SitePassword

A Hybrid Password Manager

Alan H. Karp alanhkarp@gmail.com 12 April 2023

I’ve attended a bunch of these talks. To be completely honest this topic is the least significant of any of them. That said, it is the one you encounter the most often.

———————

History

15 March 2023 - local ACM Chapter meeting
12 April 2023 - Stanford EE380
25 October 2030 - Stanford Security Lunch

“Passwords are terrible, and we won’t be using them in 10 years.”

I expect the first part of this quote sums up how many of you feel. Do you know who said it?

“Passwords are terrible, and we won’t be using them in 10 years.”

Everybody

Do you know when it was said?

“Passwords are terrible, and we won’t be using them in 10 years.”

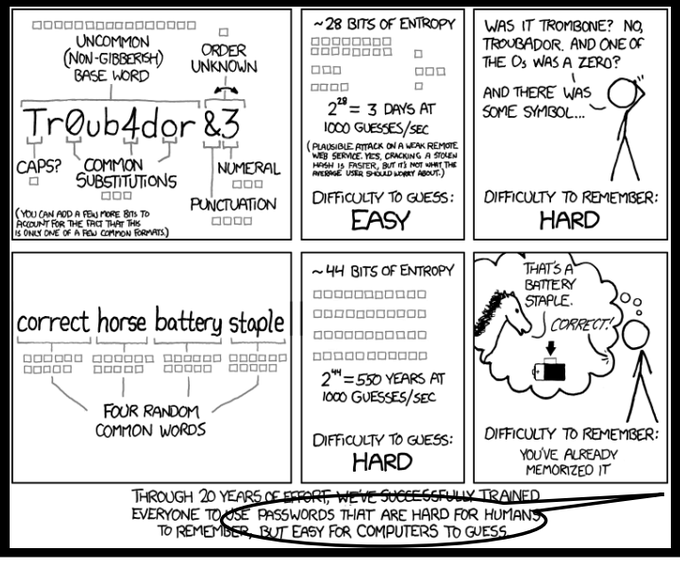
Everybody

2013 2003 1993 1983 1973 1963

Every year since we’ve been using passwords.

In other words, we’re likely stuck with passwords for quite a while.

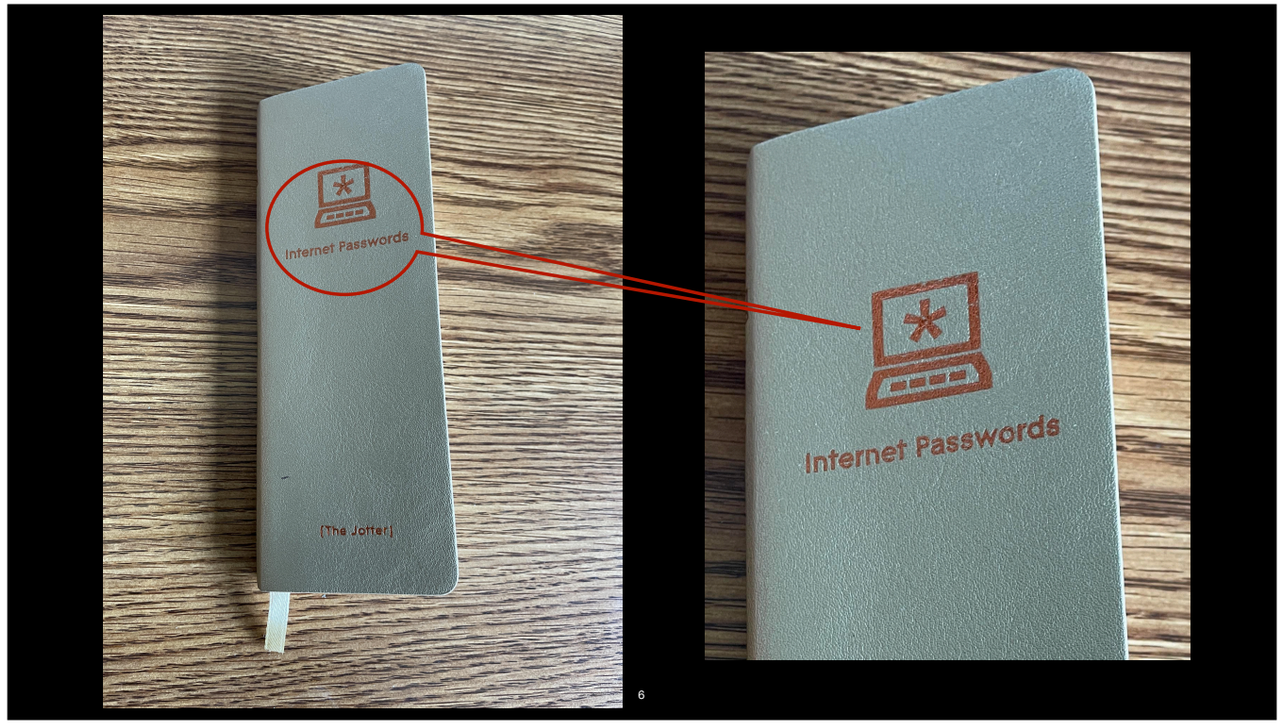
What's So Bad about Passwords?



