GUIDE

Authentication Hardening Guide for Web3 Teams

Phishing-resistant MFA, passkeys, and the recovery paths that defeat them

Prepared for	Web3 teams	Classification	Public
Prepared by	Oak Security	Date	2026-06-17

Oak Security GmbH

1. What This Guide Is For

This guide is for everyone in a Web3 team who logs into something that matters: email, source control, the cloud console, the identity provider, an exchange, a custody UI, the domain registrar. You do not need to be technical to use it. You need to know which login methods survive a real attacker and which only look like they do.

Every other control depends on authentication. Zero Trust, device hygiene, key custody, smart-contract opsec all assume the system knows which human is on the other end. Authentication is where that determination is made, and it is where a large share of Web3 incidents actually break.

The question is not whether the team has MFA. The question is whether the MFA survives an attacker who is already in the conversation. SMS two-factor and a FIDO2 hardware key tick the same box on a risk register. One is trivially phishable in any adversary-in-the-middle (AiTM) attack. The other refuses by design. A team that has confused the two has, in operational terms, no MFA on the accounts that matter.

> **NOTE**
>
> **Baseline rule:** real MFA combines factors from two different families, at least one of which is phishing-resistant. For anything that controls money, code, or identity, that means a FIDO2/WebAuthn hardware key, not SMS, not a TOTP app, not a push prompt.

The attacker's goal is almost never to crack the password. It is to bypass the authentication flow: phish a code, steal a session cookie, exhaust push approvals, swap a SIM, and walk past whatever was supposed to happen at the front door. This guide ranks the options honestly, then tells you what to deploy and what to retire.

2. The Four Factor Families

Authentication factors fall into four families, distinguished by the kind of evidence the user presents. Real MFA means evidence from at least two different families. Two passwords, or a password and a security question, are one factor used twice.

Family	Examples	Phishable?	Key weakness
Knowledge (something you know)	Passwords, PINs, secret questions	Yes, by definition	If the user can type it, an attacker who controls the form captures it.
Possession, phone-based (something you have, via a phone)	SMS OTP, TOTP apps, push approvals	Yes	Inherits every phone weakness; SMS adds SIM-swap.
Possession, hardware key (something you have, purpose-built)	FIDO2/U2F keys (YubiKey, SoloKeys, Nitrokey, Token2)	No	Origin-bound; refuses to sign for the wrong domain.
Inherence (something you are)	Fingerprint, face, voice, iris	Indirect	Not revocable. A leaked password can be changed; a fingerprint cannot.

Knowledge factors are phishable because the human, shown a convincing prompt, supplies what the prompt asks for. No policy prevents this.

Phone-based possession factors inherit every weakness of the phone (malware, lock-screen reading, replay) and of the channel. TOTP is the six-digit code rolling every thirty seconds in Google Authenticator, Authy, or Aegis, generated from a shared secret. Push is the Approve/Deny prompt from Microsoft Authenticator, Okta Verify, or Duo. SMS additionally inherits SIM-swap: convince the carrier to port the number and every code arrives at the attacker.

Hardware-key possession is the family that changes the threat model. A FIDO2 authenticator is origin-bound: when the browser asks it to authenticate, it states the exact domain, and the key produces a signature valid only for that domain. A phishing proxy on a

lookalike domain gets a signature the real verifier rejects. The user cannot hand a usable credential to the wrong site, no matter how convincing the fake.

Inherence rarely authenticates to a service directly. The fingerprint unlocks a key that does. Biometrics are usable (the user always has a finger and a face) but not revocable.

A fifth, weaker class is context (somewhere, some-device): IP allow-lists, geolocation, MDM attestation, impossible-travel. Context supplements other factors. It is not a factor on its own. An attacker on the same coffee-shop Wi-Fi passes the IP check.

3. Phishing-Resistance Is the Property That Matters

The single most useful question to ask of any factor: what happens if the user tries to use it on the wrong site?

