Hardware Security Tokens In Context Of FIDO2

Seminar: Advances in Cryptography and IT-Security

Robert Hartings

January 4, 2022
RWTH Aachen
Research Group IT-Security

Organizer: Ulrike Meyer
Supervisor: Vincent Drury

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

1 Introduction

Credentials with username and password are still the most common variant of user authentication today, despite their problems with phishing or dictionary attacks, for example. To create secure accounts, it is recommended to use strong passwords (including capital and lower letters, numbers and speical characters and a minimum length) while refraning from reuse. The user has to memorize these rather complex different passwords or use a password manager which has its own disadvanteages. If credentials are reused on multiple accounts, they are vulnerable to credential stuffing attacks, in which an attacker uses stolen credentials from one service on other services hoping for same or similar credentials, making it easier for him to guess the right combination of username/email and password. 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 enters thier credentials on an website owned by an attacker.

This problem is challenge by the FIDO Alliance and the W3C by providing a possible solution: The Fast Identity Online 2 (FIDO2) standard. The main difference between the proposed standard and status quo is the paradigm shift from "something a user knows" to "something a user poses". 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, making password as we know them superfluous. The cost, inconvenience and unfamiliarity of (hardware) security keys are currently reason for thier low uptake.[LHGU21][GLSN+20]

But like other authentication variants, FIDO2 has its own unsolved problems and drawbacks. In this paper I would like to summarize these problems and possible solutions.

2 Background

2.1 Hardware Secruity Tokens (HSTs)

HSTs are used to securly store a secret key used for cryptographic functions in a tamper-resistant storage. The main idea is that the secret never leaves the secure storage. The secret key is used for deriving subsequent authentication keys for creation of public / private identities. The derived keys are mainly used to sign recived challenges, but can also be used to identiy a user.[PMD+21]

HST, also called authenticators, can be so called security keys but also integrated authenticators including Trusted Platform Module (TPM), Andriod keystore and Apple TouchID. Microsoft Hello is one example for the TPM.

Security keys from vendors like Yubico (Yubikey), Feitian (FIDO Key) and Google (Titan Key) are very popluar. In the most cases a user has to touch a sensor to verify his presence. They can also be shipped with biometric scanners / sensors, most commenly finger print sensors, to unlock the private key and authentication the requested action. If the authenticator is external, communication with the device takes place via USB (Universal Serial Bus), NFC (Near Field Communication) or BLE (Bluetooth Low Energy).[GLSN+20]

2.2 Fast Identity Online 2 (FIDO2)

The Fast Identity Online 2 (FIDO2) Project is an joint effort from the FIDO Alliance and the World Wide Web Consortium (W3C). It is an open authentication standard succeeding prior work of the FIDO Alliance on Universal 2nd Factor (U2F). [GLSN+20] It consits of two protocols. The WebAuthn protocol, maintained by the W3C, and the Client to Authtioacation Protocol 2 (CTAP2), maintained by the FIDO Alliance. Members of the FIDO Alliance include Amazon, Google, Meta and Microsoft.

The WebAuthn protocol specifices a JavaScript-based API used for communication between a service provider / WebAuthn relying parties (e.g. websites) and a WebAuthn client like a browser. All major browsers support WebAuthn today.[Aut21] The CTAP2 standardize the communication between a client and the (external) authentication device.[LHGU21][AWAC20]

The main idea of FIDO2 is to use public-private cryptography instead of known credentials like username and password. Furthermore, it creates a public-private keypair unique to a given application or website, which is used to sign challenges from the service and is only generated and stored on the authenticator. This is realized through a mutal authentication using a service identificator. In case of websites the authenticator recives the domain of the requesting website. Effectivley rendering phishing useless, because a relying attacker cannot provide the authenticator with the right domain.[UAA+21] Also preventing replay attacks and password theft. Tokens acquired through server breaches cannot be reverted to the original secret key on the authenticator nor can they be used to determine private keys used for other services.

To ensure quality and security the FIDO Alliance setup a metaservice which can be inquired to verify the used authenticator. The relaying party can check if the authenticator meets the FIDO Alliance standards and has no known vulnerabilities.[AWAC20]

3 Problems of FIDO

While FIDO2 seems like a optimal solution, there are currently some disadvanteages, which cannot be ignored during the evaluation of the usability and usefullness of hardware security tokens.

3.1 Misconceptions

The study 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 different devices. The study was held online with 42 participants from the UK and US. All of them were older than 18. The used HSTs were the participant andriod smart phone.

Storage Location The majority of the participants thought that there 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 figured out that the service provider could not get there biometric data, because he is not in possession of the phone (HST).

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

Overall, only 4 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 it can provide an attacker with access to the accounts. From the 42 participants 39 thought that an attacker need their biometrics for the authentication while in fact the private key can be unlocked with fallback mecanism likes PIN, pattern or password.

Availability If the unlocking of the phone via biometric fails only five participants were aware that they can unlock the HST with fallback options like PIN, pattern or password. The other participants stated that they have not setup a backup method at the service provider or they have to contact the service provider to recover their account.

Multiple Devices / Delegating Access Also, there are missconecptions concerning device sharing. One missconecption is that is possible to use another device (after registring ones biometric data). Only six participants ware aware that the login is tied to the authenticating device and that the biometric is only used to decrypt and unlock the private key for the authentication process. Transfering the private key from one authenticator to another is not intended in the current WebAuthn specifaction. If this kind of behavior is needed a roaming authenticator is needed. Furthermore, it is possible on some services to register more than one HST for a single account.

To grant a trusted person access 39 participants ansewered that it is not possible since the person would not have the required biometic data. The other participants argued that it would be possible using fallback methods or registering the biometric data of the trusted person on the HST (phone).

