"If I could do this, I feel anyone could:"
The Design and Evaluation of a Two-Factor Authentication Manager

Garrett Smith
Brigham Young University

Tarun K Yadav
Brigham Young University

Jonathan Dutson
Microsoft

Scott Ruoti
The University of Tennessee

Kent Seamons
Brigham Young University

Abstract

Two-factor authentication (2FA) is a strong defense against account compromise. However, usability studies reveal challenges with 2FA setup. The process to manually setup and remove 2FA methods differs across websites. We present a system design for a 2FA manager to automatically setup and remove 2FA methods. Potential benefits are reduced time, fewer mistakes, consistent terminology, a single workflow for users to learn, and the ability to rapidly transition to a new 2FA method—e.g., when replacing a lost 2FA method. We create two proof-of-concept implementations of our design, one as a browser extension and one integrated as a feature in an existing password manager. We evaluated the browser extension implementation approach using a between-subjects user study (N=60). Our results show fewer mistakes and reduced time compared to manually adding and removing 2FA methods. Qualitative results show that users found the automated process easy to use and were enthusiastic about the 2FA manager's ability to help them rapidly replace 2FA methods in the case they lost their 2FA device.

1 Introduction

Password authentication is vulnerable to remote attackers. Two-factor authentication (2FA) addresses this threat by requiring that in addition to a password ("something you know") users must also authenticate using a factor that is difficult to steal remotely: "something you have" or "something you are". While 2FA does not entirely prevent remote attack compromise, it does reduce the likelihood of such an attack and mitigate the impact of a successful attack [6, 7, 11].

Although 2FA provides security benefits, it is difficult for users to set up correctly [1, 3, 10]. This difficulty is cuased by: (1) A wide variety of "something you have" implementations (e.g., hardware security tokens, codes sent over SMS, and phone applications using push notifications) that are different to setup and use [9]. (2) Hundreds of websites implementing 2FA [4], with different setup interface, workflow, and terminology [10]. (3) No support to mass-enroll or remove on multiple accounts, users must adopt 2FA manually, one account at a time.

We propose a 2FA manager that provides a unified, automated process to setup and remove 2FA methods on users' accounts. To achieve our vision, we make the following contributions.

1. We analyzed the workflow of 2FA enrollment at a number of websites to identify an abstract design of the 2FA setup process. We propose a 2FA manager that partially automates the 2FA setup process, providing a unified, fast, and easy-to-use method for setting up 2FA methods across a wide range of websites. The manager is especially well-suited to help users set up accounts en masse, such as when they first begin using 2FA or acquire a new 2FA device.
2. We describe two implementation architectures of the design: including a standard web-API to simplify, streamline, and speed up the setup and removal process. We discuss lessons learned from two prototype implementations of our 2FA manager as a Chrome Browser extension and an implementation that integrates with KeePass, an open source password manager.
3. Our results show that the tested manager results in fewer mistakes and reduced time compared to manually setting up and removing 2FA methods. Qualitative results show that users found the automated process easy to use and were enthusiastic about the 2FA manager's ability to help

USENIX Symposium on Usable Privacy and Security (SOUPS) 2022.
August 7–9, 2022, Boston, MA, United States.

them rapidly replace 2FA methods in the case they lost their 2FA device.

To evaluate our proposed design, we conducted a between-subject user study (N=60) of a simulated prototype of our design to answer the following research questions:

RQ1 Does automated 2FA setup/removal increase success rate?

RQ2 Does a user's prior 2FA experience increase the success rate for 2FA setup/removal?

RQ3 Does automated 2FA setup/removal reduce completion time?

RQ4 Does a user's prior 2FA experience reduce completion time for 2FA setup/removal?

RQ5 Does automated 2FA setup/removal increase the perceived usability of the setup/removal process?

RQ6 Does a user's prior 2FA experience increase perceived usability of an automated 2FA setup/removal process?

2 Design

To improve the usability and scalability of 2FA, we propose a 2FA manager for managing the setup process. Previous studies and our meta-analysis of some of the most common 2FA methods in use today helped us determine suitable tasks for a 2FA manager. Individual websites implement different terminology, instructions, requirements, and inconsistent 2FA setting's location. We aim to provide a consistent and quick 2FA setup and removal experience into a single interface. A centralized place for setup/removal could avoid confusion caused by the wide variety of inconsistent 2FA setup processes websites offer today.

