Hardware Security Tokens With a Focus on FIDO2

Seminar: Advances in Cryptography and IT-Security

Robert Hartings

February 2, 2022
RWTH Aachen
Research Group IT-Security

Organizer: Ulrike Meyer
Supervisor: Vincent Drury

Abstract. The persistent threat situation to password-based authentication [Hun22] has led to the need for alternative authentication flows, which are preventing the most common attacks against password-based authentication. FIDO2 was proposed as such an alternative authentication flow using roaming and platform HSTs to securely store the authentication credentials. Like other authentication flows FIDO2 has disadvantages and drawbacks. In this paper, besides introducing FIDO2 in combination with HST, these problems are collected from multiple scientific sources, mainly focusing on possible phishing attacks, weak hardware security tokens, and poor user education. For each problem, a solution or an approach of improvement is presented.

1 Introduction

Despite their shortcomings when faced with common attacks such as phishing or dictionary attacks, credentials with username and password are still the most common variant of user authentication today. To create secure accounts, it is recommended to use strong passwords that contain upper and lower case letters, numbers, and special characters, with a minimum length of at least eight characters. Users also need to ensure they do not reuse a password making the memorization of multiple complex passwords necessary if a password manager is not used, though this comes with its own disadvantages like phishing and clipboard attacks [LEHY17]. The reuse of credentials makes them vulnerable to credential stuffing attacks, in which an attacker uses a stolen username and password combination hoping to exploit a victim's reuse of same or similar credentials on other services. Username and password are always vulnerable to phishing because it cannot be ruled out that even the most experienced user will make a mistake and enter their credentials on a website owned by an attacker.

This problem is challenged by the FIDO Alliance and the World Wide Web Consortium with a possible solution: The Fast Identity Online 2 (FIDO2) standard. The main difference between the proposed standard and the status quo is the paradigm shift from "something a user knows" to "something a user possesses". The FIDO2 standard includes a successor to the *universal 2nd factor* (U2F), which was also developed by the FIDO Alliance, and, in addition to the familiar second factor, also offers the possibility for *single factor authentication* (1FA), therefore making passwords redundant. The cost, inconvenience, and most users' inexperience with hardware and software security tokens are the current reasons for their low uptake [LHGU21,GLSN+20].

Similar to other authentication variants, FIDO2 has its own unsolved problems and drawbacks. In this paper, we will summarize these problems and showcase possible solutions.

2   Background

2.1   Hardware Security Tokens (HSTs)

In the context of FIDO2, hardware security tokens are called authenticators, but to the public, they are also sometimes known as security keys. In FIDO2, a distinction is made between roaming and platform authenticators. The category of roaming authenticators includes, example given, YubiKeys by Yubico, FIDO Keys by Feitian, and Titan Keys by Google. As required by the *client to authentication protocols* (CTAP) and WebAuthn standard, the evidence of user interaction is done via a touch sensor or biometric scanner (most commonly fingerprint sensors). Besides the *universal serial bus* (USB) security fobs, phones and laptops can also be used as roaming authenticators. The two largest operating systems implementing hardware security tokens on phones are the *Android Keystore* and *Apple TouchID*. The communication with the client takes place via USB, *near field communication* (NFC), or *Bluetooth low energy* (BLE). *Apple TouchID*, *Android Keystore*, and *Windows Hello* are examples of platform authenticators. The main difference between roaming and platform authenticators is that platform authenticators run on the same device as the WebAuthn client and therefore do not use cross-platform communication and CTAP [GLSN+20].

To be compliant with the standard, the user has to touch a sensor to verify their presence and unlock the secret. The security key fobs are commonly shipped with simple presence sensors, but can also be shipped with biometric sensors, most commonly fingerprint sensors. Nowadays, almost all phones ship with either fingerprint sensors, face recognition sensors, or both, and they include capabilities for pins and patterns.

Hardware security tokens are used to securely store a secret for cryptographic functions in tamper-resistant storage. The secret never leaves the secure storage and is used for deriving subsequent authentication keys for the creation of public-private key pairs. The derived keys are mainly used to sign challenges received from the relying party application, but can also be used to identify a user [PMD+21].