- A **password** fails immediately. The user types it into the attacker's form; the attacker has it.
- A **TOTP code** fails almost as fast. The user types six digits into a fake prompt; the attacker's proxy forwards them to the real site inside the thirty-second window. No signal that anything went wrong.
- A **push approval** fails on the tap. The prompt arrives, the user taps Approve, and the attacker who initiated the login moments before is authenticated. The push says nothing about which login is being approved.
- A **FIDO2/WebAuthn factor refuses.** The browser tells the key to sign for `login.oakco-security[.]com` (a lookalike). The key was enrolled at `login.oaksecurity.io`, a different string. It has nothing to sign with for the wrong origin and produces no usable response. The phishing flow ends.

The property that separates the last factor from the rest is **origin-binding**. The authenticator signs the domain the browser shows it, and only that domain. The binding is enforced inside the authenticator, where the user has no opportunity to override it. There is no mistype, mis-tap, or trick that produces a signature for the wrong place. Hardware keys provide origin-binding through FIDO2. Platform passkeys provide it through WebAuthn. Nothing else in widespread deployment provides it at all.

4. The MFA Strength Ladder

Arrange the realistic configurations from worst to best on one ladder.

Rung	Configuration	Phishing-resistant?	Notes
1	Password alone	No	One phishable factor.
2	Password + SMS OTP	No	Cheapest to defeat; SIM-swap or AiTM relay.
3	Password + TOTP	No	A step up from SMS, not the destination.
4	Password + push (incl. number-matching)	No / partial	Number-matching adds friction, not immunity.
5	Device-bound platform passkey, sole factor	Yes, but device-bound	Origin-bound, but on the same host as the browser and any malware. Single-device reliance.
6	Password + FIDO2 hardware key (second factor)	Yes	Key offline, touch required, origin-bound. The floor for accounts that matter.
7	Password + passkey resident on a FIDO2 hardware key	Yes	Both factors off the host. PIN rate-limited in firmware. The recommended endpoint.

Rungs 1 through 4 are all phishable. They differ in cost to the attacker, not in whether the attack works. None withstand a modern AiTM proxy on a fresh domain.

Rung 5 is genuinely better than any phishable 2FA: a device-bound platform passkey is origin-bound and cannot be typed into the wrong site. But it lives on the same general-purpose device as the browser, email, and any malware. An unlocked keychain exports it. A

coerced biometric unlock (the wrench attack) uses it. The single-device reliance is the weakness.

Rungs 6 and 7 move the secret off the host. Rung 6 is a password plus a FIDO2 hardware key as the second factor: offline, touch-required, origin-bound, unreachable by an AiTM relay. Rung 7 puts the passkey itself on the hardware key as a resident credential with a PIN, keeping the password as the second factor. Both factors are off the host. The PIN is rate-limited inside the key's firmware; a few wrong attempts and the key wipes itself. Losing the laptop changes nothing.

> **NOTE**
>
> **Oak recommendation:** rung 7 wherever the service supports it (password plus a passkey resident on a FIDO2 hardware key). Rung 6 everywhere else. Rungs 1 through 4 only for accounts that genuinely do not matter, and never as the primary factor on anything controlling money, code, or identity.

5. Doing MFA Properly

The operational picture follows from the ladder.

- **No SMS 2FA.** The SIM-swap risk is real, repeatable, and frequently exploited.
- **Authenticator apps are acceptable but not phishing-resistant.** A step up from SMS, not the destination.
- **Hardware security keys are the destination.** FIDO2 or U2F is the protocol.
- **At least three keys per user**, kept in distinct physical locations, so loss or destruction of one does not lock the user out.
- **PIN or biometric activation on the key itself.** A stolen key without the PIN is useless.
- **The hardware key must be the actual second factor**, not merely registered alongside push. A key enrolled next to an active SMS or push fallback downgrades to the weaker method's strength, because the attacker chooses the path.

6. The AiTM Attack That Defeats Phishable Factors

The reason hardware-key authentication is a baseline, not a luxury, is clear once you look at the contemporary phishing kit. The class to understand is adversary-in-the-middle. The tooling (Evilginx, Modlishka, Muraena) has commoditised it: any adversary who can register a domain can run it.

