# A Hybrid Password Manager

Alan H. Karp
alanhkarp@gmail.com

## Abstract

Most password managers store passwords so you don't have to remember them. That choice imposes costs on the company providing the password manager and risks on its users. Another option is to calculate passwords as they are needed. SitePassword is such a calculator that provides all the key features that users want in their password manager – remembering metadata, synchronizing across machines, and autofilling the login form. In addition, it can store passwords of a user's choosing.

### 1. Introduction

"Dealing with passwords is awful" is a statement few would disagree with. Password managers make the situation less awful, improving both security and usability by making it easier for you to have a different, strong password for every site. Of course, users care at least as much about ease of use, trust in the the password manager, and the ability to get their passwords from any machine as they do the strength and uniqueness of their passwords.

There are two kinds of password managers. The vast majority of them remember passwords and other metadata, such as userids. They make the stored passwords available from any machine by using encrypted databases and cloud storage.

Remembering passwords adds cost for the company providing the password manager for it to manage user accounts and to pay for the necessary cloud storage. It imposes risks on the user because the stored passwords can be stolen. It has also happened that companies have dropped support for their password databases,[1] leaving users in the lurch.

Password calculators, the second type, don't need to store passwords. They combine a master password with other data to calculate a strong password for a website when you need to login.

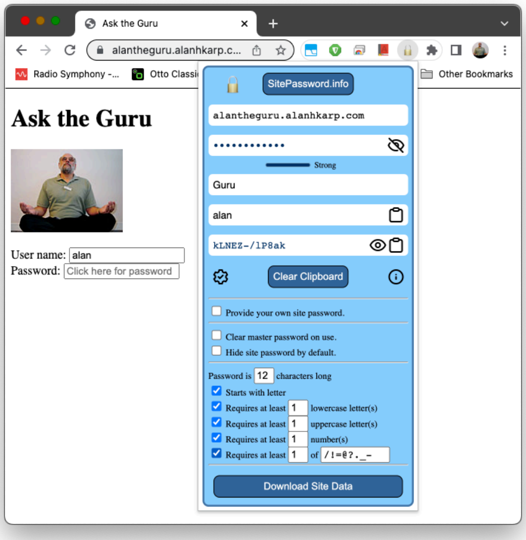
SitePassword is a password calculator designed for usability and security that supports all the features users want in a password manager. SitePassword can also store passwords the user has created making it a hybrid of the two types.

[1] https://www.remembear.com/

## 2. SitePassword Overview

SitePassword is an extension that runs in Chromium browsers.[2] It has three components. The popup, with the user interface shown in Figure 1, is where the user provides the necessary metadata. A content script running on the page with the login form is responsible for finding the userid and password fields on the page and filling them in on request. A service worker manages the metadata and calculates the site password when the content script asks for it.

SitePassword remembers your settings and, if desired a user selected password, for each site and synchronizes across your machines without the need for it to manage user accounts or cloud storage. It does that by putting the metadata in bookmarks, which virtually all browsers synchronize across machines. Using bookmarks makes the settings available on machines that cannot install the extension, such as mobile devices. The security aspects of this choice are covered in Section 5.



Ask the Guru

User name: alan
Password: Click here for password

alantheguru.alanhkarp.com
••••••••••••
Guru
alan
kLNEZ-/1P8ak

□ Provide your own site password.

□ Clear master password on use.
□ Hide site password by default.

Password is 12 characters long
☑ Starts with letter
☑ Requires at least 1 lowercase letter(s)
☑ Requires at least 1 uppercase letter(s)
☑ Requires at least 1 number(s)
☑ Requires at least 1 of /!=@?._-

Figure 1: The SitePassword popup showing all available options.