Passwords are hard for people to remember and easy for computers to guess.

And as usual xkcd explains the problem with passwords best. Passwords are hard for people to remember and easy for computers to guess.

What do we do in such a situation?



One solution is to write them down. I’m surprised there’s actually a market for this product!

Do You Use a Password Manager?

One of the top recommended security practices but only 40% use them

A far better solution is to use a password manager, but fewer than half of people use one.

I'd like to get a count of this group. I would have set up a Zoom poll, but I quit reading the instructions after page 3. Instead, just put the name of your password manager in the chat and "None" if you don't use one. Also, tell me if you're happy with your password manager.

For those of you who put None, I have just one thing to say, "Shame on you."

Raise your hand if you use a password manager. Shame on you that don't. Now lower your hand if you're happy with your password manager. Those of you with your hands up, I'd be interested in knowing what you don't like.

You can lower your hands now.

Do You Use a Password Manager?

One of the top recommended security practices but only 40% use them

- 1Password ($36)
- LastPass ($36)
- DashLane ($60)
- Keeper ($35)
- ZohoVault (Free or $54)
- Avira ($32)
- RememBear (Shutting down)
- PassBolt (Free if you host)
- Bitwarden (Free or $40)
- LogMeOnce ($48)
- NordPass (Free or $36)
- PasswordBoss (Free?)
- RoboForm ($24)
- Your Browser (Free)

There are plenty to choose from, and this list is just a sample.

Some are free, others cost a few $$ a month, but beware. Sometimes they shut down.

Password Manager Characteristics

- Requirements
  - Easy to use
  - Engenders trust
  - Use from anywhere
  - Secure
- Other desirable features
  - Strong passwords
  - Different for every site
  - Easy to change (Ask me why that's bad)

We expect certain things from a password manager.

If it isn't easy to use, if it doesn't give you the warm fuzzies, if you can't get your Netflix password when you're at a friend's house, you won't use it. Of course, if it gets hacked, you'll stop using it.

Difficulty using it and lack of trust are the main reasons people give for not using a password manager.

One important reason for using a password manager in the first place is so you don't have to remember a different, strong password for every site. Unfortunately, many password managers simply store whatever passwords you happen to use.

Some web sites force you to change your password on a regular basis. I have a rant on why forcing you to change passwords on a regular schedule is a bad idea.
Some password managers automate the process for you.
———————————————-
Why forcing password changes is a bad idea:

People use simple algorithms when forced to change passwords too frequently, e.g., MyBank2022, MyBank2023. Bad guys know this.

Home user: If a bad guy gets your bank password, your money is already gone. You only need to change your password if you think it might have been compromised.