Idea of resolving this problem To prevent this type of misconceptions, service providers and HST manufacturers should launch an information campaign about HSTs and how they work. If one or both of them fail to do their job or additional advertising is needed, consideration should be given to whether government institutions, such as the BSI (German Federal Office for Information Security), should launch joint / own campaigns to inform the population about better methods of securing their accounts online.

3.2 Downgrade Attacks

Besides FIDO different methods of multifactor authentication exists, like One-Time-Passwords (OTP), confirmation SMS and calls and the usage of recovery codes. Commenly the user can choose between the configured MFA schemes, but expect FIDO2 none of the mention schemes is secure against real-time phishing. While reviewing Alexa's top 100 websites Ulqinaku et al. found out that most of these websites forces users to register at least one different MFA to use FIDO2 in the first place. Effecitively creating a vulnerability even when FIDO2 is used, because it 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 advertised on the Google Account Dashboard.

A downgrade attack on FIDO2 can only be done if the user has different and weaker MFA registered to his account. The attacker can use this to ignore / skip the authentication via FIDO2. If an user visites a malicious website and tries a login to his account, the attacker can rely the response to victim. If the victim chooses to use FIDO2 the attacker simply ignores the response of the client and lets the user choose from a different MFA scheme, which is vulnerable to real-time phising.

The attacker has to know when and if the user has inserted his security key to continue with the authentication. Normaly the browser would present a box above the webpage containing the domain and asking for the security key, but it is also possible via API functions to detect the presence of a HST without displaying this box. The attacker can display such a box with the content he

likes on his website, but this is limited to the page and is not displayed above the content. For a normal user this is not easy to spot.[UAA+21]

3.3 Threats to HST

While FIDO2 seems secure it relies on a secure HST. Therefor, if the HST is compromised FIDO2 is not secure anymore. 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 an manufacturer or a re-seller. An attacker can also buy genuine token and return malicious HST to the seller on refund, because most sellers won't check if the HST got tempered with. Malicious HST result in one or more of the following attack vectors: firmware modifications, hardware modifications or secret extraction. Through the firmware modification an attacker can pre-initialize an token or add malicious code which exploits e.g. USB interfaces. In hardware modification an attacker can wire the HST up with an wireless transceiver, like GSM or Bluetooth. But also he is able to build token replicas as instructions are public available and can be used without expert knowledge. The main goal of both modification is to extract secrets, e.g. keys or seeds, from the HST. This is done most commonely via falut injections, timing side-channels and bus snooping. [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 transcation. 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 An attacker can extract pre-configured keys or seeds through the above mentioned mehtods, which allows him to determine the used keys.

This attack is only releveant to HST which are shipped with pre-configured secrets by the manufcature like YubiKeys. In case of most HST but also YubiKeys these secrets can be changed by the user whenever they want.

Secret fixation Using hardware or software modifications or both an attacker can pre-load a key to the HST, which makes the key computation deterministic.

Predictable RNG modifcation The Random Number Generator used for the secure creation of keys can be manipluliate to only create predicatable by using hardware and software modifications. In case of unintentionally weak randomness the attacker does not need to modify the token and can abuse it.

Ransom attack Like other ransom attacks, this attacks targets a denail of service. The HST is manipluliate in such a way that it stops operating after some time, demanding a ransom to resume working or release the secrets. This is attack has limited valabilty for FIDO2, since in most cases recovery codes are generated. This attack is more feasible for hardware walltes (out side of the scope of this paper).

USB pivoting On a nother node the HST can not only be used to attack logins, but also to attack the whole client via the USB interface. If the HST is equipped with malware it can act like USB Rubber Ducky (emulate a keyboard) or trigger a buffer overflow.

Pfeffer at al. present (already existing) methods to detect tempered HST. Modifcation on hardware or firmware level can be detected with tamper-evident package using holographic stickers. But this is only a low level of protection since holographic sticker are easily replaceable and the attacker can be a manufacturer or re-seller. A HST token can be single-piece cast, like the Yubikeys, or can be opened. Single-piece cast can be easily inspected but can be breakable with household chemicales when not using more chemical resistant plastic. Openable HST can be inspected by users increasing security by visually comparing manufacturer pictures with the HST. This process has its downsides as it is error-prone and cumbersome. Siginals on the printed circuit board (PCB) are intercebteable or maniplulatable needing shielding with can be done with a secure CPU or a secure element (external co-processor). The key never leaves this secure element and therefore cannot be intercept. To prevent firmware manipluliation automatic and manual software verification can be used. In case of automatic verification it will distinguished between local and remote validation. The local validation only validates the integrity by conducting a signature check. The remote validation is a more sophisticated where the internal status is validated by a third party. Both methods needs to visible to the user otherwise the user cannot make related trust decisions. The manual verification can only be done with some HST and the corresponding software has to be searched as it is not easily findable. Moreover, this method is neither explained nor advertised by vendors or service provides, letting 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 thier secrets themselve. But manual verification and dispense of pre-configured secrets reduce the userfriendliness of HST, which can result in lower markt shares. [PMD+21]

3.4 Threats to OFA FIDO2

FIDO2 allows the usage of a hardware security key as a single factor (OFA / SFA) or a multi factor (MFA).

3.5 More Problems

Account Recovery

Account Suspension

Distribution

Convenience

4 Conclusion

educate the user / population, support FIDO2 on more websites

References

[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.

[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.

[LHGU21] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "it's stored, hopefully, on an encrypted server'': Mitigating users' misconceptions about FIDO2 biometric WebAuthn. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 91–108. USENIX Association, August 2021.

[PMD+21] Katharina Pfeffer, Alexandra Mai, Adrian Dabrowski, Matthias Gusenbauer, 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 engineering downgrade attacks against FIDO protocols. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3811–3828. USENIX Association, August 2021.