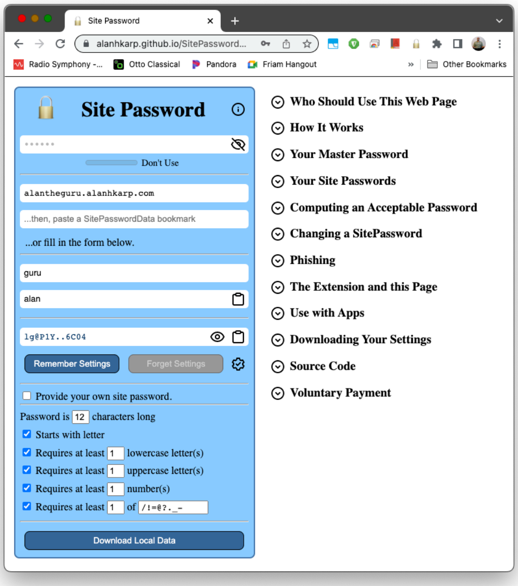
Unlike most password calculators, SitePassword provides features usually found only in password managers that store passwords. In addition to synchronizing across machines, SitePassword finds and autofills both userid and password fields and provides a web page for use on machines where the extension is not installed. The Appendix compares SitePassword to other password calculators in the Chrome Store.

## 3. Using SitePassword

When the user first encounters a page with a login form at a given domain, the password field contains a placeholder *Click SitePassword*. Clicking the SitePassword icon opens the popup. As you fill out the form the site password field updates on every keystroke, making it clear how uncorrelated the passwords are.

As you mouse over to the password field, your userid gets filled in and the placeholder changes to say *Click here for password*. Click and your password gets filled in. When returning to that login page on any machine that synchronizes

[2] Brave, Chrome, Edge, Opera, and Vivaldi. Firefox is not a pure Chromium Browser.



Site Password

••••••

alantheguru.alanhkarp.com

...then, paste a SitePasswordData bookmark

...or fill in the form below.

guru

alan

1g@P1Y..6C04

□ Provide your own site password.

Password is 12 characters long

☑ Starts with letter

☑ Requires at least 1 lowercase letter(s)

☑ Requires at least 1 uppercase letter(s)

☑ Requires at least 1 number(s)

☑ Requires at least 1 of /!=@?._-

Figure 2: The SitePassword web page showing all available options and the table of contents of the instructions. Note that the calculated password is the same as in Figure 1.

your bookmarks and has the extension installed, your userid is automatically filled in, and you only need to click on the password field. The result is that using SitePassword is even easier than typing the same, weak password for every site.

Although it is not recommended, SitePassword allows you to specify a password you create for a site. Simply fill in the form, check the Provide Password check box, and edit the site password field. Your password XORed with the computed site password is stored in the bookmark for the domain. Once created in this way, SitePassword will insert it into the password field.

SitePassword also provides a web page, shown in Figure 2, that can be used to get your passwords when the extension is not available, such as on your mobile devices. You don't even have to remember your settings if you synchronize bookmarks to the device. You can paste the corresponding bookmark into the form.

You can also download a nicely formatted table of your settings that you can print for times when your bookmarks are not available. Losing this piece of paper adds only a small amount of vulnerability as discussed in Section 5.

## 4. Usability Considerations

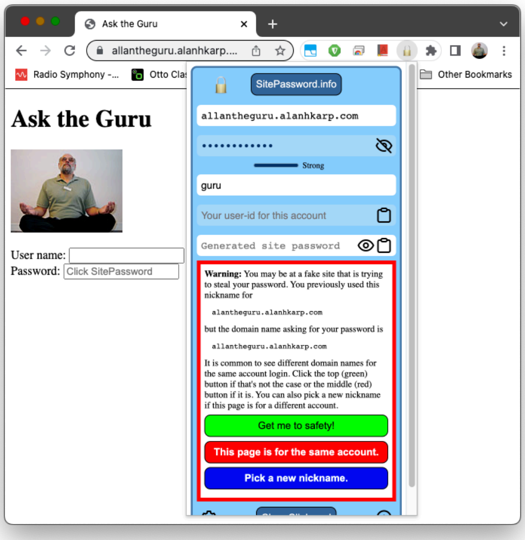
People who don't use a password manager often give usability as the main reason [1]. SitePassword was designed to be as usable as a simple game. Each step should be obvious; you should be able to experiment without worrying about breaking something; warnings make clear what next steps to take. Although SitePassword comes with extensive instructions, as shown in Figure 2, the hope is that they won't be needed.

A very simple way to guide the user is to put the browser focus on the field that should be filled in next. Another is to disable fields until they are ready for user interaction. For example, a user provided password is XORed with the computed site password, which can only be done once the site and user names are available. The Provide Password check box is disabled until those fields are filled in.

Another way to help the user is to put the information the user needs where the user is looking. An example is the placeholder put in the password field of the login form. More extensive information is provided by tooltips, which also serve as a backup in case the placeholder isn't visible.