Enterprise user: If a bad guy gets your corporate password and wants to lurk, you will have malware that will capture any new password you create. You must clean your

machine(s) before creating a new password.

What is a Password Manager?

A software application designed to **store** and manage online credentials. It also generates passwords. Usually, the passwords are stored in an encrypted database locked behind a master password. — malwarebytes.com

A password manager is an app on your phone, tablet, or computer that **stores** your passwords, so you don't have to remember them. — ncsc.gov

A password manager is a service that helps you generate and **store** long, unique passwords for all your online accounts. — consumerreports.org

I've been talking about password managers, but I haven't actually said what they are.

malwarebytes says it's an application that stores your passwords.

The government says it's an app on your device that stores your passwords.

Consumer Reports says it's a service that stores you passwords.

Sounds like there's a consensus. A password manager stores your passwords.

So What's So Bad about That?

- Where are your passwords stored?
  - On your machine?
  - In the cloud? Who's cloud?
- How are they stored?
- It costs money
    - To manage your account
    - To pay for the cloud resources

That definition raises some questions.

Where does it store your passwords?
On your machine? How do you get your passwords when you're at a friend's house?

In the cloud? Whose cloud? I have a cloud; I expect some of you have a cloud, but my sister doesn't have a cloud. That means your passwords are probably in the password manager's cloud.

How are your passwords stored? Encrypted, not always as it turns out, but who has the key? If them, can you trust them not to lose or abuse it? If you, what happens if you forget your password manager password?

And of course it costs money to manage user accounts and pay for cloud resources.

And then There’s This to Worry About

We recently detected unusual activity within a third-party cloud storage service, …

We have determined that an unauthorized party, …was able to gain access to certain elements of our customers’ information. **Our customers’ passwords remain safely encrypted due to LastPass's Zero Knowledge architecture**.

And then there’s this.

I got this notification from LastPass back in December. Apparently, someone stole their database. This is the kind of event that leads people to stop trusting their password manager.

Now there’s evidence that the password vaults have been penetrated to steal cryptocurrency seed phrases.

A Different Approach

Maybe we should look into a different approach.

A Different Kind of Password Manager

Don’t Remember Your Passwords, Calculate Them

- It can be free
  - No accounts to manage
  - No password storage needed
- Can even be used without network access
- **You are in control**
  - Carry a piece of paper with everything you need to get your passwords.
  - It’s not great if you lose it, but it isn’t terrible either.

Instead of storing your passwords, we can calculate them.

It can be free because there are no accounts to manage or external storage to pay for.

Since the calculation only needs local data, you can use it even without a network.

Most importantly, you are in control. You can carry a piece of paper with everything you need to get your passwords. It’s not great if you lose it, but it’s nothing close to the problems you have if you lose that booklet with your passwords on it.

A Brief History of SitePassword

I’m going to start the discussion of SitePassword with a history lesson so you can see where I stole the ideas from.

The Earliest Days

2003

In the early 2000s I got tired of having to remember all my passwords. I had just learned about hash functions and thought that could be the solution to my problem. Since I was between projects at HP Labs, I wrote a Python application.

The Earliest Days

2003

What you see here is a screen shot of a Windows executable that a colleague wrote. Other people contributed. One wrote a Java applet, someone else a version for PocketPC. Others produced versions for Palm, Nokia, and the UNix command line.

Lesspass is a modern version of this application.