2.2   Fast Identity Online 2 (FIDO2)

The FIDO2 project is a joint effort by the FIDO Alliance and the World Wide Web Consortium (W3C). It is an open authentication standard succeeding prior work by the FIDO Alliance on U2F [GLSN+20]. It consists of two protocols: The WebAuthn protocol, maintained by the W3C, and the CTAPs, which are maintained by the FIDO Alliance. Members of the FIDO Alliance include Amazon, Google, Meta, and Microsoft.

The WebAuthn usage is not limited to the FIDO2 context. The protocol only specifies a JavaScript-based API used for communication between a WebAuthn relying party application (e.g. websites like google.com) and a WebAuthn client like a browser (e.g. Chrome, Firefox). The defined API enables the creation of strong, scoped, public-private key pairs. The key pairs are used as credentials for user authentication. The private key stays on the HST, while the corresponding public key is sent to the relying party. Public keys acquired through server breaches cannot be reverted to the corresponding private key or the secret stored on the authenticator. Furthermore, they cannot be used to determine private keys used for other services.

Through scoping, the standard ensures that the key pair can only be accessed by origins, a combination of protocol; domain; and port, belonging to the original relying party [UAA+21]. This scoping renders phishing useless because a relaying attacker cannot provide the origin belonging to the key pair. The API is used in two cases: (1) Registration (2) Authentication. In the case of registration, a public-private key pair is generated on behalf of the relying party on the authenticator and is subject to user consent. When the user agrees, the private key is saved on the authenticator and the public key is sent to the relying party, along with additional information like metadata of the used authenticator. In the case of authentication, the user is presented with a selection menu of accepted credentials and the origin that is requesting these keys. In both cases, user consent and scoping are enforced by adherence to WebAuthn standards User agents and the used authenticator (see Figure 1) [HJJ+21]. Nowadays, all major browsers and operating systems support WebAuthn [Aut21].

The CTAPs standardize communication between a client and a roaming authentication device. The protocols expect the authenticator to obtain evidence of user interaction. Neither the channel between the client and the roaming authenticator, nor the transport layer's encryption are currently standardized. The protocol contains the legacy CTAP1 protocol formerly known as the U2F protocol and the current CTAP2 protocol. The CTAP1 authenticators are also called U2F authenticators and the CTAP2 authenticators are also known as WebAuthn or FIDO2 authenticators [BHJ+21,LHGU21,AWAC20].

To ensure quality and security, the FIDO Alliance has set up a metadata service that can be inquired to verify the authenticator used for the login. The relying party can check whether the authenticator meets the FIDO Alliance standards and has known vulnerabilities [AWAC20].
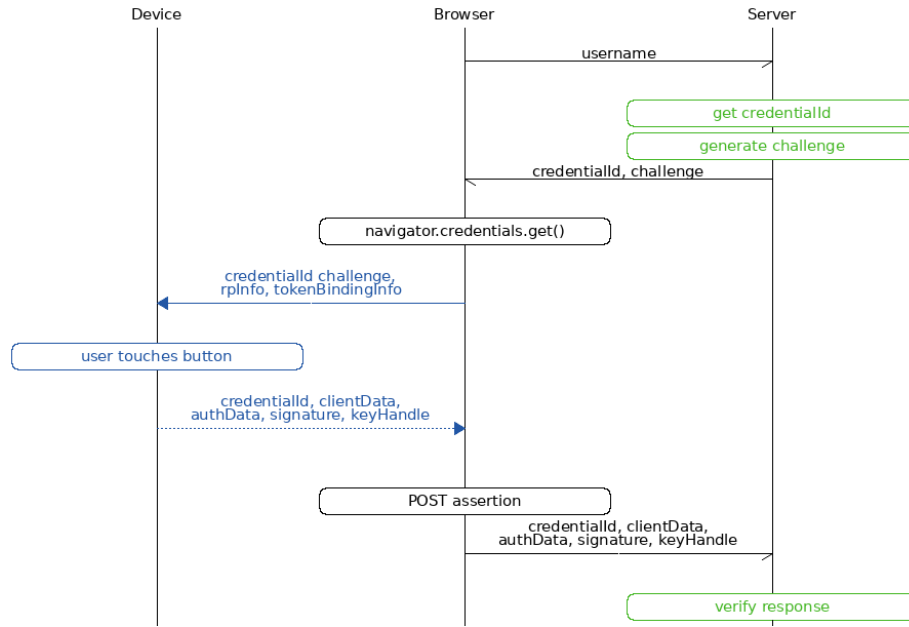
Fig. 1. FIDO2 authentication sequence diagram [Yubico, 2022]

