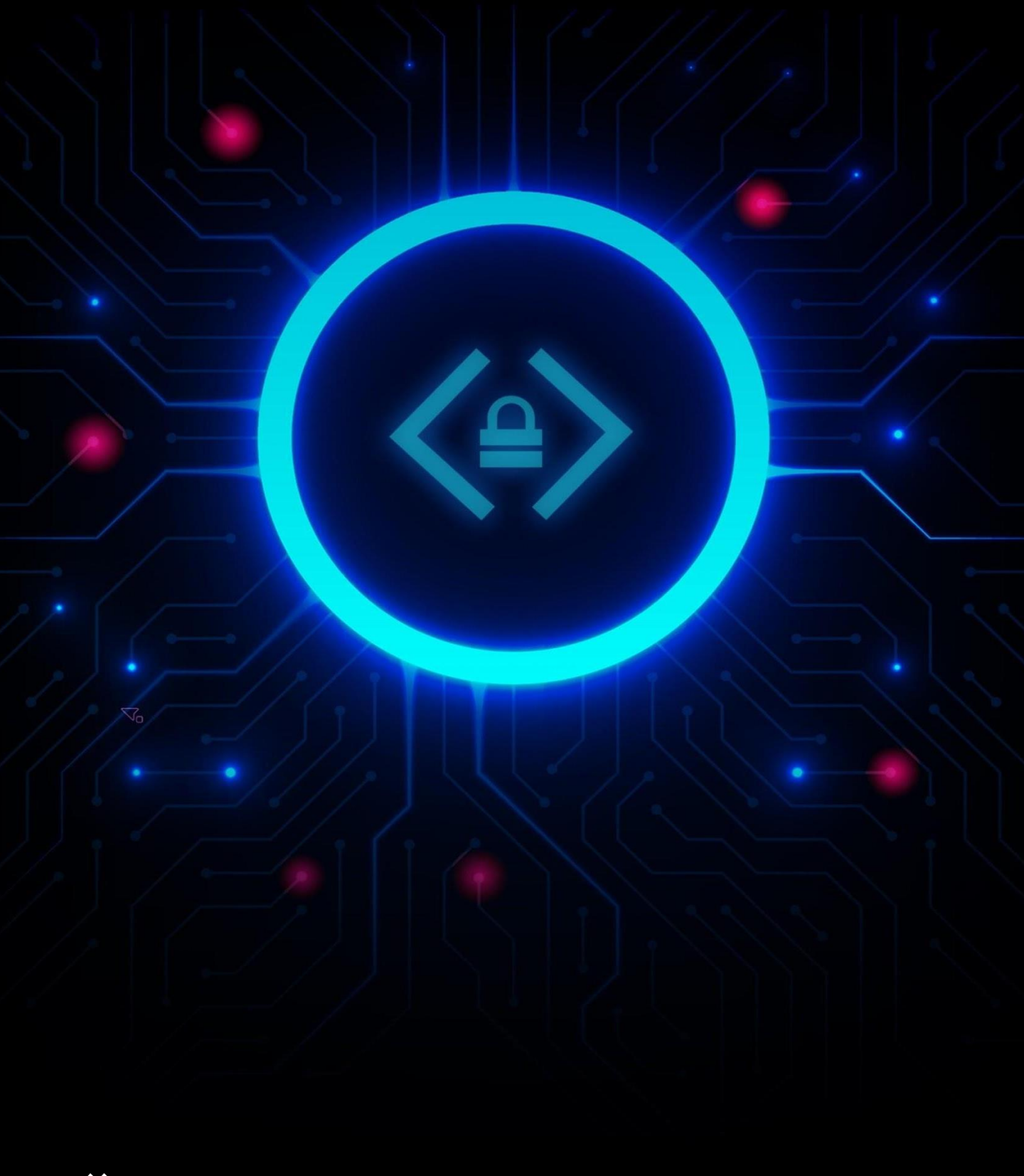


MEMCYCO

Technology Teardown

FIDO2 ‘Passwordless’ Authentication: The End of Credentials Phishing Attacks?

MEMCYCO

Is ‘Passwordless’ authentication everything it promises to be?

As much as they’ve ‘gotten us by’, it's no great secret that traditional passwords aren’t bulletproof. In fact, the quest to end passwords once and for all is almost as old as their innovation.

Now, FIDO2 ‘passwordless’ technology, based on methods such as facial recognition and fingerprints, promises to finally take us across the Rubicon, potentially rendering credential harvesting phishing attacks obsolete. It’s a bold pitch.

Are Fraud, Security, and Risk teams about to enjoy a golden era of fewer attacks and lighter workloads?

Despite its name, FIDO2 is the third standard from the FIDO Alliance, after the FIDO Universal Second Factor (UAF) and FIDO Universal Authentication Framework.

Its core aim is to eliminate the use of passwords, with license-free standards for ‘passwordless’ authentication.

In this article

ATO & DIGITAL IMPERSONATION

We’ll examine the use of passwords in one of the most prevalent attack scenarios: account takeover (ATO) via phishing, and similar attack vectors.