HP Anti-Phishing Toolbar for IE

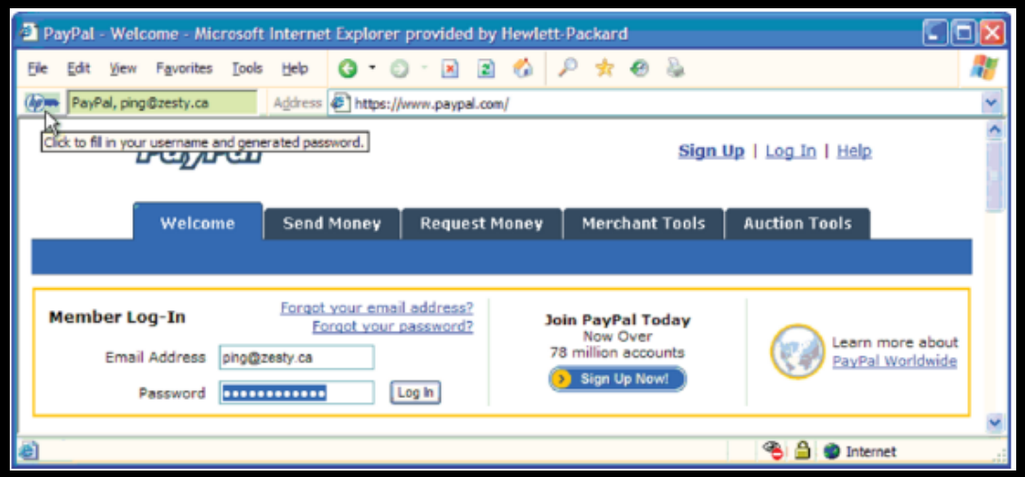
2005

A year or so later, folks from the HP Atalla Division came to us. (Atalla was a security company with close ties to the finance industry that HP had recently bought.) They wanted to show they had something to deal with phishing at an upcoming banking conference. They asked if we could turn the password calculator into a toolbar for IE. (Remember how everybody and his uncle was producing toolbars back then?)

I asked our intern, Ka-Ping Yee, how long that would take. “A couple of weeks,” he said grumpily. After the meeting, I tracked him down and asked him what was wrong, and he said, “You’re treating me like a code monkey.” “Ping,” I said, “You’re an intern. A code monkey is exactly what you are.”

HP Anti-Phishing Toolbar for IE

2005



Less than 3 weeks later we had what you see here. Ping did a spectacular job, as I later learned he always does. You fill in your nickname for the site, here PayPal, and your userid. The toolbar remembers them, so they show up automatically when you return to the site. You click on the button with the HP logo, and your username and password get filled in. If it's a phishing site, your settings don't show up, so clicking the button does nothing. This is probably the first application that deserves the term password manager.

The Atalla folks showed it at the conference, and It was well received, especially because it met the FISMA two-factor requirement, one factor being the master password, the other the local database holding the nickname for the site. As a result, HP offered this as an official product. The product was free, but businesses could pay for a version that put their logo on the button. So, if Wells Fargo wanted to distribute it to their customers, they could pay HP $1 per account to have their logo on the button. The nice part for them was that users would be clicking the Wells Fargo logo every time they logged in anywhere. In spite of that inducement, I don't know if anyone actually paid, though.

An important point is that the basic approach received a careful review by some serious security folks.

Chrome Extension

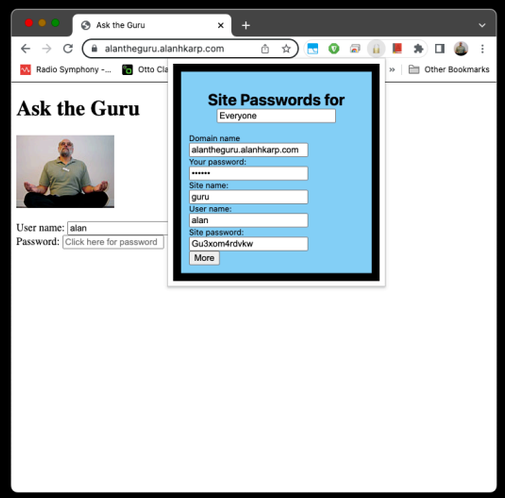
2012

It wasn’t long after that people began to recognize how vulnerable IE was and started to migrate to other browsers, mainly Firefox. I didn’t really have a choice. I had to choose Chrome because they cited our research in the academic paper they published on the browser.