The flow:

1. The user receives a phishing link pointing to a lookalike domain. It has a valid TLS certificate (Let's Encrypt issues one the day the domain is registered), so the browser shows the padlock.
2. The lookalike is not a static fake page. It is a reverse proxy. It fetches the real login page and relays it in real time. The page the user sees is literally the real page: pixel-perfect, because it is the original.
3. The user types username and password. The proxy captures both and forwards them. The real site issues a 2FA challenge (TOTP, push, SMS). The proxy relays it. The user satisfies it. The proxy captures the response and forwards it.
4. The real site issues a session cookie. The proxy captures it and now has authenticated access as the user. Nothing looks unusual from either the site's or the user's perspective.

Note what each layer failed to stop. Browser phishing-domain detection: bypassed, the domain was fresh and not yet blocklisted. The TLS padlock check: bypassed, the certificate was real. The lookalike-domain check: bypassed, the user did not stare at the URL bar. User education does not scale against this. Telling users to be careful asks them to perform URL-bar inspection that experienced security professionals fail under time pressure.

What stops AiTM is not vigilance but a factor that refuses to sign for the wrong domain. FIDO2 and WebAuthn do exactly this: the browser states the operating domain, the authenticator compares it to the enrolment domain, and a mismatch produces nothing usable. The AiTM proxy, operating on a different domain by definition, has no response to relay.

Backstops complement the authenticator. Conditional access at the IdP can require a managed device and a hardware key together for sensitive apps. Short session-cookie TTLs reduce the value of a stolen cookie. Step-up re-authentication for sensitive actions (exporting a vault, rotating a key, sending funds) forces a fresh authentication at the moment that matters. If a user has entered credentials somewhere suspicious, assume AiTM: rotate the password, revoke every active session at the IdP (which invalidates the captured cookie), then rotate the 2FA secret. Rotating the password without revoking sessions does nothing; the cookie is still valid.

7. Passkeys, in Detail

A passkey is a WebAuthn credential: a keypair generated on the user's authenticator, public half registered with the service, private half never leaving the authenticator. The service issues a challenge only the private-key holder can sign, and the protocol is origin-bound, so the credential cannot be misused on a phishing site. As a replacement for the password, a passkey is unambiguously better than what it replaces.

The harder question is where the passkey lives. The industry collapses three different deployments under one word. They are not interchangeable.

Deployment	Where the key lives	Verdict
(a) Synced passkey	Cloud-synced store: iCloud Keychain, Google Password Manager, 1Password sync	Convenient. Bounded by the security of the sync account, whose recovery often falls back to a password and a phone number, defeating the phishing-resistant chain at the recovery layer.
(b) Device-bound platform passkey	A specific device's TPM or Secure Enclave; does not sync	The floor. Cannot be exfiltrated across devices, but sits on the same host as the browser and any malware. An unlocked keychain exports it; a wrench attack uses it.
(c) Passkey on a FIDO2 hardware key, as a second factor	Resident credential on a YubiKey or equivalent, PIN-protected, used alongside the password	The recommendation. Both factors off the host, PIN rate-limited in firmware, physical touch required. Laptop theft or infection affects neither factor.

NOTE

Oak recommendation: option (c) wherever the service supports both passkeys and passwords. Option (b) as the floor: better than phishable 2FA, fine for accounts that do not control money, code, or identity. Option (a) only for low-stakes accounts where convenience outweighs the marginal security loss. Regardless of option, enrol at least three hardware keys per person in distinct physical locations.

8. The Few Passwords That Still Matter

In a passkey-first world, a small number of passwords carry real weight because they sit at the bottom of the trust chain. Everything else is recoverable from them, so they deserve disproportionate care.

- **Password-manager master password.** The gatekeeper to every other credential.
- **Disk-encryption passphrase** (FileVault, BitLocker, LUKS). Typed once at boot, protects everything at rest.
- **Seed-phrase passphrase** (the BIP-39 25th word). The difference between stolen device and stolen funds.
- **SSH passphrase** on a production key. The second factor when the host is compromised.

