitigate or remediate vulnerabilities. We will refer to these high-level actor types in the rest of this document.

Users

While anonymous usage of other types of cryptocurrency applications and platforms is possible, centralized exchanges like Kraken, Coinbase, Gemini, and Binance are legally required to obtain identifying information during account signup from their users as part of know-your-customer (KYC) processes intended to help prevent fraud (including ATO) and money laundering. Though some application security and risk considerations in this document might also apply to decentralized exchanges and their potentially anonymous users, we will primarily discuss centralized exchanges with known users here.

- A typical **platform user** can log in and perform actions on the platform using any approved platform client, such as changing their account settings, transferring currency into the account, or making trades via the platform.

    - To use the centralized exchange platform, the user divulges their identifying information to the platform.

- **High-risk accounts** have additional characteristics that make attackers especially likely to target them. The platform may enact extra precautions for or provide white-glove service to high-risk users who may fall into subcategories like the following:
    - A high-value account user custodies an amount of currency with the platform over a certain platform-defined threshold.
    - A VIP account belongs to a person or organization of note who attackers may especially target.

Defenders

“Platform” and “CEX” refer to a given centralized cryptocurrency exchange platform and similar services.

- **Platform employees** can access the platform’s internal network and may also have access to potentially sensitive user data and identifying information.
    - The security team implements and maintains platform infrastructure and data defenses.
    - Support personnel handle user inquiries such as account recovery cases.
    - Engineers on-call for particular applications, teams, or areas of responsibility for the platform handle unexpected alerts and issues as they arise.
- Third-party vendor employees implement and maintain defenses for that vendor’s technology and may periodically release security patches and other updates.

Attackers

- A **remote attacker** is positioned on the public internet.
    - The remote attacker can send traffic to and receive traffic from the platform.
    - A remote attacker targeting a particular user may gain access to the user’s SMS via targeted social engineering of the SMS vendor (leading to SIM swapping).
    - A remote attacker targeting a particular user may gain access to the user’s email via phishing, use of email provider password reset flows, or other means.

    - A remote attacker that gains access to a vendor’s service offering incorporated in the platform, such as SMS delivery, may compromise it, which could result in the eventual compromise of platform users.

- An **internal attacker** is positioned on the platform’s internal network.
    - An internal attacker may intercept and manipulate internal platform traffic.
    - A well-meaning but naive platform employee (insider) may take incorrect action on behalf of the user, such as turning off MFA while working on a manual account recovery case.
    - A compromised or malicious platform employee (insider) may take unauthorized actions such as performing malicious trades “on behalf of” the user.

Cryptocurrency Platform Recommended Practices

Cryptocurrency platforms and other services within the cryptocurrency sphere can implement a wide array of defenses to protect their users from account takeover. Some of the defenses we discuss here are specific to CEXes, while others could also apply more broadly to other types of online platforms and services. This section outlines these recommended practices.

Password Handling

- Never store or cache plaintext passwords.

- Store passwords only in hashed and salted form using a password hash function, listed here in order of most to least preferred: Argon2, scrypt, bcrypt, PBKDF2. Refer to OWASP for guidance on selecting parameters such as an appropriate number of rounds to use for these functions.

- Consider using a "pepper," a salt-like value that is not stored alongside hashes. The goal of pepper use is to add another complication beyond appropriate salting to make password cracking more difficult if an attacker is able to exfiltrate the authentication database.

- Enforce a minimum password strength using a component like the zxcvbn library.

- Use the Have I Been Pwned API to detect previously breached passwords when users sign up or change their password.

Multi-Factor Authentication

- Require users to configure at least one authentication factor other than their username and password. For example, Gemini requires MFA via Authy for all accounts.

- When the user requests a password change, deposit/withdrawal account change, or other sensitive account settings change, require re-authentication with both the user's password and a second factor.

- The most secure MFA method is the use of a U2F hardware token security key like a Yubikey. Such security keys are resistant to phishing and SIM swapping. Encourage users to choose this method over others, explaining these security benefits. See NIST's requirements for conveying the risks of so-called "restricted authenticators", such as SMS-based MFA and TOTP-based MFA, to end users.
  - While out of scope for this document, note that the use of a hardware token inserted into a port on a machine that has been previously compromised with malware (like a keylogger) cannot defend the user against that malware.

