engineering proceedings

MDPI

Proceeding Paper

An Analysis of the Current Implementations Based on the WebAuthn and FIDO Authentication Standards †

Martiño Rivera-Dourado [1,2,*], Marcos Gestal [1,2,3], Alejandro Pazos [1,2,3] and José M. Vázquez-Naya [1,2]

[1] Grupo RNASA-IMEDIR, Departamento de Ciencias de la Computación y Tecnologías de la Información, Facultade de Informática, Universidade da Coruña, Elviña Campus, 15071 A Coruña, Spain; marcos.gestal@udc.es (M.G.); alejandro.pazos@udc.es (A.P.); jose@udc.es (J.M.V.-N.)
[2] Centro de Investigación CITIC, Universidade da Coruña, Elviña Campus, 15071 A Coruña, Spain
[3] IKERDATA S.L., ZITEK, University of Basque Country UPVEHU, Rectorate Building, 48940 Leioa, Spain
* Correspondence: martino.rivera.dourado@udc.es
† Presented at the 4th XoveTIC Conference, A Coruña, Spain, 7–8 October 2021.

Abstract: During the last few years, some of the most relevant IT companies have started to develop new authentication solutions which are not vulnerable to attacks like phishing. WebAuthn and FIDO authentication standards were designed to replace or complement the *de facto* and ubiquitous authentication method: username and password. This paper performs an analysis of the current implementations of these standards while testing and comparing these solutions in a high-level analysis, drawing the context of the adoption of these new standards and their integration with the existing systems, from web applications and services to different use cases on desktop and server operating systems.

Keywords: WebAuthn; authentication; FIDO

Citation: Rivera-Dourado, M.; Gestal, M.; Pazos, A.; Vázquez-Naya, J.M. An Analysis of the Current Implementations Based on the WebAuthn and FIDO Authentication Standards. *Eng. Proc.* **2021**, *7*, 56. https://doi.org/10.3390/engproc2021007056

Academic Editors: Joaquim de Moura, Marco A. González, Javier Pereira and Manuel G. Penedo

Published: 27 October 2021

1. Introduction

Username and password is the *de facto* authentication method used in almost every web application, but it is threatened by several attacks. The most relevant one is phishing. During the last few years, some of the most relevant IT companies have started to develop new solutions which are not vulnerable to these attacks. In this context is where they form the FIDO Alliance to start developing a protocol to use hardware devices and public-key cryptography to perform authentication.

WebAuthn [1] is a new W3C authentication API for browsers to make use of hardware or software FIDO security keys [2] for replacing or complementing the username and password authentication method. Therefore, this new method can be applied in two different use cases: (1) using the security key as a second factor authentication method, usually after a password; (2) using the security key as a first factor authentication method, identifying and authenticating the user, without the need of a username or password. Moreover, web applications are not the unique systems where FIDO security keys can be of use. Operating Systems, like Windows and Linux, have solutions that make use of this new authentication method.

2. Materials and Methods

The analysis carried out in this paper has involved two main scenarios that implied two different approaches: web applications and Operating Systems. For both of them, the Solo Hacker from Solokeys, the Yubikey 5 NFC from Yubico and the Titan Security Keys from Google were used as a FIDO hardware authenticators and a PC as a host for the tests. Regarding web applications, the testers have used the Chromium browser (v.91.0) as a client and developer tool for debugging the operations, using the DebAuthn web application [3]. On the other hand, Windows 10 and Ubuntu 20.04 LTS Operating Systems

were tested inside Virtual Machines using Virtualbox, interfacing with the FIDO hardware key through USB.

3. Web Applications

As the aforementioned two use cases are different and involve specific configuration of the registration and authentication operations, the current implementations among the different existing and compatible web services is also diverse. In this paper, we analyzed and identified the different use cases two of the most relevant online platforms present in the FIDO Alliance: Google and Microsoft free accounts.

Google free accounts offer the usage of security keys as a second-factor authentication method, which they name as 2-Step Verification. As shown during the tests, the implementation from Google avoids the usage of resident credentials (a.k.a. discoverable credentials) [1], which limits their solution to use WebAuthn authenticators only as a second-factor authentication method, maintaining the password always as a first-factor. During registration, user verification trough a PIN was not required nor a user handle identifier was installed in the device. Although Google offers an Advanced Protection Program [4] which enforces the usage of a second-factor authentication mechanism with security keys, the first-factor authentication method is still based on a password. However, this implementation requires using two WebAuthn authenticators with non-resident credentials: one device for daily usage and the other as a backup in case of device loss. For this purpose, Google has developed their own Titan Security Keys, although the current version only supports non-resident credentials.

On the contrary, Microsoft free accounts implement WebAuthn only as a first-factor authentication option in their Advanced security options, excluding it from the list of second-factor authentication methods. However, Microsoft also implements other first-factor authentication methods, like push notifications to a smartphone application, SMS codes, Windows Hello or even sending a code via email.