The rules for these: high entropy (at least six diceware words or sixteen random characters); never reused; never typed on an untrusted device; memorised, or written on paper and stored the way you would store cash. No password manager stores them. If it did, one compromise of the manager would own everything below it.

9. Session Cookies: The Stolen Factor

After authentication, the service hands the browser a session cookie. Every subsequent request uses the cookie instead of re-authenticating: no password, no 2FA, no challenge. The cookie is the authentication for the duration of its validity.

A stolen cookie therefore equals full account access for as long as it lives, and it bypasses every authentication factor by definition, because the authentication already happened. AiTM proxies harvest cookies. Stealer malware (RedLine, Vidar, Lumma) targets the cookie store of every browser on a compromised host. Browser-extension compromises read cookies directly. The cookie is the prize, not the password.

Mitigations:

- Short session TTLs on sensitive applications. No ″stay signed in″ forever.
- Mandatory re-authentication for sensitive actions (exports, rotations, funds movement).
- Token binding (DPoP, WebAuthn-backed token binding) where supported, tying the cookie to the device cryptographically so a stolen cookie alone fails on another device.
- On suspicion, ″sign out everywhere″ at the IdP (the Active sessions or Sign out other sessions control, not the local logout), which invalidates cookies across every connected application at once.

10. SSO and Identity-Provider Blast Radius

Single sign-on consolidates authentication across many SaaS apps behind one identity provider (Google Workspace, Microsoft Entra, Okta). The benefit is real: fewer passwords, faster offboarding, one place to enforce MFA policy. The cost is equal: one compromise is N compromises. The IdP is the master key to every connected application.

Defend the IdP like a treasury.

- **Hardware-key MFA mandatory** for every administrator and every standard user. No temporary TOTP fallback ″while we sort out the keys″; the temporary fallback is the path the attacker uses.
- **Break-glass accounts** configured outside SSO, with hardware keys stored in a safe, used only when the primary admin path is compromised, and audited on every use.
- **Conditional access** requiring a managed device plus a hardware key together for privileged applications: custody UI, GitHub organisation owner, domain registrar, DNS provider, email hosting.
- **Monitor** new MFA enrolments, new OAuth grants, and impossible-travel events. These are the first signals of a successful IdP compromise, often visible hours before the consequences.

11. Account Recovery: The Back Door

Every "I lost my phone / I lost my key" flow is a parallel authentication path, and the parallel path is often weaker than the primary. The classic failure: a hardware-key-protected account whose recovery email has only SMS MFA. The attacker resets the email, then resets everything that hangs off it. The recovery flow defeated the primary flow without ever attacking it.

The audit is straightforward but rarely performed.

- For every high-value account, know exactly what resets the password, what resets the 2FA, and whether the recovery email is as hardened as the primary.
- Remove phone-number recovery wherever the service permits it. Replace it with backup hardware keys and printed recovery codes in offline storage.
- Print recovery codes on paper, stored alongside other emergency documents (passports, wills). Never in the password manager that they recover, never photographed, never in cloud notes.
- Configure a trusted-recovery contact where supported (Google trusted contacts, Apple recovery contacts), and make it someone other than the account owner.
- Test the recovery flow on a low-stakes account periodically. The moment of need is the worst time to discover it does not work.

12. Push Fatigue and Number Matching

Classic push notifications (Approve login? with one tap) improved on SMS until attackers found push-bombing. The attack is artless: holding the password, the attacker fires dozens of push prompts at three in the morning until the half-asleep user taps Approve to make the noise stop. The Uber 2022 breach is the best-known incident, not the only one.

Number matching (the two-digit code the user types back into the prompt) improves matters without fixing them. The user is forced into a moment of intent rather than a reflex, but the push is still phishable: the user types the number the fake page shows. Geographic and application context in the prompt adds friction without changing the threat model. The attacker who reached the password and the push channel satisfies the additional context most of the time.

> **NOTE**
>
> **Oak position:** push with number-matching is acceptable as a step-down factor for low-privilege applications. It is not acceptable for the IdP admin role, the custody UI, or anything touching production infrastructure. Those require hardware keys.