Chrome Extension

2012

- Some Problems
  - No sync across machines
  - Unhappy with personas
  - Only finds 60% of password fields
- Only available for testers
- Only I used it for the next 10 years



Instead of a toolbar, Chrome lets developers produce plugins they call extensions. You can see that this extension is a blatant copy of what Ping did.

I wasn't entirely happy with it. Settings are easy for you to remember most of the time, but when they're not, it's really annoying that they aren't synchronized across your machines.

Personas are what you call it when you have a Facebook account for work and one for personal use, and you want to use different credentials for them. Or, you have a shared machine at home. Everyone uses the same Netflix login, but each person wants their own Instagram credentials. This version handled personas, but the code was more complicated than I was comfortable with.

Also, it only found the password field about half the time, so you lost autofill, the main usability feature.

That's why I only made it available for testers. I did use it myself for the next 10 years, though.

And Then
2021

- Google says I have to update it.
- “No problem,” I say to myself. “It’ll only take a week or so.”
- A year later …….

Then, in late 2021 Google told developers that we would have to update our extensions. The startup I was working for was in the process of shutting down, so I figured I’d take a few weeks to make the changes. Here I am talking to you over a year later, and I only put it into the Chrome Store for testing in January of this year.

Why It Took So Long

- Google's changes
  - From a persistent background page to a transient service worker
  - Everything is async
- Fixed what I didn't like
  - Always(?) finds password field
  - Synchronizes across machines
  - Handles different personas

So, why did it take me so long? Here's what it looks like now. Much better thanks to Dale Schumacher, but that wasn't the reason for the delay.

The original version used something Google called "Manifest 2," which had a persistent background page. SitePassword never stores your master password, but I could keep it in the process running the background page and be sure it was there when you needed it.

That's not true for the new version that has to use what Google called "Manifest 3." There's no persistent background page, just a transient service worker that can stop at any time. When it does stop, it loses its memory, so I had to find a different place to remember your master password. It turns out that something called "session storage" has all the right properties.

The second complication is that almost anything you want to do with Manifest 3 involves async. Now, I've done my share of async programming, but this was the first time I converted a sync program to async. Let's just say that I made more than my share of mistakes.

In addition, since I was out of work, I decided to fix the things I didn't like about the earlier version. For reasons I'll explain later, it took me some 6 months to get to the point where I could find the password field on every single one of the 120+ sites that I test with. I also spent a fair amount of time trying different approaches to synchronization and handling personas until I came upon my solution. Let Google do the heavy lifting

Let Google Do the Heavy Lifting

- Store settings in bookmarks
  - Every browser syncs bookmarks
  - But Google doesn’t merge duplicates
- Supporting personas with Google Profiles
  - Switch Profile without logging out
  - Each Profile has its own bookmarks

I store your settings in bookmarks, which has a number of advantages.

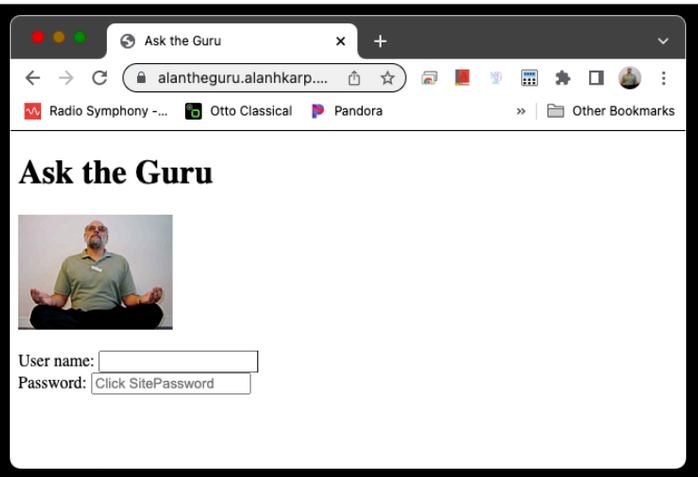
- No need for user accounts or cloud storage.
- Bookmarks sync to your mobile devices.
- Each persona automatically gets its own bookmarks.
- You don’t lose your settings if you have to re-install the extension.
- You can import your bookmarks to other browsers.