Websites often have rather specific password rules, such as how many lower/upper case letters, numbers, and special case characters are needed and even which ones are acceptable. The user's only involvement in getting a valid site password is in filling out the form in the popup to specify the rules. SitePassword automatically finds a suitable password by continually hashing until one is found, stopping after 1,000 tries. While failing to find a legal password could happen by chance, the most frequent cause is incompatible settings, such as asking for more numbers than the specified length of the password. If that's not the case, trying again with a different nickname for the site will find a valid password with high probability. No such failure has occurred in a decade's use of this algorithm.

There are times when you need to type the site password manually. The most common case is logging into an app on a smart TV. It is also possible that SitePassword won't find the password field on a page that has disabled pasting. Hence, SitePassword uses a font that doesn't make, for example, lower case l (ell) look like upper case I (eye).[3]



Ask the Guru

User name:
Password: Click SitePassword

allantheguru.alanhkarp.com

guru

Your user-id for this account

Generated site password

**Warning:** You may be at a fake site that is trying to steal your password. You previously used this nickname for

alantheguru.alanhkarp.com

but the domain name asking for your password is

allantheguru.alanhkarp.com

It is common to see different domain names for the same account login. Click the top (green) button if that's not the case or the middle (red) button if it is. You can also pick a new nickname if this page is for a different account.

Figure 3: The SitePassword phishing warning.

Phishing is a threat that SitePassword is in a position to warn about, but too many websites make it ambiguous by using multiple domain names for logging into the same account. Unfortunately, it is virtually impossible to use the domain name to distinguish phishing from a legitimate use of a different domain name. For example, are the domain names in Figures1 and 3 for the same account? You can't tell from the trailing part of the domain name because alanhkarp.com might be providing these domain names as a service to independent parties. Perhaps allan is attempting to phish users of alan's website.

[3] An earlier version did not follow this advice, resulting in the author needing several tries to log into Netflix on his new smart TV.

The fact that most phishing warnings are spurious leaves two choices. Either ignore them or explain them. SitePassword made the latter choice in the hope that the explanation shown in Figure 3 suffices.

## 5. Security Considerations

A guiding principle is that a system cannot be secure if users are not aware of the implications of their actions. To that end, the SitePassword user experience is based on a set of usable security principles [2]. For one, SitePassword makes the easy way the secure way, as noted in Section 3. It also requires a click on the password field both to avoid a family of attacks [3, p. 37] and to give the user control. Leaving a password on the clipboard is not good practice, so SitePassword indicates when that might be the case. Finally, potential phishing results in a warning as shown in Figure 3.

Of course, that's not enough. It's also necessary to keep the user informed. To that end, SitePassword provides a strength meter for the master password and uses the same colors for the site password in case a generated one is weak.

Other protections aren't visible to the user. The user's master password is never stored. It is kept in session memory, a transient space only available to the extension for the duration of the browser session. The callback that fills in the password is only registered on visible password elements, making clickjacking more difficult. If that element is in an iframe, the domain name of that iframe is used to select the metadata used to calculate the site password. Using the userid and site's nickname in the computation provides salt to protect the master password from pre-image attacks.

Some decisions slightly weaken security to give a big gain in usability. For example, the site password is visible by default to increase the user's trust in the generated passwords.[4] While users have the option to change the default, any risk is mitigated by the fact that you only open the popup to enter your master password at the start of a browser session, to set up a new password, or when you need to copy-and-paste your site password. Another decision was choosing a default site password length of 12 because some websites do not accept longer ones, although that has become less of an issue in recent years.

The main decision that weakens security is using unencrypted bookmarks to store your settings. The benefits are huge in that there's no need for user accounts or cloud storage, the settings are available on your mobile devices that can't install the extension, and support for personas is provided by browser profiles. The risk is low; all an attacker gains is knowledge of the sites you log into and your userids at those sites. That's not nothing, but know-

[4] Besides, it's cool watching it change as you type.

ing that gives no hint of either your master or site passwords.

There are two reasons not to encrypt. First, the user selects to the bookmark by domain name when pasting it into the web form, which means a key piece of data must be in the clear. Second, is the question of what to use for the encryption key. The master password may not work, because the user might have different master passwords for different classes of web sites. It wasn't deemed worth the complication of dealing with that just to protect your userids, often your name or email address, which are easy to guess.