3   Problems of FIDO

While FIDO2 has advantages over password-based two factor authentication, there are currently some disadvantages and problems that should be considered when evaluating the usability and usefulness of hardware security tokens.

3.1   Downgrade Attacks

Besides FIDO, different methods of multifactor authentication exist, like *one-time-passwords* (OTP), confirmation SMS and calls, and the usage of recovery codes [2]. Commonly, the user can choose between the configured MFA schemes, but, except for FIDO2, none of the mentioned schemes are secure against real-time phishing.

While reviewing Alexa's top 100 websites Ulqinaku et al. found out that most of these websites force users to register at least one additional *multi factor authentication* (MFA) scheme to use FIDO2 in the first place, effectively creating a vulnerability even when FIDO2 is being used because other MFA undermines the security of FIDO2. Only Google, with Google's Advanced Protection, offers a program not relying on weak MFA, but it is opt-in and not actively advertised on the *Google account dashboard*.

The downgrade attack on FIDO2, as shown by Ulqinaku et al., requires that the victim has different and weaker MFAs registered to their account. The goal of

the attacker is to display a fake login screen to the user and let the victim choose a weaker MFA scheme, which is vulnerable to real-time phishing. To achieve this goal the attacker has to set up a login page that is comparable to the website from which the attacker wants the login data or sessions. Because the user expects the security key pop-up, the attacker has to display the pop-up on the page as well. The attacker has to know if and when the user has used thier hardware security token to continue the attack. Otherwise, a user could get suspicious. The browser would usually present the pop-up above the webpage containing the domain and ask for the security key, but the WebAuthn standard also contains an API function to detect the presence of an HST without displaying this pop-up. The difference is not easy to spot for a normal user [UAA+21].

The authors of the paper suggested a solution for this problem: To disable weaker alternative schemes. They mentioned that this suggestion will come with the problem of non-scalable recovery. This recovery adds significant cost to the service provider and would not scale to millions of users, while also providing an inconvenience for the user. Registering multiple keys can be another solution but may also be costly and create a barrier. Risk-based authentication can be used to determine the authenticity of a recovery process, but recent studies [CA20] show that these can also be circumvented. Another solution suggested by the authors is to always show the hint when an application tries to access the HST, making it slightly easier for a possible victim to spot the attack.

3.2   Threats to HST

The security of FIDO2 is dependent on secure HSTs. Therefore, if the HST is compromised, FIDO2 is not secure anymore. As Pfeffer et al. pointed out, an attacker can get a hold of an access token on the supply chain between a manufacturer and the end-users, either by intercepting the delivery or inserting malicious HST as a manufacturer or a re-seller. An attacker can also buy a genuine token and return malicious HST to the seller through a refund because most sellers will not check if the HST got tampered with [Hua19]. Malicious HST can be turned into attack vectors using one or more of the following methods: (1) firmware modification (2) hardware modification. Using firmware modification, an attacker can pre-initialize a token or add malicious code and hardware which exploits e.g. USB interfaces. In the case of hardware modification, an attacker wires the HST up with a wireless transceiver, like a *global system for mobile communication* (GSM) or a Bluetooth module. Though token replicas are also counted as hardware modification. These replicas are rather easy to create as instructions are publicly available and can be used without expert knowledge. The main goal of both modifications is to extract or fix secrets, e.g. keys or seeds, of the HST so that it has deterministically derived private keys [PMD+21]. This results in the following attack scenarios.