- Allow and encourage the user to configure multiple second authentication factors so that they have backup second factors in the case, for example, they lose access to a device on which an authenticator app is configured.
  - Encourage the use of backup factors on different devices from the primary second factor or that are distinct hardware tokens.
- Provide each client a set of backup codes in case they lose access to all other second authentication factors. Specify how users should save and store backup codes and any "seed phrases" or seed QR codes that the user may additionally elect to store at their own risk.
  - As noted in the two-factor authentication section, clients should store these in a password manager or print them out on physical paper, then delete the digital copy (and empty the operating system's trash folder).
- Allow and encourage users to disable or opt out of insecure MFA methods such as SMS in favor of more secure MFA methods. Explain that SMS-based MFA is vulnerable to phishing and SIM swapping (for example, see Gemini's documentation).
- Require clients storing over a certain amount of currency with the cryptocurrency custody service, regardless of whether they are registered as or otherwise considered by the platform to be "VIP clients," "private clients," or "private wealth clients," to:
  - Use hardware tokens as part of two-step login authentication;
  - Use distinct second factors (e.g., different hardware tokens) for account login versus transacting;
  - Not use face- or thumbprint-based login + second-factor replacements;
  - Allow push notifications for potential fraud detection notifications;
  - Allow push notifications whenever a new device, contact method, or authorized signer is added to the account;
  - Use only devices authorized with the service to interact with the account;
  - Use multi-signature (*m* > 1 of some N total number of authorized parties must sign the transaction within a certain time limit) for all transactions to take place;
    - Each authorized party should use a hardware token as their second authentication factor.

        - Multi-signature should be the default setting for transacting from high-value accounts unless the account holder opts *out* and knowingly accepts the liability of being the single signer for transactions.

Fraudulent Activity Detection

Fraudulent behavior may be observably different from normal user activity. Actions that require authentication or authorization generally cross a trust boundary within the system, so should be considered potentially risky. Such actions should always be audit-logged following the platform's internal data sensitivity and retention guidelines.

Check multiple potential fraud indications periodically as potentially risky actions occur for an account, not just during initial authentication/login. Particular indicators and metrics associated with the account in question can be automatically checked to score and rank each potentially risky user action. Such signals could be fed into a data aggregation, inference, and alerting system that may also be integrated into the platform's risk-based authentication strategy and into the platform's SIEM to inform security engineers.

As a toy example, suppose a user sends funds to a destination address that they very recently added to their account's address book. This should be considered more risky than, say, a regularly scheduled transfer of funds to a destination address associated with the account for some time. If the transfer also happens close in time to a failed user login attempt before login actually succeeded before the transfer, it should be considered still more suspicious.

Potential Signals

We break down some commonly available user identity aspects into three categories: location, pattern of expected behavior, and device indicators. Platforms can monitor indicators such as the following to help detect actions that could lead to ATO and other unauthorized activity:

User location

- If the user has ever connected from the originating IP address before.
- Whether the connecting IP address' ISP, AS, time zone, and geolocation correlate with known user data.
- Whether the IP is denylisted or associated with, for example: a proxy service, a VPS/colocation service, or a Tor exit node.
- If *other* platform accounts have also attempted login or similar actions from the same IP/location with similar HTTP headers and other metadata within a certain amount of time, this could indicate an attacker is attempting credential stuffing.

- If a fraud analytics service can be queried for similar attempts across other platforms.

- If activity is indicative of "impossible travel," such as logins from two different countries within an unreasonably short timeframe.

Pattern of expected behavior

- How well this request's timing correlates with the timing and frequency of past requests of the same category (login, transfer, password change, etc.) from the user.

- Use of an atypical or backup second factor to complete an MFA login, especially if the used second factor in question is SMS or email.

- Frequency of failed access attempts (is this one of several?)

- Frequency, size, destination, and timing of outbound transactions or transaction attempts (is this an unusually large or unusually small transaction for this account? Has this account made transactions over a long period of time to the destination in question?)

- Use of an in-app communication flow on a different registered device, or even an email/SMS/phone call to a communication method *not* involved in the action in question, to request that the account holder confirm that they originated the behavior.

- Whether the user's emergency contact(s) also recognize or expect the behavior.