The only problem I had to deal with is the fact that Google allows multiple bookmarks with the same name, but I found a simple solution. SitePassword deletes one if they are identical and asks the user if they are not.

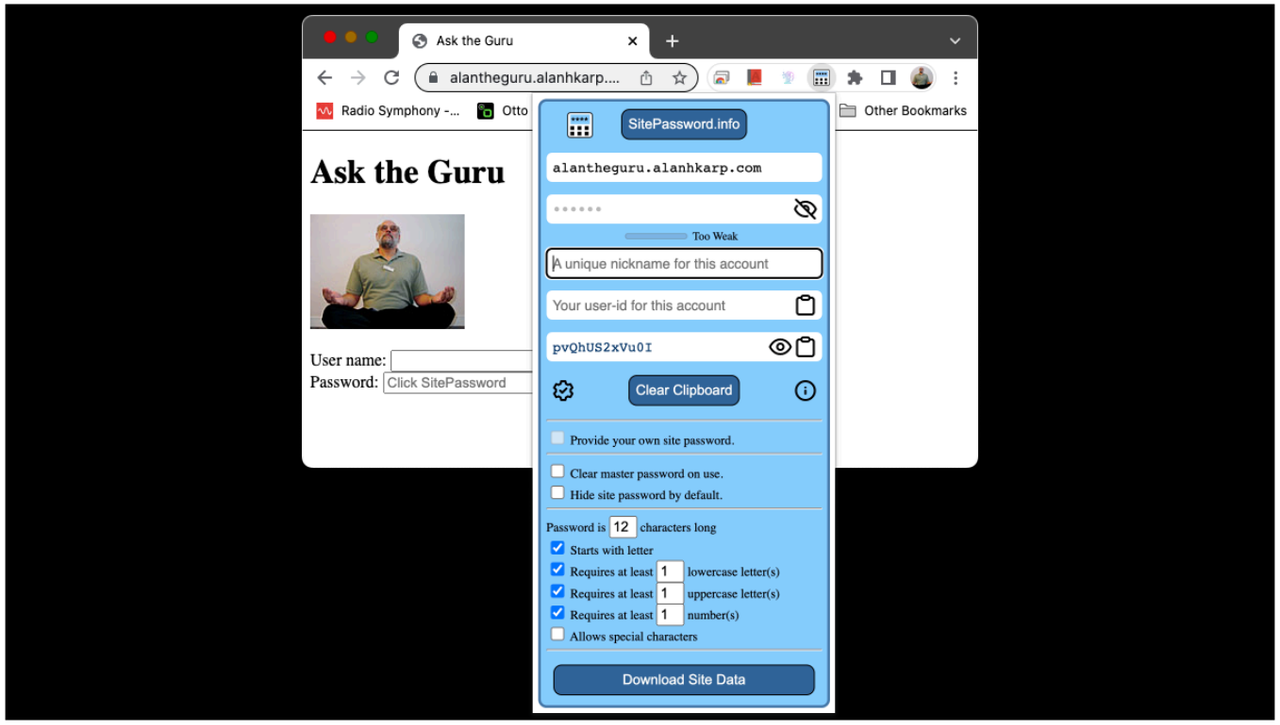
Google Profiles provide everything needed to support personas.

