proceedings

Proceedings

Implementing a Web Application for W3C WebAuthn Protocol Testing [†]

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

[1] Grupo RNASA-IMEDIR, Departamento de Computación, Facultade de Informática, Universidade da Coruña, Elviña, 15071 A Coruña, Spain; marcos.gestal@udc.es (M.G.); jose.manuel.vazquez.naya@udc.es (J.M.V.-N.)
[2] Centro de Investigación CITIC, Universidade da Coruña, Elviña, 15071 A Coruña, Spain
[*] Correspondence: martino.rivera.dourado@udc.es

[†] Presented at the 3rd XoveTIC Conference, A Coruña, Spain, 8–9 October 2020.

Published: 18 August 2020

Abstract: During the last few years, the FIDO Alliance and the W3C have been working on a new standard called WebAuthn that aims to substitute the obsolete password as an authentication method by using physical security keys instead. Due to its recent design, the standard is still changing and so are the needs for protocol testing. This research has driven the development of a web application that supports the standard and gives extensive information to the user. This tool can be used by WebAuthn developers and researchers, helping them to debug concrete use cases with no need for an *ad hoc* implementation.

Keywords: WebAuthn; authentication; testing

1. Introduction

Authentication is one of the most critical parts of an application. It is a security service that aims to guarantee the authenticity of an identity. This can be done by using several security mechanisms but currently, without a doubt, the most common is the username and password method. Although this method is easy for a user to conceptually understand, it constitutes many security problems. Some of them are password stealing through phishing and leaked password hash cracking by using dictionary attacks [1]. User passwords can also be compromised at the client side by the use of malware or hardware devices. For this purpose, there exist some techniques that capture the password on user input, often involving keyloggers [2]. All these attacks leave the user unprotected and without control over their credentials.

One of the approaches that big technology companies such as Facebook or Google started to implement is based on the use of hardware authenticators [3], such as the Yubikey [4] or the Titan Security Key [5]. This idea evolved into a web authentication protocol, called WebAuthn, which was published in March 2019 as a W3C Recommendation [6] as a result of some collaborations with the FIDO Alliance. Aside from the authenticator, the Relying Party, an "entity whose web application utilizes the Web Authentication API to register and authenticate users [6], must support the validation of the authenticator device responses.

Nowadays, although the standard has become a W3C Recommendation last year, the consortium is already building a second version, currently as a Working Draft, that corrects, improves and adds functionalities with respect to its first version. Therefore, there is an important need for testing the protocol. In many cases, a test environment can be useful, avoiding the need of an *ad hoc* implementation for reproducing a use case. For this reason, this research drove to the implementation of a web tool that allows developers and testers to debug browsers and authenticators compatible with the standard.

2. Tool Development

The developed tool displays the performed operations together with all necessary configuration details, as well as showing the researcher all the information exchanged through the web browser between the authenticator and the Relying Party. In order to provide a flexible test environment, the tool does not require a user account for its use. Built as Single Page Application and by using asynchronous requests to a web server, it allows to temporarily register credentials that can afterwards be authenticated for testing purposes. The server holds the minimum information required for the operations validation and can be accessed and deleted at the user interface. Moreover, the checks for browser compatibility and all errors occurred at the browser during any operation will be displayed at the interface. This approach gives the researcher both the server and the client data during the whole operation in a single web application.

The tool offers two main operations that follow the schema described in the Figure 1: registration and authentication. Once the user selects the operation, the web application will request the options for the selected WebAuthn operation, called Attestation for registration and Assertion for authentication (steps 0–1). In both cases, they include a challenge, which is a binary buffer generated and stored at the server. Once these default configurations are received, they can be edited by the user through a reactive form, allowing the researcher to configure settings such as requesting registration with resident credentials or changing the cryptographic algorithms. After this edition, the tool will interface with the authenticator through the WebAuthn API implemented in the compatible browser, sending these configurations to obtain the authenticator response after the user's physical interaction (step 2).