Known device indicators

- Several services allow platforms to detect whether a phone number has had a recent SIM swap. For example, see *How to detect a SIM Swap before sending an SMS OTP* from the Twilio blog.

- Whether application-specific user settings recorded in cookies, browser local storage, or elsewhere on the user device match those seen in concurrent or prior sessions.

- HTTP header indicators such as user-agent string, header ordering, allowable compression types, and others may also provide insight into whether the device type, browser, and operating system match expected or reasonable values.

- Browser fingerprint, which incorporates not only the user-agent and possibly some application-specific data but also other data that can be used to identify users and compare behavior across sessions.

- If there have been recent session management actions such as the termination of other logged-in sessions, which may indicate an attacker trying to retain access within a session that they control.

Data Handling and Storage

Fraudulent activity detection and risk score calculation often require retaining sensitive or personal (PII) user data for some time and potentially imply feeding it into machine learning-based systems. Such a system needs not only data to operate on but also needs to be compliant with applicable regulation. However, regulatory compliance is fully out of the scope of this work; the reader should consult qualified legal counsel to answer any related questions.

We highlight the following generally applicable points:

Data should be deletable

- All user data, sensitive/personal/PII or otherwise, must be deletable when and if the authenticated and authorized user requests that deletion.

- Data must be fully deleted: when it is no longer needed, or when it falls outside its regulation-defined retention period, or when the user requests deletion.

- When data is deleted, any associated or derived metadata should also be deleted.

- Deletion should be verifiable across the entire platform.

- When data is deleted, it also needs to be removed from any machine learning models (inclusive of foundation models/LLMs) that have been trained on that data.

Classify datasets by what they contain

- Use a high-level internal data classification system for all datasets and data stores.

- The following bullets comprise an example data sensitivity classification scheme. These categories range from least to most sensitive:

    - Public-appropriate or **not sensitive** (appropriate for most use cases without approval)

    - Contains **anonymized metadata** derived from users' sensitive or personal data, or from proprietary internal data, or from proprietary external data (internal eyes only, not appropriate for public release, new use cases require review)

    - Contains **proprietary or copyrighted internal data** (deny access by default, authorized need-to-know internal eyes only, new use cases require review)
    - Contains **proprietary or copyrighted *external* data** belonging to others (deny access by default, authorized need-to-know internal eyes only, new use cases require review)
    - Contains **financially sensitive internal data** (deny access by default, authorized need-to-know internal eyes only, new use cases require review)
    - Contains **identifying, personal, or sensitive user or employee information** (deny access by default, authorized need-to-know internal eyes only, new use cases require the most stringent review)

Data use cases must be documented and easily searchable

- The data classification system must be accompanied by an approval process that leverages a defined list of acceptable use case examples for datasets with non-public classifications.
    - This means, for example, a dataset such as users' phone numbers approved for one use case *cannot* be leveraged for any other use case without the use case specifically receiving documented approval.
- This data classification system should also be accompanied by clearly recorded restrictions on how data sets—especially those containing any of the common types of sensitive data we defined above—can be stored, accessed, and transferred, not just on who is allowed to have access and for which use cases.
- Build on use of the internal dataset classification system to create a searchable, fine-grained data inventory that describes specifically what data are stored where, what roles/groups/applications can access them, what classification(s) each data item in a given dataset has, and how long each data item can be retained under the relevant regulation.

Risk-Based Authentication

Risk-based authentication (RBA) is a widely used scoring system designed to estimate the probability that a login attempt is malicious based on criteria like those we covered in the Fraudulent Activity Detection section, such as whether the IP address of the request matches a recently used one, and whether the device being used has been seen before or is new. The sensitivity of the platform and the requester's attempted action should also be considered. The use of such signals and scores should be clearly recorded so that during incident response, it is possible to understand how a given score was derived.

The higher the score, the riskier the attempt should be considered, and the more verification the platform should require before considering the user authenticated and/or authorized and before allowing simultaneous logins from other registered devices. This verification must be independent of and in addition to any MFA verification already enforced for the account.

Examples of verification checks that could be individually required or potentially even combined for the user to complete a risky login attempt include:

- Clicking a one-time link sent to the verified email address associated with the account.
- Using a one-time code emailed or texted to a different communication method associated with the account.
- Secure an emergency-only backup key in the security hardware (TPM, Apple Secure Enclave) of a registered device. Then, when a risky login occurs, retrieve the key and use it to sign a message to provide to the server.
- Answering "yes, this was me" to a confirmation push notification on another device that is already logged in.