When registering or authenticating with a WebAuthn authenticator as a first-factor, Microsoft requires the usage of resident credentials and user verification via PIN. During the registration operation, the credential with the user handle identifier is installed in the device and, during the authentication operation, this identifier is returned together within the authenticator response. It is worth mentioning that, when registering the Solokey device in the Microsoft account, the server aborts the operation. Microsoft cancels the registration when a specific FIDO authenticator is not in their list of allowed manufacturers, filtering them during the attestation verification process. For this reason, the Yubikey authenticator was used instead.

4. Operating Systems

The FIDO CTAP standard can be used to communicate with FIDO authenticators natively and defines their behaviour and available operations, so it can be used in other online and offline systems. In this context, Yubico has developed a PAM (Pluggable Authentication Module) [5] for using FIDO authenticators as a token to authenticate users on Linux-based Operating Systems. It includes a binary to obtain key handles and public keys from the authenticator, allowing to create an entry in the configuration file that maps an user with a credential.

Regarding the Windows Operating System, Microsoft has developed their own Window's native WebAuthn API, for which Yubico has recently added support in their libfido2 library [6]. The problem of this approach is that developers are not able to interact with FIDO devices natively, so FIDO CTAP2 extensions which are not included in the Windows API will not be used. In this context, we have tested different configurations of WebAuthn requests on the browser, concluding all of them in Windows launching their native platform for the interaction with the FIDO devices. This approach diverges from the solution in Linux systems, where browsers and PAM modules are in charge of performing the FIDO CTAP communication.

Although Windows has included an utility for managing security keys in their sign-in options, it does not yet support native sign-in with FIDO security keys for local accounts. However, Microsoft offers a business solution for FIDO2 authentication with security keys through their Azure Active Directory Multi-Factor feature, using Kerberos tickets to authorize users with on-premise Active Directory controllers [7]. For this reason, Yubico has developed their Yubico Login [8] solution that allows Windows sign-in with Yubikeys, although they not use FIDO CTAP2 features, so they are not compatible other security keys. This implementation uses Yubico HMAC challenge-response programmable slots available in Yubikey 4 and 5.

5. Conclusions

WebAuthn has been implemented as an authentication option in some of the most relevant web services, like Google and Microsoft free accounts. While Google has developed their own security keys to be used as a second-factor, Microsoft has chosen WebAuthn as a first-factor authentication method with resident credentials in devices of their allowed list of manufacturers. This makes the implementation from Google more conservative, as it uses WebAuthn as a second-factor, making their solution more compatible with browsers, platforms and FIDO devices. In contrast, Microsoft allows users to avoid passwords with WebAuthn, as they have been doing with other first-factor sign-in options like push notifications.

Operating Systems have started to support WebAuthn and FIDO standards for other authentication mechanisms, further than web applications. For this reason, Yubico developed local OS authentication solutions both for Linux and Windows. However, while the Linux PAM module can be used with any authenticator compatible with WebAuthn, the solution for Windows is only available for Yubikeys. Finally, Microsoft native implementations make difficult to some developers to completely use FIDO functionalities, and make their Azure AD software the only option for using any FIDO device as a sign-in option in their Operating System.

Author Contributions: Conceptualization, M.R.-D.; Methodology, M.R.-D., M.G. and J.M.V.-N.; Testing, M.R.-D.; Investigation, M.R.-D., M.G., A.P. and J.M.V.-N.; Resources, M.G., A.P. and J.M.V.-N.; Writing—original draft preparation, M.R.-D.; Writing—review and editing, M.R.-D., M.G., A.P. and J.M.V.-N.; Supervision, M.G., A.P. and J.M.V.-N. All authors have read and agreed to the published version of the manuscript.

Funding: CITIC, as Research Center accredited by Galician University System, is funded by "Consellería de Cultura, Educación e Universidade from Xunta de Galicia", supported in an 80% through ERDF, ERDF Operational Programme Galicia 2014–2020, and the remaining 20% by "Secretaría Xeral de Universidades" (Grant ED431G 2019/01). This project was also supported by the "Consellería de Cultura, Educación e Ordenación Universitaria" via the Consolidation and Structuring of Competitive Research Units—Competitive Reference Groups (ED431C 2018/49).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Web Authentication: An API for Accessing Public Key Credentials Level 2. Available online: https://www.w3.org/TR/webauthn-2/ (accessed on 18 May 2021).
2. Client to Authenticator Protocol (CTAP). Available online: https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fido-client-to-authenticator-protocol-v2.0-ps-20190130.html (accessed on 18 May 2021).
3. Debauthn: WebAuthn Authenticator Debugging Tool. Available online: https://debauthn.tic.udc.es/ (accessed on 19 July 2021).
4. Advanced Protection Program. Available online: https://landing.google.com/advancedprotection/ (accessed on 19 July 2021).
5. Pluggable Authentication Module (PAM) for U2F. Available online: https://github.com/Yubico/pam-u2f (accessed on 29 June 2020).
6. Pull Request #336 · Yubico/libfido2. Available online: https://github.com/Yubico/libfido2/pull/336 (accessed on 19 July 2021).
7. Azure Active Directory Passwordless Sign-In | Microsoft Docs. Available online: https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-authentication-passwordless (accessed on 19 July 2021).
8. Computer Login Security with YubiKey | Yubicoand . Available online: https://www.yubico.com/products/computer-login-tools/ (accessed on 19 July 2021).