Specifically, we’ll focus on ATO via website and SSO sign-in page impersonation. You know how it goes; a rogue link gets clicked, fake page looks legit, credentials get harvested. It’s downhill from there; at worst, ATO attack meltdowns can make cataclysmic headlines.

THE ‘PASSWORDLESS’ PITCH

We’ll size up where the passwordless pitch – of ‘phishing-free’ authentication lands on the scale between ‘fact’ and ‘fiction’. Can FIDO 2 passkeys really offer total protection against phishing-based attacks on customers and company crown jewels?

Finally, we’ll touch on some of the common attack methods that the FIDO2 standard of passkeys remain open to, and why, standalone, they’re not enough.

MEMCYCO

Most see to the ATO fire. Few smell the ATO smoke.

The heart-stopping implications of ATO are worth losing sleep over. Businesses yet to experience them will likely have seen the headlines. Thanks to ATO's cataclysmic nature, the digital impersonation methods often at their epicentre tend to take a back seat in the conversation.

Customer-targeted

Website impersonation ATO

1. **The bait:** Customer receives fake email or SMS with malicious link, mimicking a legit website or service they may be affiliated with. Often, they're prompted to sign into their account. The trap is set
2. **The trap:** On clicking link, customer arrives at the fake site, often virtually identical to the real thing. Of course, this 'spoof' page is controlled by the attacker.
3. **Credentials harvesting:** Convinced the page is legit be real, customer attempt login, possibly including multi-factor authentication (MFA), handing their credentials to attackers in the process.
4. **Customer ATO:** With customer credentials captured, attackers can now access their account on the real site, or even create a passkey if they can bypass or emulate MFA.
5. **Exploitation:** Once inside, attackers can lock the door, exploiting account access at their leisure; stealing personal and financial information, making fraudulent transactions; the world is their oyster.

Employee-targeted

SSO page impersonation ATO

1. **The bait:** Employee receives fake email or SMS uncannily impersonating a routine company communication, inviting them to sign into their account via what they think is the company SSO sign-in page.
2. **The trap:** Following the link, the employee lands on a fraudulent SSO sign-in page; a mirror image of the company's legitimate SSO portal.
3. **Credentials harvesting:** Convinced of SSO page authenticity, employee inputs single sign-on credentials, unwittingly handing them to attackers. If needed, they may complete MFA, unknowingly tightening the noose.
4. **Employee ATO:** With employee credentials in hand, attackers can stealthily access the employee's account on the legit company site, laying the groundwork for a silent account takeover.
5. **Exploitation:** Now inside the account, attackers are free to roam, browsing company crown jewels, fulfilling any number of nefarious objectives, while sealing the door behind them.

MEMCYCO

The jargon: ‘passwordless’ FIDO2 passkeys explained

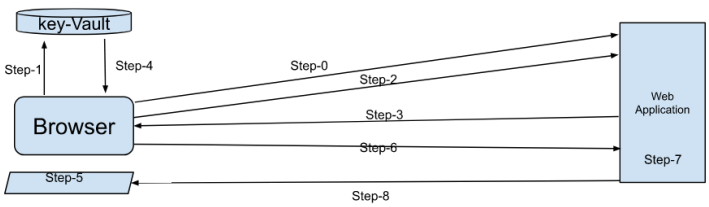
Why do questions around FIDO2 passkeys efficacy matter? The biggest names in software are embracing FIDO2 passkeys, driving adoption with the promise of eliminating phishing-based attacks. FIDO2 passkey innovations offer impressive defenses, but there remain a couple of important caveats to consider.

“ Let me get this straight: no more phishing fraud or scammed customers? ”

...Maybe that’s a stretch

In today’s on-demand world, passwords create user frustration. Online, seconds can feel like an eternity.

Unlike password-based authentication, FIDO2 passkeys promise frictionless convenience. They also offer notable defense against phishing-based ATO attacks. That said, they’re not bulletproof for stopping phishing-based attacks.



0 - Passkey Establishment
1 - Key-pair generated & Stored in key-vault. Public-key stored on the backend.
2 - Login request is sent to the web-application with the username
3 - Challenge generated by Web Application
4 - Private-key fetched from key-vault (fingerprint/pin authentication)
5 - Challenge is signed with the private-key
6 - Signed-Challenge is sent to backend
7 - Web-Application verifies the signature
8 - Web-Application issues a session-token cookie

How FIDO2 passkeys work

- ✓ **User attempts to sign into a web application**. Passkey mechanism is initiated, relying on user/password authentication only for the first sign-in. Whenever the device is lost, broken, or replaced, the passkey mechanism is reinitiated. (Step 0)
- ✓ **Once user is authenticated with user/password pair** (and possibly MFA), the ‘passwordless’ passkey mechanism generates a key pair on the device, stores the private key in a key vault, and sends the public key to the backend server of the web application. (Step 1)
- ✓ **Subsequent sign-ins to the web application** require users to input username only. A login request is then sent to the backend (Step 2).
- ✓ **At that point, the backend generates a ‘challenge’** (i.e., a random string) and sends it to the browser (Step 3). The browser then performs ‘passwordless’ authentication using methods like fingerprint, facial recognition, PIN, etc., vis-a-vis the operating system.
- ✓ **Once the user has authenticated to the OS**, the browser gets access to the private key (Step 4) and uses it to sign the challenge it got from the backend (Step 5).
- ✓ **Browser sends data packet with signed challenge** and username to the web application backend (Step 6).
- ✓ **Backend verifies signed challenge using public key** (Step 7), before issuing a session token, allowing further transactions (Step 8).