Reauthentication with know your customer (KYC) information

KYC is mandatory for US financial institutions including CEXes in order to comply with government regulations and reduce risk of fraud. KYC can be leveraged not just during user signup, but as well for risky login verification, especially for VIP risky login. MFA and metrics-based verification might be insufficient to permit a user to complete any login or other action that meets a certain risk score threshold, especially if the account in question is a VIP's or has a high balance. An account takeover attempt may imply the victim's email and/or phone number are already under the attacker's control, but KYC features could also potentially be leveraged to help with identity verification to complete a risky login without the use of such potentially compromised communication methods.

For user login attempts that receive the highest risk scores, completing the current login action and regaining the ability to log in normally should require the requester to confirm their identity. This could be accomplished by providing official documentation such as a government-issued ID for verification, such as during Apple account recovery, or even by providing a selfie to compare to a picture of the user's ID, potentially via a KYC API such as those provided by ID.me, Veriff, or Jumio.

Platforms should:

- Implement and leverage risk-based authentication.
- Use the indicators listed in Detection of Fraudulent Activity to calculate the risk score.

- Risk should be proportional to the account's value. Once an account reaches a certain value threshold, certain protections, such as requirements for additional authentication and verification factors, should be turned on by default (made opt-out rather than opt-in).

- Not limit fraudulent activity detection to the time of initial authentication, as noted in the previous section. Indicators of risk should be checked for every sensitive operation, such as transfers of funds, password changes, changes to authentication methods, etc.

- Require the user to pass a KYC-style identity check, such as those provided by services like Veriff or Jumio (see the related Account Recovery suggestions below) to complete a given login attempt over a chosen maximum risk threshold, especially for VIP users or users with high-value accounts.

User Notifications

By default, all activities that trigger fraud detection or RBA escalation should also immediately notify the user via in-app push notification or message instead of an external communication method like email or SMS, since the user's email or SMS might have been compromised to enable the account takeover in the first place.

Suspicious activity alerts should be displayed conspicuously the next time the user logs into the website. For high-risk accounts, alerts could also be sent to user-assigned trusted third parties (e.g., a spouse or business associate). Each notification should include, at minimum, the time, location, and device information of the suspicious action. Consider locking the account and/or postponing the transaction until the user verifies its authenticity from at least two additional authentication factors or communication methods associated with the account.

Potentially undesirable activity reported to the user via push notification should, at minimum, include:

- Adding a new authorized device (i.e., the device can log in and/or receive push notifications) to the account

- Adding a new authorized transaction signer to the account

- Login attempt or successful login with unexpected client request metadata (e.g., location, browser, device ID, operating system)

- Changing or adding new account security settings

- Changing or adding new account contact methods

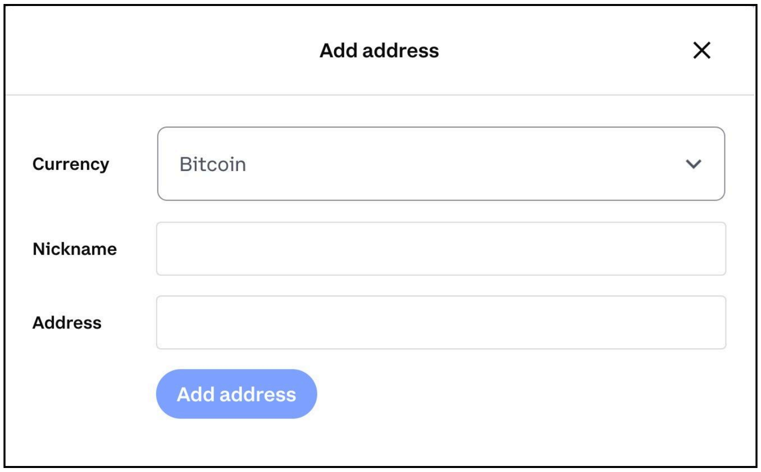
- Changing or adding new user-assigned trusted third-party contacts