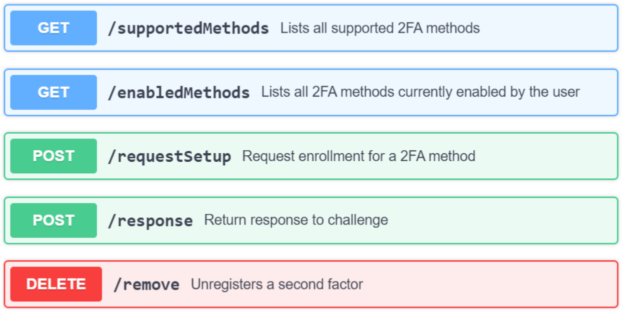
2.1 Client-Side

We identified four steps to automating 2FA setup:

1. User authentication to an account: The manager facilitates user authentication to the website for which the user wants to set up 2FA.
2. Selection of a 2FA method: The manager prompts the user to select a 2FA method from all the 2FA methods supported by the website and notifies the website.
3. The transfer of a *2FA identifier* between the user and website: The manager transfers a unique 2FA identifier between the user and the website. A 2FA identifier is a unique data-representation of a second-factor.
4. A *challenge-response* exchange to prove possession of the identifier: The manager initiates a challenge-response process to verify that a user possesses the second factor.

2.2 Server-Side

We describe two proposals for simplifying the interface between a service provider's authentication flow and a 2FA manager. The first is to introduce new standard HTML5 elements for 2FA, which are easier to support in the short term. The second is a standard API for 2FA assistants which could take longer to adopt but is more robust.



GET /supportedMethods Lists all supported 2FA methods

GET /enabledMethods Lists all 2FA methods currently enabled by the user

POST /requestSetup Request enrollment for a 2FA method

POST /response Return response to challenge

DELETE /remove Unregisters a second factor

Figure 1: API endpoints

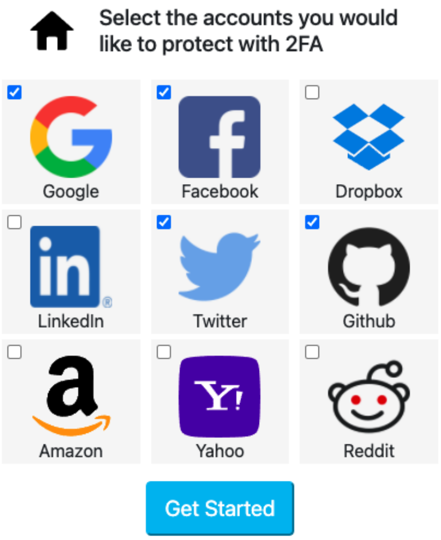
Web Standards Currently HTML5 offers standard elements to define different user input fields required for authentication, including password and email [8]. Using these declared types allows password managers to automate some authentication processes such as auto-filling user's account credentials [5]. We propose extending these input fields to include other authentication fields such as OTP code fields and QR codes for receiving the private keys for time-based one-time password (TOTP). Just as email and password fields have standard types, so should 2FA-specific fields. Websites that use these standard types would allow automated tools such as our 2FA manager to interface directly with them to simplify and automate much of the setup process.

2FA Setup API Our second proposed approach is a 2FA Setup API that standardizes the entire 2FA setup process and enables a simple interface for automation tools. A standard API supports more scalable, robust 2FA automation because it eliminates customizing a script for each website.

We developed a proof-of-concept Web API based on the four steps discussed in section 2.1. The API is designed to be used by authenticated users (or 2FA managers), and facilitates the selection of a second-factor method, the transfer of the 2FA identifier, and the challenge-response exchange. Our API is designed to be relatively simple to implement but robust enough to support many second factors. Authenticated 2FA managers can use five endpoints to manage their 2FA methods (see Figure 1).

2.3 Proof-of-Concept Implementations

We created two proof-of-concept 2FA managers to test the feasibility and usability of our design. These were built without the standard elements and API described in Section



Select the accounts you would like to protect with 2FA

☑ Google ☑ Facebook □ Dropbox

□ LinkedIn ☑ Twitter ☑ Github

□ Amazon □ Yahoo □ Reddit

Figure 2: Select accounts to enable 2FA

2.2. One manager is a KeePass password manager extension, and the other is a browser extension.

The first step in our 2FA manager setup process is to select the account(s) where a new 2FA method will be set up. The menu shows all of the websites that the 2FA manager currently supports as seen in Figure 2. After selecting a website, the user is then prompted to authenticate unless the 2FA manager has access to the users credentials. The interface then allows the user to select one of the 2FA methods supported by the website. Depending on the 2FA method selected, the 2FA manager will then follow the method specific flow. For example, when a user selects TOTP-based 2FA the extension prompts the user to scan a QR code with an authenticator app and then enter the code provided from the app. On successful completion of the second factor setup, the 2FA manager notifies the user, who can then return to the 2FA management menu.

