CPSC 502.04: Research Project in Information Security (F21)
Supervisor: Rei Safavi-Naini <rei@ucalgary.ca>

Name: Kiley Carson

UCID: 30009983

Project Proposal

Due 09-28-2021 @ 23:59

Introduction

Today's most common form of authentication is no longer secure enough as a means of protecting online accounts, making strong security difficult, if not impossible to achieve. The standards for what is an acceptable password varies between users, and even strong passwords may be compromised by phishing attacks or identity fraud. [1]. FIDO2 is the latest specification from the Fast Identity Online (FIDO) Alliance, that aims to solve this problem. The goal is to enable convenient passwordless authentication that is secure against phishing and identity fraud. Users are able to log into their accounts using biometric or hardware tokens. Many existing devices are already equipped with the corresponding hardware/software making this a lot easier, and if not, FIDO2 tokens can be acquired. An advantage of FIDO2 is that the token can be used by multiple different web services instead of creating and remembering different passwords. Additionally, it has a smaller attack surface for cyber criminals. If a web service provider did want even more security, FIDO2 could be used with multi-factor authentication (MFA).

The Project

The goals of the project are:

- To understand the FIDO2 authentication system, its cryptographic building blocks, and its security model and proof
- Enhance security of FIDO2 by using One-Time Passwords (as a form of Multi-Factor Authentication), in particular the Google Authenticator System
- Provide a proof-of-concept implementation for the enhanced system and evaluate its performance

There is a need for security enhancement in the FIDO2 protocol. Firstly, there exists two separate "trusted setup" phases; one which occurs in the WebAuthn/PlA protocols, and the other in the CTAP2/PACA protocols. The setup procedures generate an attestation key pair for the authenticator and configure a user PIN (embed the PIN in the authenticator). Additionally, the security of the FIDO2 protocol is based on the assumption that authenticators are tamper-proof [1]. If one of the safety measures is circumvented or one of the assumptions doesn't hold, the outcome for the security of the FIDO2 protocol is catastrophic.

In the CTAP2 protocol, static storage is assumed to be initialized using a procedure carried out under special setup trust assumptions [1]. An authenticated channel is assumed for the communications between the client and the authenticator during setup as there are no "pre-established authentication parameters". In the authenticator setup phase the user embeds its PIN into the authenticator via a client (browser) and as a result, the authenticator stores a PIN-related long term state. If this is violated the authenticator could store the wrong PIN-related long term state.

In the passwordless-authentication (PlA) protocol used in the analysis of WebAuthn, the token is associated with an attestation public key that is pre-registered to the server. The "trusted setup" phase in the PlA protocol consists of the Key Generation algorithm [1]. Because of this Key Generation algorithm, the attestation key is assumed to be registered with the server. Authenticators are assumed to be tamper-proof, and the key-generation stage, where an attestation key pair is created and installed in the token, is either carried out within the token itself, or performed in a trusted context that leaks nothing about the attestation secret key. The registration subprotocol follows the key generation algorithm; here, the server requests the token to register some initial authentication parameters. If this succeeds, the server can later recognize the same token using a challenge-response protocol.

A One-Time Password could be used to strengthen the PIN-based access control for authenticators (PACA/CTAP2) protocol in FIDO2. It could also supplement or replace the user gesture predicate to add additional security guarantees to the user (i.e. that the server identity being used in the PlA session is the one that was intended). There are no passwords during user registration or authentication in FIDO2. The PIN used in FIDO2 is meant to authorize a client (browser) access to an authentication device (token), however, the server does not use passwords at all.

Previous Work

FIDO2 Security Proof

Several flaws were identified in the first provable cryptographic analysis of the authentication properties guaranteed by FIDO2 [1]. Specifically, CTAP2 cannot achieve unforgeability (UF) security because in the binding phase it uses unauthenticated Diffie-Hellman Key Exchange (DHKE) which is vulnerable to Man-in-the-Middle attacks [1]. A corruption model with security notions ranging from strong unforgeability (SUF) to unforgeability with trusted-binding (UF-t) was used to do a modular security analysis of the FIDO2 protocol [1]. The result was that the composed protocol of WebAuthn+CTAP2 (FIDO2) was able to achieve user authentication security guarantees under the weakest corruption model (UF-t) [1].

Suggestions for improvement were also identified. In CTAP2 a protocol change is suggested in the binding phase to achieve stronger security. Specifically, replacing unauthenticated DHKE with a password-authenticated key exchange (PAKE) protocol as PAKE protocols take as input a common password and output the same random session key for both parties [1]. Additionally, a proposal for a generic protocol called "secure Pinbased Access Control for Authenticators" (sPACA) was given and a proof of its strong security [1]. sPACA was also proven to be more efficient than CTAP2, and the authors "advocate[d] the adoption of their protocol as a substitute for both stronger security and better performance" [1].