- Allowlisting a new address for withdrawals
- Requesting an account unlock

Address Allowlists

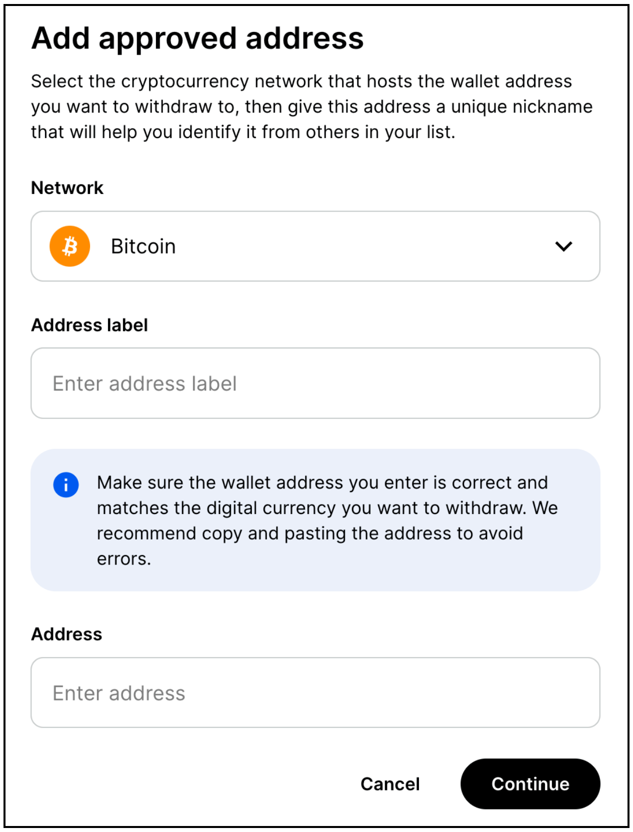
Custodians should require extra verification as described in Risk-Based Authentication on the first use of a new destination address, and should support withdrawal address allowlisting. When withdrawal address allowlisting is turned on, the only permitted funds withdrawal destination addresses should be those included in the user's withdrawal allowlist. Adding an address to the allowlist should trigger a waiting period before funds can be withdrawn (see Waiting Periods). The allowlist should be enabled by default for sufficiently high-value accounts.

We include UI examples below from Coinbase's address book allowlist (Figure 1), Gemini's approved addresses (Figure 2), and Kraken's withdrawal addresses (Figure 3).



Currency: Bitcoin

Figure 1: Coinbase's address book allowlist



Add approved address

Select the cryptocurrency network that hosts the wallet address you want to withdraw to, then give this address a unique nickname that will help you identify it from others in your list.

Network

Bitcoin

Address label

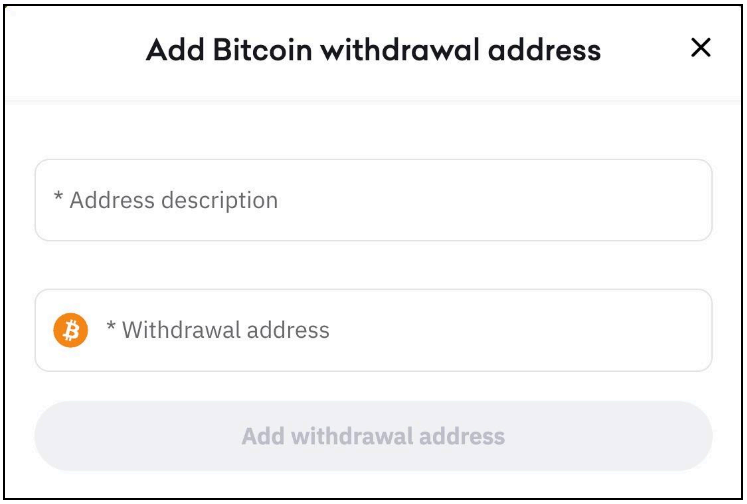
Enter address label

Make sure the wallet address you enter is correct and matches the digital currency you want to withdraw. We recommend copy and pasting the address to avoid errors.

