



VULNERABILITY REPORT 1723

12th September 2023

FIDO2 registration/deployment related Vulnerability

Report on potential FIDO2 deployment Vulnerability related during independent pentest.

INTRODUCTION


This vulnerability was discovered by 4 researchers, who presented it in their research article titled “Evaluating the Security Posture of Real-World FIDO2 Deployments”.

The research specifically targeted FIDO2 authenticators and proves that given a combination of social engineering & phishing scenarios, which will enable the registration of a malicious authenticator, which in turn paves way for an attacker to take control over user’s accounts or subsequently the RP server itself. The researchers are Dhruv Kuchhal, Muhammad Saad, Adam Oest and Frank Li.

DISCLAIMER

Recipients may share this paper with peers and partner organizations. Information in this paper can be circulated within the FIDO Alliance community. Information contained in this document may be released outside of FIDO Certification Secretariat, the Researcher in question, FIDO Accredited Labs and the SRWG in the FIDO Alliance.

REVISION HISTORY

| Revision | Date | Description | Triage Level |
|----------|---------------------------------|-----------------|-------------------------------------------------------------------------------------------|
| V1.0 | 12 th September 2023 | Initial version | GREEN  |

VULNERABILITY INFORMATION

This vulnerability exploits the client device's weak security, through which an attacker can inject malware. Also, it explains the absence of proper verification by the Relying Parties when registering the authenticators, and thus paving way for the attacker to gain access to a user account unauthorized by injecting malware. This is done through a complex phishing scenario and social engineering scenario, where a device or its browser extensions, where the authenticator resides, used by the legitimate user is compromised.

AFFECTED SOFTWARE/HARDWARE

All FIDO authenticators (FIDO) and CTAP (version 2.0 and 2.1) at L1, as a means of connecting to carry out an authentication related operation.

ATTACK COMPLEXITY (Methods, Techniques and Tools)

Method 1: REGISTRATION PHASE ATTACKS

Considers the threats where the attacker aims to register their malicious (i.e., attacker-controlled) authenticator to a user's account. This allows the attacker to authenticate into the account at will, effectively taking over the account. To gain access to a user's account, the attacker could either utilize existing account takeover techniques to gain access themselves, or they could deploy malware to the user's device which already has the access required.

Scenario 1:

In the absence of proper verification of the authenticator by the Relying Parties, an attacker can exploit non-FIDO2 credentials, such as passwords (e.g., via phishing), to take over an account and register a malicious authenticator.

Scenario 2:

A user could inadvertently install a malicious user-level (i.e., non-root) application that misrepresents itself as the target RP's official application. (phishing).

For this scenario to succeed:

- the attacker must trick the user into registering the malicious authenticator while thinking they are registering their real authenticator (or wait for the user to conduct an action that would require the same form of user verification)
- When this occurs, the application could intercept the legitimate FIDO2 registration request and instead register a malicious virtual authenticator embedded in the application itself (with the user verifying the malicious authenticator registration, thinking they are registering their actual authenticator or doing the user verification for a different action).
- Once registered, the malware could report the FIDO2 credentials (visible to the malware in plaintext) back to the attacker, allowing the attacker to clone a similar virtual authenticator with the same stolen credentials and remotely compromise the user's account.

Scenario 3:

Overprivileged devices, such as rooted Android devices or jailbroken iOS devices, allow malware to circumvent built-in security mechanisms enforced by the OS. A root-level malware can intercept and respond to a FIDO2 registration request from a malicious virtual authenticator. A root-level malware can intercept and respond to a FIDO2 registration request from a malicious virtual authenticator. Unlike the malware attack described earlier, here the user can still with the legitimate RP application, but their FIDO2 operations are hijacked at the root level.

Scenario 4:

The security posture of a user's account could significantly weaken if their authenticator allows for weak user verification methods (i.e., PIN), which malware could successfully bypass. This is due to the fact that the weak verifications methods such as PIN could be easily guessed or observed.

Scenario 5:

For a legitimate authenticator, if user verification can be circumvented by malware, or its attestation or private keys can be compromised, either remotely or with physical access to the authenticator, the authenticator is known to be vulnerable.

Scenario 6:

Users could install an application (or a Chrome extension) that includes an authenticator and allows the user to manage (export and sync) their FIDO2 credentials. While some users might choose this for its usability, it provides little security guarantees, as the FIDO credentials reside in user space and can be stolen by malware. Stolen credentials can be easily seeded in a cloned virtual authenticator for the attacker to gain access. This happens

because of the fact that the RPs allow virtual authenticators to get registered.

Method 2: AUTHENTICATION PHASE ATTACKS

In this method, the malicious software on the client can affect FIDO2 authentication assuming that a malicious or vulnerable authenticator is not registered to the user's account. In this setting, the attacker is not able to compromise the authenticator, and thus can only target the FIDO2 authentication phase. At first, the case considered was where malware attempts to leverage existing legitimately registered credentials to authenticate, which is only possible with a limited set of authenticator properties. Outside of those conditions, malware cannot directly utilize existing credentials for authentication. Instead, malware must involve the user through a social engineering attack that results in the user authenticating during insecure situations. And thus, a unique social engineering attack was identified, where malware can trick users into authenticating sensitive actions without them realizing.

Scenario 1:

FIDO2 has built-in protection (KHAcessToken/authenticatorClientPIN) to prevent a user-level malware from accessing keys previously registered by a legitimate application, and thus user-level malware is not able to interfere with the FIDO2 authentication phase. In the absence of user verification at the roaming authenticator itself (e.g., button push, biometric validation), the root-level malware could successfully trigger an authentication using the user's FIDO2 credentials, without them realizing it.

