horcrux-manager: Securing Password Management with Secret Sharing and Autofill Defenses

Samuel Havron
University of Virginia
havron@virginia.edu

***Abstract*—The abstract. Work-In-Progress Note: not all citations are proper yet or included in references! See the bib file for most references.**

I. Introduction and Motivation

some references to be used: [7] [3] [5] [2] [1] [4] [8] [6]

The web is full of mixed advice on creating and managing passwords for consumers' online accounts (Gaw, 2006), and users often ignore frustrating, burdensome security advice for rational reasons (Herley, 2009). Coupled with the risk that recent major password database leaks pose to users' security across websites with reused passwords (Roberts, 2016; LLV, 2016), there are many practical problems of secure password creation and management by users that have yet to be solved.

Users are often asked to remember many different passwords for their online accounts, with an estimated 43-51% of Internet users reusing the same password across multiple sites due to password fatigue (Das, 2014). Commercial password managers (PMs) such as LastPass, Dashlane, and 1Password attempt to reduce this burden by only asking the user to remember one master password, which serves as the key to an encrypted file containing all of the user's online account passwords, which are typically generated randomly by the password manager itself.

Password Managers (PMs), while introducing new security problems of their own, offer a solution to password fatigue for users and promise to eliminate the practice of insecure, reusable passwords across different websites. However, Ion et. al [3] found that many general users did not consider password managers as an important, or even trusted security practice to stay safe online: "password managers were regarded with skepticism by non-experts, who instead preferred to remember passwords, partly because, as one participant said, 'no one can hack my mind."'. Yet the same study found that computer security "experts" reported using a password manager as one of the most important practices a user can do to maintain online safety.

Many cloud-based PMs rely on client-side encryption/decryption and other strong security practices to protect their users' data against autofill vulnerabilities (such as XSS and CSRF attacks) and server side defenses (often with hosting on Amazon Web Services). However, recent compromises to major password managers (Khandelwal, 2016; Titcomb, 2015; and Goodlin, 2015) pose concerns for users who may begin to lose trust in their PM provider, or perhaps already have. In this paper, we describe a new password management system with stronger security guarantees for cloud-based PMs on both the client- (autofill policies) and server- (portability/storage policies) side.

We introduce 'horcrux-manager' (HM), a new password manager which does not rely on trusting any single cloud provider with the users' passwords, nor does it trust any website's form submission to properly protect the user from autofill vulnerabilities. "horcrux-manager" is developed as a Firefox web extension, and continues to provide numerous features of a modern password manager that users have come to expect, in addition to its security improvements.

II. Threat Model and Assumptions

The threat model for HM is an adversary who is assumed to steal encrypted credentials from a variety of databases belonging to the user (less than a threshold value t as described in section 3), and can execute numerous autofill attacks such as XSS/CSRF against a given domain a user is visiting. We assume that any credentials stolen could be unencrypted, and attempted to recombine with the proposed secret-sharing scheme.

On the part of the user, we expect careful selection and variety in database keystores chosen for their credentials, as well as use of a strong master password. If the adversary can obtain or brute-force the users' local token and their master password, the HM will undoubtedly be compromised. However, attacks on database credentials and autofill vulnerabilities are much more difficult for an adversary to compromise.

Most commercial password managers rely on Amazon Web Services or company servers to store encrypted user credentials. We assume that those who do not are not using secret sharing as part of their proprietary software.

III. Protecting Consumers from Cloud Storage Theft

HM is unique in that it does not trust any one database store to keep a user's encrypted passwords. Instead, HM splits passwords across multiple databases from an arbitrary amount of cloud providers using "horcruxes"; horcruxes are pieces ("shares") of each of the user's passwords for a given website, split into shares using Shamir's Secret Sharing [6]. HM distributes n shares ("horcruxes") of a given credential

across n servers specified by the end-user (e.g. AWS, Azure, personal server), requiring a security parameter of exactly $\{k \mid 0 < k \leq n\}$ horcruxes to reconstruct the credential when requested (a request requires knowing a master password and possessing a local token).

Using a smaller threshold could help improve the speed of retrieval, as well as allow the user to still reconstruct passwords in the event of any keystore being compromised or taken offline.

IV. Protecting Consumers from Autofill Vulnerabilities

The proposed password manager also includes a means to prevent many common autofill vulnerabilities from occurring while still offering autofill capability for users, by using an "auror". The auror is a network traffic analysis tool that runs as Javascript in the browser as part of the PM extension; upon visiting a website with a login form, the auror checks whether or not the website relies on client-side encryption before sending the user's credentials. The auror then populates the form fields with "dummy" credentials and waits for the user to submit the form, replacing the dummy credentials with the real username and password once the connection between the browser and the website server is secured over TLS, and JavaScript cannot modify or learn the password as it is sent out over the network. Many autofill vulnerabilities, such as those described in (Li, Z., 2014, and Silver, 2014) can be avoided by implementing the "auror" and not allowing the user's real password to be sent across network traffic until the last possible moment, as many autofill (generally XSS and CSRF) attacks will only gain the useless dummy credentials.

V. Implementation and Tests

A. Firefox Extension

Code for 'horcrux-manager' (HM) is available at https://git.io/hcx-mgr. Compared to popular commercial offerings, what HM lacks in aesthetics and to some extent, usability, it more than makes up for in protecting consumers from the major security and privacy problems which LastPass, Dashlane, 1Password, and others leave at risk.

B. Tests

See the 'tests' directory of our code for output logs. We tested HM on several popular websites for functionality using keystores in Amazon Web Services, as well as for successfully evading an XSS attack outlined by (source). Password reconstruction from t shares was tested ...

VI. Future Work

There are a number of improvements that can be made to HM in order to add to its usability and overall security. Adding support for keystores on Microsoft Azure and Google Cloud Platform would provide a stronger variety of sources for a consumer to trust when distributing their credentials. Actual testing of commercial password managers and HM for vulnerabilities as in [7] would further validate the project's viability as a usable tool for consumers.

VII. Conclusion

The conclusion.

References

[1] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *NDSS*, volume 14, pages 23–26, 2014.

[2] D. Goodin. Hacking tool swipes encrypted credentials from password manager. http://arstechnica.com/security/2015/11/hacking-tool-swipes-encrypted-credentials-from-password-manager/. 2015-11-2.

[3] Iulia Ion, Rob Reeder, and Sunny Consolvo. ... no one can hack my mind: Comparing expert and non-expert security practices. In *Eleventh Symposium on Usable Privacy and Security (SOUPS 2015)*, pages 327–346, 2015.

[4] S. Khandelwal. Lastpass bug lets hackers steal all your passwords. http://thehackernews.com/2016/07/lastpass-password-manager.html. 2016-7-27.

[5] Zhiwei Li, Warren He, Devdatta Akhawe, and Dawn Song. The emperors new password manager: Security analysis of web-based password managers. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 465–479, 2014.

[6] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[7] David Silver, Suman Jana, Eric Chen, Collin Jackson, and Dan Boneh. Password managers: Attacks and defenses. In *23rd USENIX Security Symposium (USENIX Security '14)*, 2014.

[8] J. Titcomb. Password manager 1password criticised for leaking users bookmarks. http://www.telegraph.co.uk/technology/internet-security/11939920/Password-manager-1Password-criticised-for-leaking-users-bookmarks.html. 2015-10-19.