Address

Enter address

Figure 2: Gemini's approved addresses feature



Add Bitcoin withdrawal address

* Address description

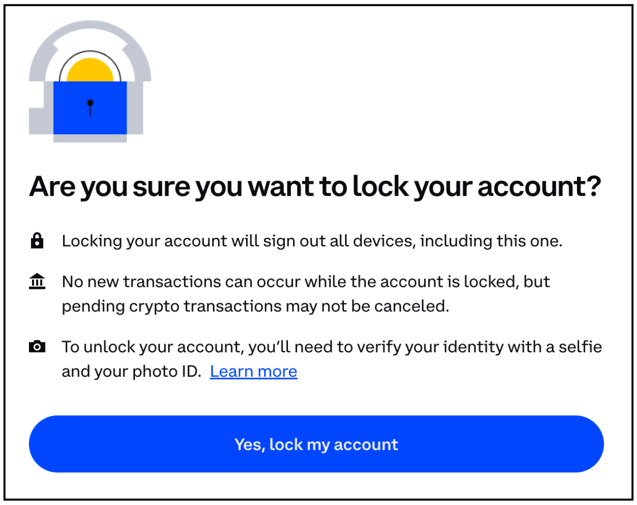
* Withdrawal address

Figure 3: Kraken's withdrawal address feature

Account Locking

Exchanges should allow users to lock their accounts as a last defense against account takeover. The design of this feature assumes that an attacker has compromised the user's password and MFA method and can successfully log in as them. To mitigate this, the account lock prevents changes to the account and hides sensitive information until the account is unlocked. When applying the lock, users should have the option to set a waiting period that must pass before an account unlock request is approved. Submitting an unlock request should require the user to authenticate, pass an identity verification challenge, and wait for the configured waiting period to pass. Unlock attempts and successful unlocks should be reported according to the guidance in the Reporting section.

We include UI examples below from Coinbase's account lock (figure 4) and Kraken's Global Settings Lock (figure 5).



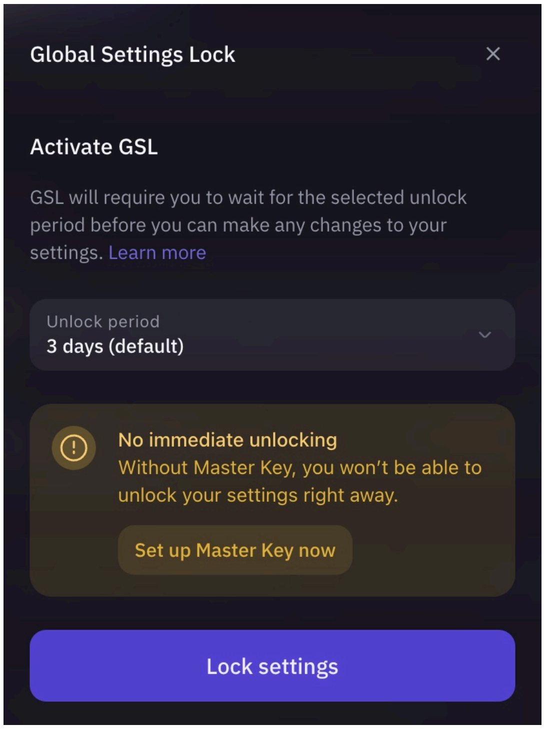
Are you sure you want to lock your account?

Locking your account will sign out all devices, including this one.

No new transactions can occur while the account is locked, but pending crypto transactions may not be canceled.

To unlock your account, you'll need to verify your identity with a selfie and your photo ID. Learn more

Figure 4: Coinbase's account lock feature



Global Settings Lock

Activate GSL

GSL will require you to wait for the selected unlock period before you can make any changes to your settings. Learn more

Unlock period
3 days (default)

No immediate unlocking
Without Master Key, you won't be able to unlock your settings right away.

Figure 5: Kraken's Global Settings Lock feature

Waiting Periods

In addition to reporting activities to the user via multiple channels, some activities should trigger a waiting period during which sensitive actions in the user's account (particularly the withdrawal of funds) are prevented, especially if they do not originate from a trusted device. The duration of the period should be at least 24 hours, and potentially up to a week, depending on the sensitivity of the activity and other risk indicators. This aims to stall a possible account takeover attempt and give the legitimate user time to notice the malicious activity and secure their account.