Scenario 2:

In this scenario, the FIDO Client resides as a Trusted Application in the TEE, so the malware (regardless of user-level or root-level) cannot directly trigger an authentication. Instead, it must leverage a social engineering approach to cause an attacker-desired authentication, as FIDO2's workflow involves a human-in-the-loop.

In this scenario, it was found that real-world implementations of FIDO2 lack explicit user consent for a specific action, as originally recommended by FIDO. As a result, users lack clarity about what specific action they are authenticating. In addition, online services apply Risk-Backed Authentication (RBA) systems that users lack transparency into, and thus users are unable to accurately predict when they may be asked to (re-)authenticate.

When combining these two aspects, a unique social engineering attack was discovered that tricks a user into authenticating an attacker-initiated sensitive action (which the user does not realize is the action being authenticated), at the same time as when the user has initiated a non-sensitive action (but does not realize that the action does not actually require re-authentication). This was demonstrated using the example of a payment transaction via an online merchant space, where it asks for transaction confirmation, which was raised by a pawned browser extension. The extension opens merchant's WebAuthn login page in a new

background window and simulates a login button click to raise a WebAuthn authentication prompt to the user. The user provides their biometrics, authorizing the attacker-controlled session. Other than the prompt, the user never sees anything, and their only action is providing their biometrics. The attack is automated and takes only seconds to execute (e.g., time for the page load and for the user to provide biometrics).

VULNERABILITY TRIAGE

Background: Vulnerability Triage Criteria

The Vulnerability Triage Protocol is defined by the FIDO Authenticator Certification Program Policy¹.

See Table 1 below for the Triage Levels and Reasoning.

| Triage Level | Triage Reasoning |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| RED | Attack in progress, or At-scale attacks exist that can be performed with readily available tools and limited skill. |
| AMBER | Vulnerability that is likely to lead to a scalable attack. |
| GREEN | Vulnerability where attack unlikely, or not scalable. |
| WHITE | Vulnerability that is outside the scope of FIDO Specifications. |

Table 1: Vulnerability Triage Levels and Reasoning

Attack Potential Calculation

| Security Vulnerability Calculated Attack Potential (IDENTIFICATION) | | |
|---------------------------------------------------------------------|--------------|----------|
| Factor | Estimate | Value |
| Elapsed Time | <= one month | 4 |
| Expertise | Proficient | 3 |
| Knowledge of Target | Public | 0 |
| Windows of opportunity | Easy | 1 |
| Equipment | Standard | 0 |
| Calculated Attack Potential | | 8 |

| Security Vulnerability Calculated Attack Potential (EXPLOITATION) | | |
|-------------------------------------------------------------------|------------|-------|
| Factor | Estimate | Value |
| Elapsed Time | <= one day | 0 |
| Expertise | Proficient | 3 |

¹ The latest version of the FIDO Authenticator Certification Program Policy can be found at: <https://fidoalliance.org/certification/authenticator-certification-levels/>.

| | | |
|------------------------------------|----------|----------|
| Knowledge of Target | Public | 0 |
| Windows of opportunity | Easy | 1 |
| Equipment | Standard | 0 |
| Calculated Attack Potential | | 4 |

| | | |
|-------------------------------|--|-----------|
| Total Attack Potential | | 12 |
|-------------------------------|--|-----------|

The attack potential required to exploit this attack is "Basic"

⇒ TOE must be resistant against **Enhanced-Basic** Attack Potential level.

Vulnerability Triage Level

- ⇒ Protocol (no software required) = **WHITE**
- ⇒ Specific FIDO implementations of the protocol or specific platforms only = **GREEN**
- ⇒ General authenticator vulnerabilities = **WHITE**
- ⇒ Specific vendor authenticator vulnerabilities = **WHITE**
- ⇒ Specific authenticator vulnerabilities related to a specific FIDO implementation = **WHITE**

CONCLUSION

We acknowledge the concerns regarding the vulnerability in the FIDO2 ecosystem. It's important to emphasize that the core vulnerability doesn't stem from the FIDO2 Authenticator itself, but from specific implementations and potential compromises on the user's device. If malware infiltrates a user's device, it can allow an attacker to impersonate the user and gain access to an RP account. Additionally, the lack of thorough verification during authenticator registration by some RPs can potentially allow the introduction of malicious authenticators. This risk is inherent across all FIDO2-enabled devices.

The foundational assumption of the FIDO protocol is the integrity of the user's device and related applications involved in FIDO operations. Mechanisms like passkey attestation or MDS/CA verification are designed to verify the authenticity of authenticators. However, when a user's device or browser extensions are compromised, or if a RP fails to verify authenticators accurately, the FIDO Authenticators, as robust as they are, cannot singlehandedly counteract these emerging attack vectors. The threat, in essence, is primarily localized to the user's device and specific RP implementations, rather than the broader Authenticator framework.

Notably, there is a recognized disparity between FIDO's documented guidelines and real-world application, which is concerning. To bridge this gap, FIDO introduced a certification

program in 2016. This program encompasses functional conformance to specifications, interoperability testing, and a comprehensive evaluation against security requirements defined across six distinct levels of assurance.

Finally, the observation made by Dhruv Kuchhal about the FIDO Metadata Service and its management of vulnerability notifications is insightful. Indeed, our current approach manages the certification status within the MDS, which can change due to discovered vulnerabilities or other factors like product upgrades initiated by manufacturers. RPs are expected to set their own security policies based on these properties so to manage the risk adequately. This is something that we do not enforce in the certification program so there might be some rooms for discussions to make this mandatory and/or certifiable. However, it's crucial to note that FIDO doesn't actively monitor the vulnerabilities but relies on disclosures from vendors, labs, RPs, and researchers.

NOT FOR DISCLOSURE