For the remote user study, we used a chrome extension that simulated the setup and removal process with the same UX and timing delays as our complete working prototype.

3 Methodology

We conducted a 60-person IRB-approved between-subjects study to evaluate the usability of our 2FA manager compared to the current 2FA setup/removal process.

Study Design Our user study investigated how the method used to setup/remove 2FA affected the task success rate, completion time, and perceived usability of the setup/removal process. In this study we investigated two methods for 2FA setup and removal, manually using individual websites setup processes or using our simulated 2FA manager. We divided the 2FA usage into two groups, participants that used secure 2FA methods like TOTP or Security Keys, and participants that didn't. To test these variables we designed three different study groups:

	Method Used	Prior Usage
Group A	Manual	Non-TOTP Users
Group B	Simulation	Non-TOTP Users
Group C	Simulation	TOTP Users

Table 1: User Study Conditions

Study Setup Participants were led through the study by a web-based Qualtrics survey and a study coordinator. Participants were given unique login credentials for the Google, Facebook, and Dropbox test accounts and asked to log in. Study coordinators then read a short description of 2FA and TOTP with an explanation of the security benefits that 2FA provides. The coordinators then described the first task that the participants were assigned.

The setup and removal questionnaire both used the System Usability Scale (SUS) to measure participant sentiment about the usability of the setup processes as a whole. The questionnaire included questions to measure intention to adopt 2FA and perceived usefulness adapted from work by Colnago et al. [2]. Included with these questions were open-ended questions to investigate what aspects of the setup and removal process participants liked or disliked as well as reasons why they would or wouldn't use TOTP or the 2FA manager for their own accounts.

4 Quantitative Results

Task Success Rate—RQ1, RQ2 See Table 2 for a summary of the setup task success rate by Group. We hypothesized that our manager would have a significantly higher setup success rate than the manual setup. We found a statistically significant difference in the success rate between the manual method and the 2FA manager ($p = 0.0084$), thus we can reject the null hypothesis. We also hypothesized that participants with experience using TOTP would also have a higher success rate when using the manager compared to participants who used our manager but didn't use TOTP for their own accounts. For this hypothesis we fail to reject the null ($p = 0.93$) since we could not detect a significant difference in the success rate between Groups B and C.

For the removal task, in Group A only one participant failed to remove 2FA, failing only on the removal process for Dropbox. Groups B and C did not have any task failure.

Group	Success	Failure
Group A - Manual Setup	15 (75%)	5(25%)
Group B - Automated Setup	20 (100%)	0 (0%)
Group C - Automated Setup	18 (90%)	2 (10%)

Table 2: Success Rate by Group

Group	Setup Time (s)
Group A - Manual Setup	472
Group B - Automated Setup	315
Group C - Automated Setup	262

Table 3: Setup Completion Time by Group

We were unable to detect any significant difference between Groups A and B ($p = 0.13$).

Completion Time—RQ3, RQ4 Using the recorded video of each participant, we timed how long it took for each participant to complete the setup and removal tasks.

We hypothesized that the manager would significantly reduce the amount of time required to setup TOTP on the three accounts. We rejected the null hypothesis and found a significant difference ($t(33) = 2.6602, p = 0.006$) between Groups A and B. We calculated a Cohen's d of 0.904, indicating a large effect. We could not detect a significant difference between Groups B and C ($t(36) = 1.0694, p = 0.15$). See Table 3 for the timing data.

We found a significant difference ($t(32) = 3.5054, p = 0.0007$) in the mean removal time between Groups A and B, so we can reject the null hypothesis. We did not find a significant difference between Groups B and C ($t(36) = 0.506, p = 0.31$). These results are shown in Table 4.

System Usability Scale—RQ5, RQ6 The 2FA manager had a median SUS score of 77.5, while the manual method was 70.38. We included all participants regardless of whether they successfully completed the task in calculating the mean SUS score for each Group. The scores suggest the 2FA setup process can be improved through automation, however we found that the differences in ratings between Groups A and B ($t(38) = 1.1339, p = 0.132$) and Groups B and C ($t(38) = 0.1866, p = 0.57$) were not statistically significant. In both cases we can't reject the null hypotheses.