Actions that should trigger user notifications followed by a waiting period include:

- Signing in on a new device
- Changing the email address
- Changing or resetting the password
- Adding a new MFA method or enabling a previously disabled method
- Creating an API key
- Adding a new payment card or bank account
- Allowlisting a new address for withdrawals
- Requesting an account unlock

For examples of activities that trigger a waiting period on major exchanges, see Reset my password (Coinbase), Why can't I withdraw my crypto? (Gemini), and Why is there a withdrawal hold on my account? (Kraken).

Account Recovery

An account recovery request should garner the highest suspicion and scrutiny from a custodian or service provider. The following steps should be taken to secure the account recovery process:

- Internally, require more than one employee to sign off on privileged actions like account restoration so that a scammer compromising a single support employee's credentials cannot make transactions or take similarly risky actions "on behalf of" customers.
- Record each employee action and all communication with the client in an auditable way, such as in a ticket associated with the account that lists the client communication address and method used.

- If the account being recovered is associated with suspicious activity, notifications should be sent to all of the account's contact methods about both the suspicious activity *and* the recovery attempt.
    - The account should be locked and restricted until the user successfully responds and approves the recovery from at least two of the contact methods on the account.
- Successful recovery should require a human review of a physical, verifiable identification method like a government-issued ID (examples: driver's license, identity card, passport), as well as digital matching of an offline, backup second factor not used for account login.
    - This means that when a user configures second-step authentication, they will need to add at least two methods. The backup method should be either push notification or a hardware token, not SMS.
    - Be aware that verification images can be edited or wholly generated by AI.
    - Know-your-customer (KYC) third party services may automate this identity validation process using, for example, credit history questionnaires and/or image verification.
        - Some KYC services may include ML/AI-based doctored or false credential detection mechanisms.
        - Some KYC services may rely on human-in-the-loop verification.
        - If you choose to trust such a service, ensure it employs technology for detecting image editing.

Guiding End Users

Cryptocurrency platform users can often take personal security precautions to reduce the risk of falling victim to account takeover. However, such precautions require platforms to have already implemented security features (such as MFA) and provide users clear usage guidance. Platforms should support and recommend the following user good practices:

Endpoint security

- Keep your devices updated by installing updates for their operating systems, browsers, and apps as soon as they are available. Installing updates promptly helps protect your devices from known security vulnerabilities.

- Do not log in to personal accounts on public, shared, or work devices; stick to your own trusted devices.

- Use Chrome as your default browser on desktop and laptop devices. Chrome updates itself automatically, recognizes and blocks known phishing sites, and provides site isolation to mitigate exploits delivered by malicious websites and extensions.

- Turn on disk encryption, such as FileVault on macOS or BitLocker on Windows, to prevent access to your files if your device is stolen. iOS and Android devices enable disk encryption by default.

- Configure your devices to automatically lock the screen after 10 minutes of inactivity, and require a password or biometric authentication (such as Touch ID) to unlock it.

- Use a dedicated device for critical accounts (e.g., banking and cryptocurrency funds) if possible. This device should not be used for anything else to reduce the likelihood that it will be compromised by malware. A modern, up-to-date iOS device or Chromebook is a good choice for a dedicated, secure machine.

Password guidance

- Consider using a password manager that you trust; the EFF provides a guide to picking a good password manager here. Choose a random, unique, and memorable password for the password manager account (more information here).

- Generate strong, random passwords using your password manager's built-in functionality or, for example, using the EFF's diceware instructions and wordlist to create a randomly chosen pass*phrase*.

    - A type of strong password that password managers can automatically generate is a random string of mixed-case letters, numbers, and special characters at least 16 characters long.

    - Alternatively, password managers can generate more memorable passwords consisting of randomly selected words; at least five words should be used for the password to be considered strong.

- Never use the same password for more than one account.

    - Password reuse significantly increases the likelihood of falling victim to a credential stuffing attack if your password is found in a data breach.

    - Most password managers will alert you if any of your passwords are not unique or are present in a known data breach.

- If prompted to choose security questions and answers, treat each of your answers like a password. Do not provide factual information for security question answers. Instead, use the password generator in your password manager to create random strings of mixed-case letters, numbers, and special characters at least 16 characters long.

- Use your password manager's autofill feature. This helps mitigate some kinds of phishing attacks by filling passwords automatically only on the expected domain for a website.