13. Authentication Hardening Checklist

Use this when onboarding a team member, hardening an account, or auditing posture quarterly.

Password Hygiene (the Floor)

- [ ] Password manager in use (1Password, Bitwarden, Vaultwarden), one unique random password per account, no reuse across services.
- [ ] Master password long (>= 16 chars or six diceware words), unique, never typed on an untrusted device.
- [ ] Password manager protected by hardware-key MFA, not SMS or TOTP.
- [ ] No browser-saved passwords for sensitive accounts.
- [ ] Different usernames for personal and professional accounts where practical, to resist correlation.

MFA Strength

- [ ] Every privileged account uses FIDO2/WebAuthn (rung 6 or 7): treasury, deployment, IdP, GitHub, exchange, custody.
- [ ] At least three hardware keys per privileged user, registered everywhere, kept in distinct physical locations.
- [ ] SMS removed as a factor and as a backup wherever the platform allows.
- [ ] No TOTP-as-backup for hardware-key accounts (the backup becomes the weakest link).
- [ ] The hardware key is the actual second factor, not merely registered alongside push.
- [ ] PIN or biometric activation enabled on each key.

Account-by-Account

- [ ] Email IdP (Google Workspace, Microsoft 365): hardware-key MFA, recovery codes printed offline, recovery email is not a personal Gmail.
- [ ] GitHub / GitLab: hardware-key MFA enforced at org level; SSH keys hardware-backed where possible.
- [ ] Cloud (AWS, GCP, Azure): root locked, hardware-key MFA, root credentials in a safe; day-to-day via federated SSO with a hardware key.
- [ ] Exchanges and custody: hardware-key MFA, withdrawal allow-list configured, withdrawals cross-checked on a separate channel.
- [ ] Domain registrar: hardware-key MFA, registrar lock enabled, transfers require manual approval.
- [ ] Slack / Discord / Telegram: 2FA enforced, backup codes stored offline.

AiTM and Session Defences

- [ ] Conditional access: managed device plus hardware key for privileged apps; known networks and countries where supported.
- [ ] Session lifetime <= 24 hours for high-value accounts; no perpetual ”stay signed in”.
- [ ] Step-up re-authentication required for sensitive actions (exports, key rotation, funds movement).
- [ ] Token binding (DPoP / WebAuthn) enabled where the service supports it.

Recovery

- [ ] Recovery codes printed or handwritten, stored in a safe, never in the password manager they recover.
- [ ] Phone-number recovery removed wherever permitted; backup hardware keys enrolled instead.
- [ ] Trusted-recovery contact configured where supported, and not the account owner.
- [ ] Account-takeover drill once per year: lose the primary key, recover from backup, document what was unclear.

Onboarding a New Team Member

- [ ] Three hardware keys ordered and shipped.
- [ ] Day one: keys registered on email IdP, GitHub, password manager, Slack, cloud SSO.
- [ ] SMS factor removed from every newly created account at registration.
- [ ] Recovery codes generated, printed, sealed, stored.
- [ ] Documented: which key is primary, which are backups, where backups are kept.

14. Common Mistakes

The recurring failures are unglamorous and persistent.

- Weak passwords and password reuse, the basis of every credential-stuffing attack.
- Shortcuts on accounts perceived as unimportant, which turn out to be the recovery path for the important ones.
- Credential sharing between team members, destroying the audit trail and creating revocation problems on offboarding.
- No MFA at all on accounts the team forgot to enrol.
- Weak MFA (SMS or email codes) counted as if it were strong.
- MFA without backup: a single hardware key, lost on a flight, locks the user out for days.
- A hardware key registered alongside SMS or push on the same account, which downgrades the account to the weaker method.
- Backup codes stored in the cloud storage of the account being recovered.
- Synced password-manager passkeys treated as equivalent to hardware-bound credentials.
- A personal phone number used as the second factor on a corporate account (SIM-swap risk).
- Rotating a password after a suspected AiTM compromise without revoking sessions at the IdP, leaving the stolen cookie valid.