Run-time secret extraction: The run-time secret extraction can be subdivided into in-band and out-of-band attacks. In the in-band case, the HST is modified in a way that it leaks secrets through in-protocol (covert) channels like the

signature or other channels used in the transaction. In the out-of-band case, the HST sends the secrets via a different covert channel outside of the protocol like Bluetooth or GSM.

Delivery-time secret extraction: The attacker extracts pre-configured keys or seeds through the hardware of software modifications, which allows them to deterministically guess the public-private key pairs used in future authentication.

   This described attack is only relevant to HST which are shipped with pre-configured secrets by the manufacturer like YubiKeys. In the case of most HST, and also YubiKeys, these secrets can be changed by the end-user whenever they want with software provided by the vendor.

Secret fixation: Another method that allows an attacker to deterministically guess the generated public-private key pair is by pre-loading the secret, also called "fixing" it. To permanently fix the secret key, hardware, software or both need to be modified. Also, the attacker can use the vendor's software to achieve thier goal.

Predictable RNG modifcation: The *random number generator* (RNG) used for the secure derivation of keys can be manipulated to only create predictable sequences by using hardware and software modifications. The RNG can have unintended weak design, so that no modifications are needed for exploitation. When a weak RNG is used, the attacker can guess the key pair more easily.

Ransom attack: Like other ransom attacks, this attack appears as a denial of service attack. The HST is manipulated in such a way that it stops operating after some time, demanding a ransom to resume its work with a threat to delete the stored secrets. This attack has limited viability for FIDO2, since, in most cases, recovery codes are generated upon FIDO2 key registration. This attack is more feasible for hardware wallets (outside of the scope of this paper).

USB pivoting: Furthermore, the HST can not only be used to attack authentication flows but also to attack the whole client via the USB interface. If the HST carries malware, it can act as a USB Rubber Ducky, emulating a keyboard, or trigger a buffer overflow.

   Pfeffer et al. present already existing methods to detect a tempered HST. Modification on hardware or firmware level can be detected with tamper-evident packaging using holographic stickers or other methods. But this is only a low level of protection since holographic stickers are easily replaceable and the attacker can be a manufacturer or re-seller. An HST token can be a single-piece cast, like the Yubikeys, or can be opened. The single-piece cast can be easily inspected but can be broken with household chemicals when the manufacturer foregoes more chemical-resistant plastic. Openable HST can be inspected by users increasing security by visually comparing manufacturer pictures with the present HST. This process has its downsides as it is error-prone and cumbersome. Signals on

the printed circuit board can be intercepted or manipulated, requiring a secure CPU or other element to shield it. In theory, the key never leaves this secure element and therefore cannot be intercepted. To prevent firmware manipulation, automatic and manual software verification can be used.

In the case of automatic verification, local and remote validation must be distinguished: Local Validation only validates the integrity by conducting a simple signature check. Remote validation is more sophisticated; the internal status is validated by a third party, making token replicas harder to produce. Both methods need to be visible to the user, otherwise, the user cannot make related trust decisions. Manual verification can be done by the end-user with HST supporting it, example given YubiKeys, but the corresponding software has to be found, as it is not advertised on the pages of the manufacturer. Moreover, this method is neither explained nor advertised by vendors on delivery, leaving uneducated users in the dark.

A way to prevent attacks on pre-configured secrets is to not ship them at all and let the user generate their secrets themselves. However, the authors state that manual verification and the absence of pre-configured secrets reduce the user-friendliness of HST, which can result in lower market shares [PMD+21].

3.3   Misconceptions

The study conducted by Lassak et al. [LHGU21] about misconceptions in FIDO2 Biometric WebAuthn shows that users are not yet educated enough to understand the basic functionality of FIDO2 HSTs. There are, among others, misconceptions about storage location, recovery, and usage of multiple devices. The study was held online with 42 participants from the UK and US. All of them were older than 18. The participants used their android smartphone as the HST.