"Sign in with <service>" social logins

- Depending on your personal threat model, some social login providers may be more trustworthy than others.
- Ensure that you adequately secure the social login provider account that the platform accesses:
    - The social login account should have a strong password that you store in a password manager.
    - MFA should be enabled and used for the social login account according to the two-factor authentication guidance in this document.
    - Opt in to any additional security measures the social login provider offers, such as Google's Advanced Protection Program.
- If using a social login for the first time and you are asked to grant particular access rights or permissions to the platform where you are logging in (examples: the ability to post to your social media account; the ability to read and create events on your calendar), examine the requested permissions and check that they make sense for what you expect the platform to do on your behalf.
- It is less common for platforms to offer a second factor as part of an authentication flow that relies on a social login option. If the platform offers a second factor with platform-native login and you cannot turn on MFA for your social login provider-side account, prefer the platform-native login over use of any social login.

As an aside, a platform using a social login widget for authentication implicitly outsources part of its authentication security to the social login provider. However, social login security still depends on the platform's ability to securely and correctly integrate the provider's social login library.

Two-factor (two-step/multi-factor) authentication

See Multi-Factor Authentication above.

- Enable MFA for all critical accounts: email, Google, Apple, and Microsoft accounts, cloud storage, ISP, finances, and so on.
- Whenever possible, enable MFA for every online account that supports it.
- Prefer U2F hardware authenticators (e.g., YubiKey), since U2F hardware authenticators are resistant to phishing attacks and cannot be SIM swapped.

- SMS-based MFA is not as secure as other MFA methods, because it offers no protection against phishing or SIM swapping. However, if *no other* MFA method is offered, SMS is better than nothing.

- MFA based on voice recognition is not as secure as other methods, because it can be spoofed through the use of recordings or machine learning. This form of MFA is better than nothing only if *no other* MFA method is offered.

- TOTP-based MFA methods should also be considered less secure, since they can still be phished. However, if only these less-secure MFA methods are supported, prefer TOTP-based methods over SMS or voice recognition.

- If using an authenticator app, ***never*** store the authenticator “seed” or QR code in an insecure form, such as in a file on your computer or in an email draft. If the seed or QR must be backed up somewhere digital, store it only in your password manager.

- If using an authenticator app that supports cloud backup (e.g., Google Authenticator), ensure this feature is disabled. Backing up MFA secrets to a cloud account could lead to an attacker gaining access to your second factor if that account is compromised. Google Authenticator’s cloud backup feature did not originally support end-to-end encryption (E2EE) of MFA secret backups when it was launched in April of 2023. Google has said they intend to eventually add E2EE to Authenticator.

- If the platform provides authentication backup codes or any type of second-factor “seed” secret phrase, store them in a password manager, or print them and store them in a secure physical location, as recommended by Discord. If printing such codes or phrases, permanently delete the local digital copy once a physical copy has been created and stored.

- If using Authy for MFA, turn off the Multi-Device feature to prevent an attacker from using a SIM swap to access your MFA codes. See Gemini’s instructions on disabling Multi-Device.

- Passkeys are either an acceptable second factor *or* an acceptable password replacement. Using just a passkey without a separate second factor is not sufficient to replace a password and a separate second factor.

SIM swap awareness and prevention

Platforms should clearly document the signs of a SIM swap attack so that users can recognize them:

- Unsolicited texts, calls, or emails about changes to cellular service.

- Unexpected total loss of cellular service when not out of range, such as a persistent inability to send or receive text messages and phone calls even after changing physical location.
- A sudden inability to access accounts, which could indicate that an attacker has taken over those accounts, potentially as the result of a SIM swap.

If you recognize these signs, immediately contact your provider by dialing 611 to ensure an unauthorized SIM swap did not occur. If 611 is unreachable, talk to a representative in person.

Proper use of two-factor authentication—particularly avoiding SMS-based MFA—usually prevents a successful SIM swap from leading to an account takeover. Additionally, some providers allow customers to opt in to additional authentication, which can hinder a SIM swap attempt. This is typically implemented by setting a PIN that users must provide to customer support before any service change requests can be approved. Users should enable this feature if their provider supports it—instructions on how to do so for popular providers are given below:

- AT&T: Learn about account passcodes
- T-Mobile: Update your Customer PIN/Passcode
- Verizon: Verizon mobile Account PIN FAQs and How do I set up Number Lock to protect my number from being moved without my permission?