Group	Removal Time (s)
Group A - Manual Removal	162
Group B - Automated Removal	62.4
Group C - Automated Removal	59.67

Table 4: TOTP Removal Completion Time by Group

Participants that failed to setup TOTP on any of the accounts did not attempt the removal task and were not asked the removal SUS questionnaire. We can reject the null hypothesis ($t(37) = 2.7986, p = 0.0041$) for the scores between Group A and Group B with a Cohen's d of 1.02. There was no significant difference detected between Groups B and C ($t(36) = -0.6560, p = 0.74$).

5 Qualitative Results

Setup We first asked participants what they liked and disliked about the setup process. 41 participants (68%) expressed that the setup process was easy and took little effort, regardless of which setup process they used.

Discoverability of 2FA settings: In Group A, 9 participants (45%) mentioned that they disliked searching for the correct page to set up 2FA. Our prototype resolved the concern of discovering 2FA by including 2FA settings as part of the automation process. None of the users from Group B or C reported an issue regarding the discoverability of 2FA settings.

Inconsistency: One thing that some participants disliked across all systems was the inconsistency in the requirements for each account. Specifically, Dropbox and Facebook only required one 2FA method if the user wanted to set up TOTP while Google required participants to enable SMS first.

2FA Manager Five participants (25%) in Group C mentioned the convenience of using the manager to set up 2FA on multiple accounts at a time. 35% of participants groups B and C mentioned the usefulness of the 2FA manager while setting up 2FA.

Others: Ten percent of participants from group A reported issues with the instructions but no participants from group B or C reported issues regarding instructions.

Removal In general, participants (Group A - 11 55%, Group B - 18 90%, Group C 16 80%) also indicated how easy the removal process was. Participants in Group C again recognized that the 2FA manager could be helpful when managing 2FA for multiple accounts. Five participants (25%) mentioned liking removing 2FA en masse from accounts.

6 Conclusion

Our 2FA manager results in fewer mistakes and reduced time compared to manually setting up and removing 2FA methods. Qualitative results show that users found the automated process easy to use and were enthusiastic about the 2FA manager's ability to help them rapidly replace 2FA methods in the case they lost their 2FA device. In the future, we plan to further explore removal, scenarios for migrating to new devices, and how to attack the system.

References

[1] Claudia Ziegler Acemyan, Philip Kortum, Jeffrey Xiong, and Dan S. Wallach. 2fa might be secure, but it's not usable: A summative usability assessment of google's two-factor authentication (2fa) methods. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 62(1):1141–1145, 2018.

[2] Jessica Colnago, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Lorrie Cranor, and Nicolas Christin. "It's not actually that horrible": Exploring adoption of two-factor authentication at a university. In *Conference on Human Factors in Computing Systems*, page 456. ACM, 2018.

[3] Sanchari Das, Andrew Dingman, and L Jean Camp. Why johnny doesn't use two factor a two-phase usability study of the fido u2f security key. In *International Conference on Financial Cryptography and Data Security*, pages 160–179. Springer, 2018.

[4] Josh Davis. Two factor auth list. https://twofactorauth.org/, 2019.

[5] Nicolas Huaman, Sabrina Amft, Marten Oltrogge, Yasemin Acar, and Sascha Fahl. They would do better if they worked together: The case of interaction problems between password managers and websites. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1367–1381. IEEE, 2021.

[6] Troy Hunt. The only secure password is the one you can't remember. https://www.troyhunt.com/only-secure-password-is-one-you-cant/, Mar 2011.

[7] Troy Hunt. Passwords evolved: Authentication guidance for the modern era. https://www.troyhunt.com/passwords-evolved-authentication\-guidance-for-the-modern-era/, Jul 2017.

[8] Mozilla. <input type="password"> - HTML: HyperText Markup Language | MDN. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/password, 2022.

[9] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS)*, 2019.

[10] Joshua Reynolds, Trevor Smith, Ken Reese, Luke Dickinson, Scott Ruoti, and Kent Seamons. A tale of two studies: The best and worst of YubiKey usability. In *Symposium on Security and Privacy (SP)*, pages 872–888. IEEE, 2018.

[11] Alex Weinert and Lee Walker. Breaking password dependencies: Challenges in the final mile at Microsoft. *RSA Conference*, 2020.