Storage Location The majority of the participants thought that their biometrics were sent in an encrypted fashion to the corresponding service provider. Only 14 participants recognized that the biometrics are stored locally and only 2 recognized that the service provider could not get their biometric data, because it does not have the phone.

Likewise, only 24 participants guessed that their biometric data are not affected in the event of a database breach on the services provider's side.

Overall, only four participants were confident that their biometric data did not leave the phone when they used it for authentication.

Lost HST Because the private key used for authentication is stored on a phone, losing the phone can provide an attacker with access to the accounts if they can unlock the secret with a registered method. 39 out of the 42 participants thought that an attacker needs their biometrics for the authentication even though the secret can be unlocked with fallback mechanism like pin, pattern, or password, which are easier to obtain for the attacker.

Availability If the unlocking of the phone fails via registered biometric data, only five participants were aware that they can unlock the HST with a registered fallback option. The other participants stated that they do not have set up backup methods at the service provider or they can contact the service provider to recover their account.

Multiple Devices Also, there are misconceptions concerning device sharing. One misconception is that it is possible to use another device (after registering one's biometric data). Only six participants were aware that the public-private key pairs are tied to the authenticating device and that the biometric is only used to decrypt and unlock the private key used for the authentication process. Transferring the private key from one authenticator to another is not intended in the current WebAuthn and FIDO2 specifications. The specifications state, that if this kind of behavior is required a roaming authenticator is needed. However, some service providers also make it possible to register more than one HST for a single account, allowing multiple authenticators to grant access to the account.

Delegating Access 39 participants answered that it is not possible to grant a trusted person access since the person would not have the required biometric data. The other participants argued that it would be possible to use fallback methods or register the biometric data of the trusted person on the used HST. The latter is right, the token can be unlocked using different biometric data if they are registered on the HST.

Lassak et al. conducted another follow-up study with different participants. In this study, the participants were separated into co-design focus groups with the task to come up with notifications that address the following misconceptions. Participants received a short and basic introduction video presenting the mechanics of account creation and authentication, to ensure a general knowledge of FIDO2. Detailed information about FIDO2, especially the underlying public-private key cryptography, was intentionally left out.

In the focus groups, participants stressed the storage location of the used biometric features. Notifications created by the focus groups stated that biometric features never leave the device or that no one has access to the biometric features. The created notifications also promoted the convenience and security of biometric WebAuthn.

They also conducted a third study to review the effectiveness of ideas collected during the second study. Because the key component of the notification created during the second study was the storage location, most of the notifications created for the third study revisited this misconception. Participants received one of the following notifications during account creation: (1) Backed by brand-name, (2) cannot be hacked, (3) biometric data never leaves your device, (4) biometric data only stored on your device, (5) biometric data is never shared with third parties, or (6) a default notification without further information. This study showed that the created notification influenced the perceptions of security and privacy in a positive way.

Based on the collected data the authors stated that services using biometric WebAuthn should inform the user that the biometric data is never sent to nor stored by the website, informing them that the biometric is only used on their device. Information about biometric WebAuthn should also point out its convenience, speed, and ease of use. The authors also mentioned that the mental model of password-based sign-in is responsible for many misconceptions and that it needs more research to create notifications changing this model. They make no specific suggestions on how to overcome the discussed problems [LHGU21].

3.4   Additional Problems

Besides the mentioned problems, the FIDO2 standard has other problems which are not related to the implementation. This is highlighted in a study by Lyastani et al. [GLSN+20].

Account Recovery A fear the participants of the study had was the loss of the HST, which would have had as a consequence that they would not be able to access their accounts. Currently, other 2FA can be used to allow account access even when an HST fails. Some service provides allow for the setup of additional HST, which is recommended by the FIDO Alliance, or other 2FA schemes, which are vulnerable to e.g. phishing, as shown earlier. Only Google's Advanced Protection Program forces the user to register two HSTs.