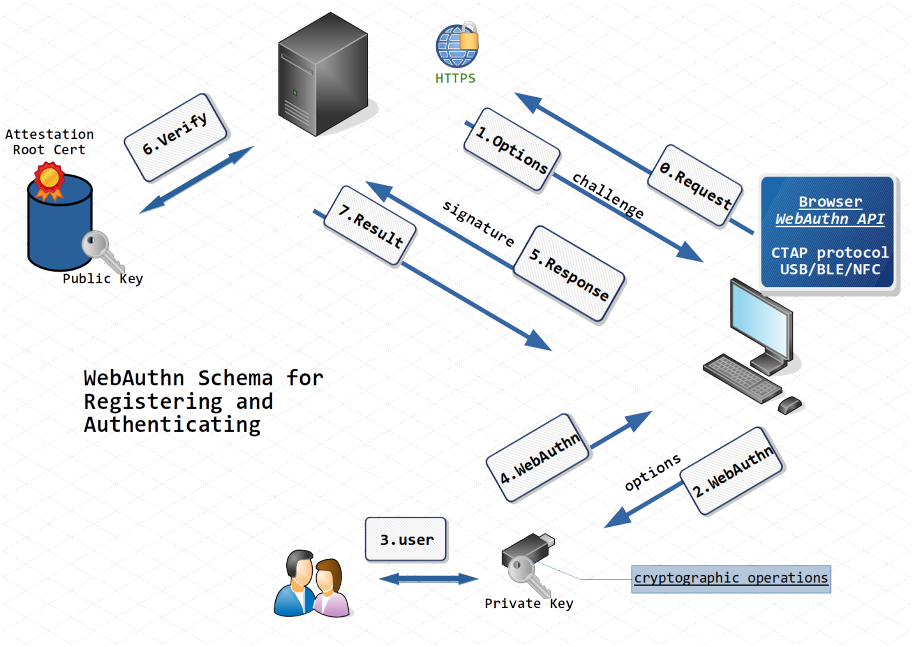

Figure 1. WebAuthn schema used by the developed tool for registration and authentication operations.

The authenticator response (step 4) is represented in some binary buffers. Like the challenge, these are encoded in Base64Url and sent to the server for performing validation (steps 5–6). During registration, once the Attestation is validated, the public key included in the authenticator response is stored at the server. On the other hand, during authentication, the public key of a registered credential will be used to validate the signature. In both operations, the result is forwarded to the web application together with all technical details, including the parsed authenticator response (step 7).

3. Results

DebAuthn (available at debauthn.tic.udc.es) is a WebAuthn debugging tool. It is a Single Page Application built with VueJS that stores temporary information tight to the session. The dashboard of the application displays the registered credentials—their counters, public keys and ids. Also, the user can delete all credentials in order to clean the workspace.

By using REST endpoints with NodeJS, the web application can send and request the necessary information, while displaying it to the user. The displayed data is divided in three steps corresponding to the main schema of the protocol: sending the options, authenticator response and server validation.

4. Conclusions

The developed tool can help both protocol researchers and developers. During the development of the new WebAuhn version, there are many contributions where a testing tool may be of use to reproduce a specific protocol use case. Furthermore, developers designing an implementation of the protocol in their systems may find the application as a useful tool for compatibility checks on authenticators and browsers, guiding their implementation decisions beforehand.

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

Funding: This work was supported by the Consolidation and Structuring of Competitive Research Units—Competitive Reference Groups (ED431C 2018/49), funded by the Ministry of Education, University and Vocational Training of the Xunta de Galicia endowed with EU FEDER funds.

Acknowledgments: We thank CITIC Research Center for hosting the developed tool.

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

Abbreviations

The following abbreviations are used in this manuscript:

FIDO	Fast Identity Online
W3C	World Wide Web Consortium
API	Application Programming Interface

References

1. ScatteredSecrets.com. How to Crack Billions of Passwords? Available online: https://medium.com/@ScatteredSecrets/how-to-crack-billions-of-passwords-6773af298172 (accessed on 18 April 2020).
2. Ahmed, Y.A.; Maarof, M.A.; Hassan, F.M.; Abshir, M.M. Survey of Keylogger Technologies. *IJCST* **2014**, *5*, 25–31.
3. Krebs, B. Google: Security Keys Neutralized Employee Phishing. Available online: https://krebsonsecurity.com/2018/07/google-security-keys-neutralized-employee-phishing/ (accessed on 18 April 2020).
4. Discover YubiKeys—Strong Two-Factor Authentication for Secure Login. Available online: https://www.yubico.com/products/ (accessed on 18 April 2020).
5. Titan Security Key Fido U2 F Usb C Nfc Ble. Available online: https://cloud.google.com/titan-security-key/ (accessed on 18 April 2020).
6. W3C. Web Authentication: An API for Accessing Public Key Credentials Level 1. Available online: https://www.w3.org/TR/webauthn/ (accessed on 18 April 2020).