Web Authentication (WebAuthn)

Together with the FIDO Alliance, The World Wide Web Consortium developed several measures for the FIDO2 project. The specification that resulted is known as Web Authentication or WebAuthn [1]. WebAuthn is a protocol that regulates the connection between the user's system and the website where the person needs to identify themselves [2]. It is a uniform authentication option that no longer relies on passwords; instead it uses biometric data or hardware tokens to identify users. Many existing devices are already equipped with the necessary hardware/software and users always carry this information with them, they can neither forget it nor pass it on without thinking. With WebAuthn, phishing could be a thing of the past.

Since users no longer need to create usernames/passwords the standard ensures that unique login information is available for each users account. Since different data is used for each account, there is no tracking across different websites [2].

The advantages of WebAuthn include convenience and ease of use as there is no need to memorize information anymore. This is great in terms of security, as the use of passwords is only conditionally secure. WebAuthn does not transmit identity data over the internet, thus, a MITM attack won't be successful [2]. All sensitive data remains on the user's device, thus, providers of services don't have to expend effort on securing username/password data. The interface is addressed via Javascript making it very easy for website operators to implement [2]. The authenticity certificate is cryptographically secured by public key encryption during transfer. WebAuthn is considered to be more secure than multi-factor-authentication (MFA).

The disadvantages of WebAuthn include situations when a new authenticator has to be registered for an existing account. An example would include when a hardware token is lost, a user will need a new token (or authenticator), and tokens aren't easy to link to existing profiles because it would be too great of a security risk [2]. A replacement authenticator, that is intended for this exact use must be held or the token authentication process must be reset. The resetting process is similar to resetting a password, and is suitable for services that do not require a high security standard [2].

CTAP2

CTAP2 is used in combination with WebAuthn, makes FIDO2 work. To ensure that only authorized individuals can log into an online account, there must be a form of authentication; in FIDO2, these are the authenticators. Communication between the authenticators (Tokens) and the user's system (PC/laptop/browser) is regulated by CTAP [3] There are other forms of authenticators that are installed directly on a laptop/PC/smartphone, however, these components do not require a separate communication protocol, as they are not external.

CTAP communication follows a specific pattern. First the browser (responsible software) connects to the authenticator/queries about information [3] The system then determines what authentication option the external device is offering and based on that, the system is then able to send a command to the authenticator. Lastly, the authenticator will send either a response or an error message if the command doesn't match the devices capabilities An advantage with the CTAP protocol is that authentication data never leaves the user's access area with this method. All sensitive data remains in the system and the browser only sends confirmation through WebAuthn that access is permitted [3].

One-Time Passwords

A one-time password (OTP) system relies on the ability for a device to generate a one-time code [4] If the code is correct, then the user is given access to the account. One-time passwords can be considered secure because the password is only good for one use. OTP generation involves 3 things that work together as a function; the OTP generation algorithm, seed, and counter value [4]. When the function is called, the counter is incremented, creating a unique counter value with each function call. It is important that any given output must never reveal any information about the seed, or what future output values will come. There are two algorithms that the Initiative for Open Authentication (OATH) endorse for use HMAC-based

One-time Password (HOTP) and Time-based One-time Password (TOTP)[4]. The Google Authenticator is an example of an offline variety of a Time-Based One-Time Password [4], [5]. It is the user's device (vs. the server) that generates the one-time codes, then both parties are expected to generate the same one-time code during a given time window. [4] For this model of OTP, a prior sharing of the secret seed between the client and the server is required.

Deliverables

The work to be done includes

- A detailed study and analysis of FIDO2 and its security.
- Design security enhancement to FIDO2 using one-time password generator as a form of MFA.
- A proof of concept implementation of the design and its evaluation.

Timeline

A timeline for this project is as follows:

- Project proposal
- Research (What can be done with One-Time "passwords", how can it strengthen FIDO)
- A design for an improved FIDO2 using One-Time Passwords
- Interim report
- Analyze System Design (Security Goals and Proofs)
- Implementation of System
- Analyze System Implementation (Comparisons of existing system, definition of limitations)
- Final report

References

[1] M. Barbosa, A. Boldyreva, S. Chen, and B. Warinschi, "Provable security analysis of fido2," in *Annual International Cryptology Conference*, Springer, 2021, pp. 125–156.

[2] (2019). "Webauthn (web authentication)," [Online]. Available: https://www.ionos.com/digitalguide/server/security/webauthn/.

[3] (2019). "Ctap: A protocol for more security & convenience on the web," [Online]. Available: https://www.ionos.com/digitalguide/server/security/client-to-authenticator-protocol-ctap/.

[4] M. L. T. Uymatiao and W. E. S. Yu, “Time-based otp authentication via secure tunnel (toast): A mobile totp scheme using tls seed exchange and encrypted offline keystore,” in *2014 4th IEEE International Conference on Information Science and Technology*, IEEE, 2014, pp. 225–229.

[5] H. Seta, T. Wati, and I. C. Kusuma, “Implement time based one time password and secure hash algorithm 1 for security of website login authentication,” in *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, IEEE, 2019, pp. 115–120.