This issue is not yet addressed by the FIDO Alliance. The current recommendation is to use additional authenticators. The authors recommend guiding the users in the task of scalable account recovery, but this will not change the fact that account recovery is a serious issue which should be handled by the FIDO instead of a single service provided to create a single way of account recovery. This problem needs attention, especially if FIDO2 should become the only 2FA scheme or even a 1FA, as it can impede future adoption [GLSN+20].

Account Suspension The authors themselves raise the concern of how to revoke the HST from an account if the HST of the user is stolen or lost and the user is not able to access the account using other MFA schemes. The FIDO Alliance argues that the risk of such thefts is lower than becoming the victim of a phishing campaign or server breach. The authors are concerned about the effect of FIDO, especially in terms of 1FA, on abusers in intimate partner violence. They are unsure if HST eases or hampers such targeted attacks. Also, the authors state that such behavior is not considered thoroughly enough yet and that users need the possibility to lock their account down without having the HST. They also mentioned that a possible inspiration can be the key revocation in PKI or GPG.

4   Adoption Barriers

Farke et al. explored the willingness of users to make use of HST in a study. Since they only had nine participants a general statement cannot be made, but

the study can provide insights into reasons that can prevent a wider appeal of HST and FIDO2. The users indicated that they feared losing the HST, the inconvenience of being required to plug in the HST, and the habit of using passwords as adoption barriers.

Convenience Three participants stated that the login with an HST requires additional effort and time, making it less convenient than the previously used credentials like username and password. It was also perceived negatively that the setup of the HST took additional time (5 - 10 minutes). This implies that even a short duration of time can reduce the willingness to use HST.

In the study, the workflow was negatively impacted, because using an HST meant that additional steps were required for each login attempt. The click count and the time to log in were mentioned. Also, it was cited that the current implementation of Windows Hello can be a hurdle for adoption because the account selection when using multiple accounts for a single service was not clear. This behavior is implementation-dependent but if a user has to search for the right account in the HST or the authentication device it can prevent the spreading of FIDO2.

The study also indicated that thought must be given to overcoming old habits and that measures must be taken to widely distribute HST.

The authors mention that the participants understand the authentication process using HST and that the authentication time was more of a limiting factor. Compared to the studies presented above, participants in this study received a training session in advance. The difference in speed could be one of the primary adoption barriers for FIDO2 and HST. Also, participants answered that unconscious decisions to use passwords instead of the HST makes it even more difficult to change the mental mindset of the users, in general, to use HST over passwords. Since only the recent version of operating systems and clients were equipped with all WebAuthn features, WebAuthn has the potential to spread better in the future. Since it no longer requires special software or specific knowledge and is thus more accessible to the general public. To overcome the mentioned adoption barriers the authors have the following suggestions: (1) Support for multiple HST (platform and roaming) (2) Requirement of adoption of HST (3) Implementation of FIDO2 in as many as possible services (4) Allow to "remember" the unlock state of HST when using PIN or biometrics [FLS+20].

5   Conclusion

FIDO2, in combination with hardware security tokens, offers a more secure way of user authentication. This authentication scheme is resistant to all currently used attacks on password flows. It even provides the possibility of single-factor authentication; retiring passwords and their problems. But as we found out, it has its problems and drawbacks. More users would be forced to use FIDO2 if it was more widely implemented on websites, including implementation as the only

2FA option. The argument of the added cost of authentication through expensive tokens can be invalidated with the fact that today around 90 percent of the population already own a mobile phone [ITU21]. In our opinion, fundamental user education is needed, but this can be done during the setup of the security tokens on the site of the service providers. More studies addressing the design and content of notification shown to end-user before or during the creation of stronger credentials, like the one conducted by Lassak et al., are needed to better understand how end-users can be recruited to use more secure authentication schemes while also preventing misconceptions from occurring. Hopefully, studies in the future can be used to create a style guide for notifications addressing security, so that the important notification can be standardized. Also, the problems presented in 3.4 need to be more thoroughly considered before FIDO2 has a more sizeable market share to avoid problems with recovery and, especially, to prevent an abuser from accessing the account of a victim.