Still with us? Congratulations
You just completed a crash course in passwordless authentication (You can zoom back out now).

MEMCYCO

No more memorizing 16 passwords? No credentials to steal? What's not to like?

The 'passwordless pitch' of ending phishing-based threats is certainly mouthwatering. And those pivoting to passwordless are making a solid decision to boost ATO defenses. That said, there are important caveats worth factoring into the FIDO2 equation.

'Passwordless' may leave an open point of failure

It may come as something of a surprise to those unaware that 'passwordless' passkeys still rely on regular passwords being inputted by users whenever a device connects to any given web application for the first time.

This step is repeated whenever a device is lost, broken, or replaced, giving fraudsters the chance to imitate account issues, duping users into inputting passwords.

'Passwordless' may mean greater company liability

When ATO happens in a 'passwordless' environment, the company is fully liable, since the user doesn't influence the authentication process that's handled behind the scenes. Thus, tighter security measures are needed.

With 'old school' passwords, liability is split between user and company, since users has an active role in inputting credentials password on each sign-in.

Can DNS cache poisoning overcome FIDO2 passkeys?

In theory, yes. More worryingly, also in practice. This would involve entering false data into a DNS cache, thus redirecting users to an impersonating site. But by itself, that's insufficient. There's still the SSL certificate validation to deal with to subvert browser warnings. To achieve this, scammers would need to employ one of three strategies – any one of which create a formidable attack vector capable of subverting 'passwordless' defenses when combined with DNS cache poisoning.

1. Self-signed certificates

Attackers may use self-signed certs, hoping users will overlook browser warnings. Social engineering techniques (like convincing users to trust these certificates via a compromised hotel Wi-Fi for 'enhanced security) can increase the chances of user acceptance.

2. Rogue certificates

These are legit certs for a real domain, stolen or deceitfully obtained. Attackers may obtain several, exploiting poor key management practices, and cybercrime ecosystems even trades in rogue-certs, allowing scammers to 'spoof' trusted sites seamlessly.

3. Rogue certificate authorities

Sophisticated attackers might even set up a fake certificate authority, mimicking legit ones and issuing SSL certificates from these rogue CAs while manipulating DNS responses for real root CA addresses. This way, the browser is tricked into verifying the fraudulent certs.

MEMCYCO

FIDO2: a formidable fortress. But there is one small catch...

Just as ‘passwordless’ authentication happens ‘behind the scenes’, attack vectors able to crack it also work out of sight. Even in theory, nobody can detect them, since there's no interaction with human knowledge and response. That means no user raises the alarm, no SOC team gets a notification. If carried out carefully, such attack vectors leave zero trace – making real-time FIDO2 fortification critical.

Cracking the FIDO2 vaults

Evil twin - Attackers establish a Wi-Fi network with the same (or similar) SSID name as the Wi-Fi network of the target company. Attackers make certain that their network has a stronger signal than that of the real Wi-Fi network, and usually also carry out a DDoS attack on the real Wi-Fi network, all with the aim of getting the employees to connect through the fake Wi-Fi network.

Similar attacks can also be performed on hotel and restaurant Wi-Fi networks. Since attackers have full control of the fake network, they can set up DNS resolution as they wish, creating a similar scenario to DNS cache poisoning.

Man-in -The-Browser (MiTB) - Carried out using browser extensions that, while appearing legit, can perform attacks like Man-in-the-Middle (MiTM) attacks. The browser extension covertly sets up a network listener on a form submission even during interactions with the legit backend server. The listener takes over data packets with the signed challenge, sending them to a malicious server connected to the browser extension. That server now has all it needs to initiate a valid session via the web app.

Session hijacking - Popular with modern scammers. As mentioned, all authentication methods, including passwords and passkeys, issue a session token that browsers use for every transaction after sign-in. Session token can be hijacked using techniques like the ones listed, as well as other elements such as OS-listeners, malware, network sniffers etc. Possession of such a token allows the attacker to bypass the authentication process and perform an ATO. Worse, Hijacked session tokens can be easily obtained on the darknet.

Supply chain integrity - Many web-applications use third party JavaScript libraries offering standard services (Captcha, GDPR compliance, accessibility scripts etc.) If those libraries get hacked, attackers are free to carry out MiTB attacks at will.

The verdict

‘Passwordless’ offers worthy defenses against phishing-based attacks. That said, bad actors with enough grit and determination can and will get around them. When they do, nobody will know to sound the alarm.

To seal the gaps, the fact remains that you still need an extra real-time layer of detection, for catching threats that slip silently past the FIDO2 net.