There are also mitigating factors. It is hard to get to your bookmarks from a web page, so malware on your machine is the main threat. Such malware is likely to cause bigger problems than exposing your SitePassword settings.

The biggest threat you face is an offline attack against your master password.[5] Say that you create an account at a rogue site. The site owner knows your userid and site password and can probably guess your nickname for the site. The site owner can now mount an offline guessing attack against your master password.

There are three mitigations. First is choosing a strong master password. According to zxcvbn [4], which is used for the password strength meter, it would take 30 days to guess a 12-character password at $10^6$ guesses per second. The second is that SitePassword hashes a minimum of 100 times before producing a site password, increasing the attackers work factor by that amount. The third is up to the user. Nothing says you must have only one master password. You could have one for banking and health accounts, a second for subscriptions, and a third for sketchy sites. This choice is not readily available for password managers that store your passwords.

User provided passwords are XORed with the password calculated for the site before storage with the other settings in the bookmark for the domain. As a result, an attacker who learns the password for a site can retrieve the computed password but would have to mount a guessing attack to learn the master password.

## 6. Related Work

There are a number of reviews [3] of the kind of password manager that stores passwords, so this section only reviews password calculators.

An early very simple calculator [5] hashed the user's master password with a user selected nickname for the site to produce a password that the user would copy and paste into the password field. It was later adapted into the HP Antiphishing Toolbar for Internet Explorer, which added many of the features of a modern password manager.

[5] This same attack is possible if a password database gets stolen, something that is out of the user's control when the password manager stores the database in its cloud.

PwdHash [6] combines the user's password with the domain name of the page. Its key advantage is familiarity; the user simply types into the login form's password field. SitePassword trades this familiarity for the simplicity of just clicking on the password field.

The PwdHash algorithm means that you get different passwords if the site uses different domain names for the same account, a problem SitePassword solves by associating the settings with a user chosen nickname for the account. PwdHash's greatest strength is the protection it affords when typing passwords into pages with malicious JavaScript, which is not a problem for SitePassword because the user only types into the popup.

The Appendix has notes on all the password managers in the Chrome store as of March 2023 that provide at least one password manager feature.

### 7. Future Work

There is still work to be done. A user study would identify any glitches in the user experience. A comprehensive security review of both the design and the code should also be done. Unfortunately, there is no funding to support either of them. In lieu of a user study, SitePassword is only available to those who know its URL in the Chrome Store. The hope is that early users will provide sufficient feedback. Informal discussions with a number of security experts has confirmed the validity of several aspects of the design.

Work has started on porting SitePassword to both Firefox and Safari, but there is no plan to create apps for mobile devices because of existing security problems on those platforms [3, Chapter 4].

## 8. Conclusions

SitePassword is a hybrid password manager that includes the features people want most – ease of use, trustworthiness, remembering of metadata, and availability of passwords across devices. Because it doesn't require user accounts or cloud storage, it is free. It is also open source, allowing a company to produce a version with corporate branding

## Acknowledgements

The author would like to thank Dale Schumacher for providing a professional look and feel to the user interface, significant code improvements to the web version, and generally useful discussions. Douglas Crockford provided significant feedback as an early adopter, and Jasvir Nagra provided guidance needed to make the content script work properly. Useful input came from the Friam security group.

### References

1. Sarah Pearman and Shikun Aerin Zhang and Lujo Bauer and Nicolas Christin and Lorrie Faith Cranor, Why people (don't) use password managers effectively, Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019), pp 319–338, 2019

2. Yee, Ka-Ping, User Interaction Design for Secure Systems, Proceedings of the 4th International Conference on Information and Communications Security, pp. 278–290, http://zesty.ca/pubs/icics-2002-uidss.pdf, 2002
3. Oesch, Timothy, An Analysis of Modern Password Manager Security and Usage on Desktop and Mobile Devices, Ph.D. Thesis, University of Tennessee, https://trace.tennessee.edu/utk_graddiss/6670/ 2021
4. Wheeler, Daniel Lowe, Zxcvbn: Low-Budget Password Strength Estimation, Proceedings of the 25th USENIX Conference on Security Symposium, pp. 157–173, 2016
5. Karp, Alan H., ”Site-Specific Passwords”, HPL-2002-39, https://www.hpl.hp.com/techreports/2002/HPL-2002-39R1.pdf, 2002
6. Blake Ross and Collin Jackson and Nick Miyake and Dan Boneh and John C Mitchell, Stronger Password Authentication Using Browser Extensions, 14th USENIX Security Symposium (USENIX Security 05), 2005