Acknowledgments

References

[2fa22]   2factorauth. 2fa directory. https://2fa.directory/, 2022. Accessed: 2022-01-28.

[Aut21]   Auth0. Does my browser support webauthn. https://webauthn.me/browser-support, 2021. Accessed: 2021-12-22.

[AWAC20]   Fatima Alqubaisi, Ahmad Samer Wazan, Liza Ahmad, and David W Chadwick. Should we rush to implement password-less single factor fido2 based authentication? In *2020 12th Annual Undergraduate Research Conference on Applied Computing (URC)*, pages 1–6, April 2020.

[BHJ+21]   John Bradley, Jeff Hodges, Michael B. Jones, Akshay Kumar, Rolf Lindemann, and Johan Verrept. Client to authenticator protocol (ctap). https://fidoalliance.org/specs/fido-v2.1-ps-20210615/fido-client-to-authenticator-protocol-v2.1-ps-20210615.html, 2021. Accessed: 2022-01-16.

[CA20]   Michele Campobasso and Luca Allodi. Impersonation-as-a-service: Characterizing the emerging criminal infrastructure for user impersonation at scale. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1665–1680, New York, NY, USA, 2020. Association for Computing Machinery.

[FLS+20]   Florian M. Farke, Lennart Lorenz, Theodor Schnitzler, Philipp Markert, and Markus Dürmuth. "You still use the password after all" – exploring FIDO2 security keys in a small company. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 19–35. USENIX Association, August 2020.

[GLSN+20] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael
           Backes, and Sven Bugiel. Is fido2 the kingslayer of user authentication? a
           comparative usability study of fido2 passwordless authentication. In *2020
           IEEE Symposium on Security and Privacy (SP)*, pages 268–285, May 2020.
[HJJ+21]   Jeff Hodges, J.C. Jones, Michael B. Jones, Akshay Kumar, and Emil Lund-
           berg. Web authentication: An api for accessing public key credentials level
           2. https://www.w3.org/TR/2021/REC-webauthn-2-20210408/, 2021. Ac-
           cessed: 2022-01-16.
[Hua19]   Andrew Huang. Supply chain security: "if i were a nation state...".
           https://www.youtube.com/watch?v=RqQhWitJ1As, 2019.   Accessed:
           2022-01-31.
[Hun22]   Troy Hunt. Pwned websites. https://haveibeenpwned.com/PwnedWebsites,
           2022. Accessed: 2022-01-28.
[ITU21]   ITU.   Mobile   phone   ownership.   https://www.itu.int/itu-
           d/reports/statistics/2021/11/15/mobile-phone-ownership/, 2021.   Ac-
           cessed: 2022-01-28.
[LEHY17]   Carlos Luevanos, John Elizarraras, Khai Hirschi, and Jyh-haw Yeh. Anal-
           ysis on the security and use of password managers. In *2017 18th Inter-
           national Conference on Parallel and Distributed Computing, Applications
           and Technologies (PDCAT)*, pages 17–24, 2017.
[LHGU21]   Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "it's
           stored, hopefully, on an encrypted server'': Mitigating users' misconcep-
           tions about FIDO2 biometric WebAuthn. In *30th USENIX Security Sym-
           posium (USENIX Security 21)*, pages 91–108. USENIX Association, Au-
           gust 2021.
[PMD+21]   Katharina Pfeffer, Alexandra Mai, Adrian Dabrowski, Matthias Gusen-
           bauer, Philipp Schindler, Edgar Weippl, Michael Franz, and Katharina
           Krombholz. On the usability of authenticity checks for hardware security
           tokens. In *30th USENIX Security Symposium (USENIX Security 21)*, pages
           37–54. USENIX Association, August 2021.
[UAA+21]   Enis Ulqinaku, Hala Assal, AbdelRahman Abdou, Sonia Chiasson, and
           Srdjan Capkun. Is real-time phishing eliminated with FIDO? social en-
           gineering downgrade attacks against FIDO protocols. In *30th USENIX
           Security Symposium (USENIX Security 21)*, pages 3811–3828. USENIX
           Association, August 2021.