## Appendix

There are a LOT of password managers in the Chrome extension store. The majority store passwords, but there are also a lot that calculate them. I installed all of the password calculator extensions that claim to have at least one password manager feature and attempted to use each one to figure out what features it supports. Some of them I couldn't figure out how to make work; others didn't work at all. Some have web pages for doing the calculation that no longer exist. A few have instructions not to use them!

The good news is that all of these password managers that autofill require a user action. The bad news is that user action is often in the popup. That risks putting the password in unintended fields, such as the answers to security questions or those there due to a clickjacking attempt.

A surprising number are unusable, in contrast to those that store passwords. For example, some that use the domain name as an input generate different passwords for different domains for the same account, *eg*, client.schwab.com vs. www.schwab.com. Others have no option to generate passwords if the extension is not available or to put the password on the clipboard if autofill doesn't work. Most do not provide rich enough settings to guarantee generating a legal password.

The table summarizes properties of the all password calculators in the Chrome store that work and have at least one password manager feature. The column headings are:

**Extension**: The name of the extension to search in the Chrome store
**Inputs**: The values used to compute the password, settings include such things as the number of numbers, upper and lower case letters, and how many and which special characters to include.
**M**: Which values are remembered across invocations

**S**: If settings are synched across machines

**F**: Autofill (u = userid, p = password)

**R**: If there is a way to conform to the site's password rules (P = partial)

**O**: If you can get your passwords on other machines if the extension isn't available

**P**: If you are warned of potential phishing

**A**: If you need an account

**D**: If there is an app for mobile devices

**E**: If you can use your existing passwords

**Notes**: Things that don't fit the other categories.

| Extension | Inputs | M | S | F | R | O | P | A | D | E | Notes |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **SitePassword** | masterpw<br>userid<br>nickname<br>settings | All settings and user provided passwords | Y | u<br>p | Y | Y | Y | N | N | Y | Finds pw field on 120+ test sites, |
| **MindYourPass** | pw per site<br>part of domain, pw size, special chars | username | Y | u<br>p | P | Y | N | Y | Y | Y | Autofill often falls |
| **OnePass** | nickname part of domain, pw size | masterpw<br>nickname | N | N | P | P | N | N | N | N | Recommends using any MD5 generator for O |
| **Password Generator** | nickname<br>settings | all | Y | p | Y | Y | N | N | N | N | Autofill fails on iframe, uses sync storage |
| **GenPass** | masterpw<br>domain<br>pw size | masterpw<br>nickname | Y | N | N | N | N | N | N | N | Different passwords for client.schwab and www.schwab |
| **Password Maker** | masterpw<br>nickname<br>settings | masterpw | N | ? | P | N | N | N | N | N | Autofill doesn’t work with iframe |
| **Pure** | masterpw<br>nickname<br>grid | None | N | p | P | N | N | N | N | N | Only opens if it finds password field, fails on iframe |
| **Entropass** | masterpw<br>nickname<br>hidden key<br>pw size | nickname | Y | p | P | N | N | N | N | N | Can’t tell how hidden key is synched |
| **SynthPass** | masterpw<br>part of domain, pw size | All | Y | u<br>p | N | N | N | N | Y | N | Doesn’t work if pw in iframe |
| **PwdHash** | password<br>domain | None | N | na | N | Y | N | N | N | Y | Type into pw field with prefix |
| **PawHash** | masterpw<br>nickname<br>settings | None | N | p | P | N | N | N | N | N | Doesn’t work if pw in iframe |
| **OfflinePass** | masterpw<br>domain<br>year, counter | masterpw<br>userid | N | N | N | Y | N | N | N | N | Different pw for client.schwab and www.schwab, |
| **SimplePasswords** | masterpw<br>nickname | masterpw | N | p | N | N | N | N | N | N | Autofills on generating password |
| **Just1Password** | masterpw<br>domain<br>email | email | N | N | N | N | N | N | N | N | Different pw for client.schwab and www.schwab, |