A Taste of SitePassword

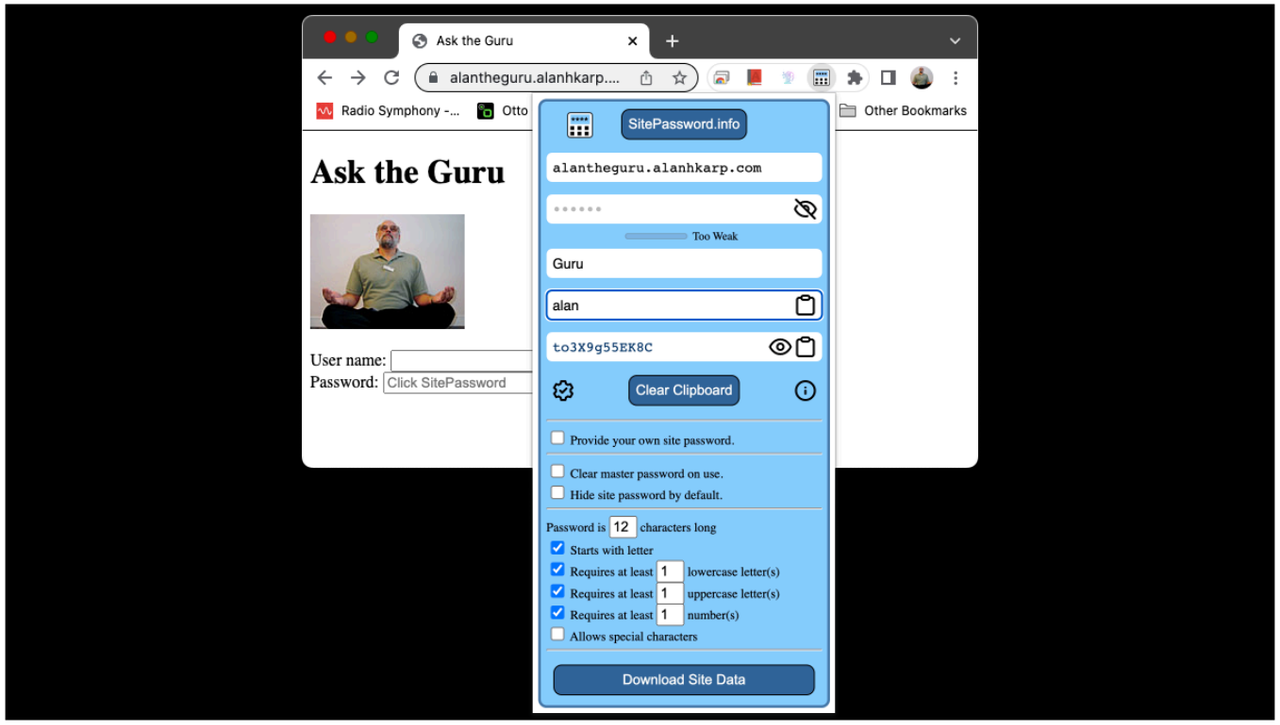
Here’s a taste of SitePassword. I’ll show a demo later if time permits.



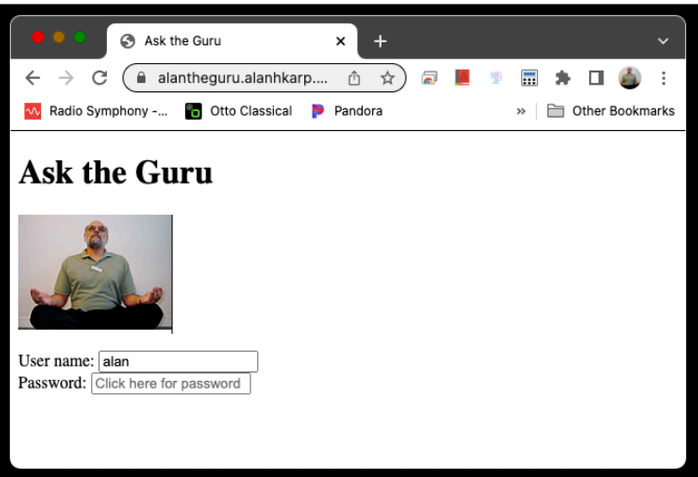
The first time you visit a login page after installing the extension, you see something like this page. The password field says to “Click SitePassword.”



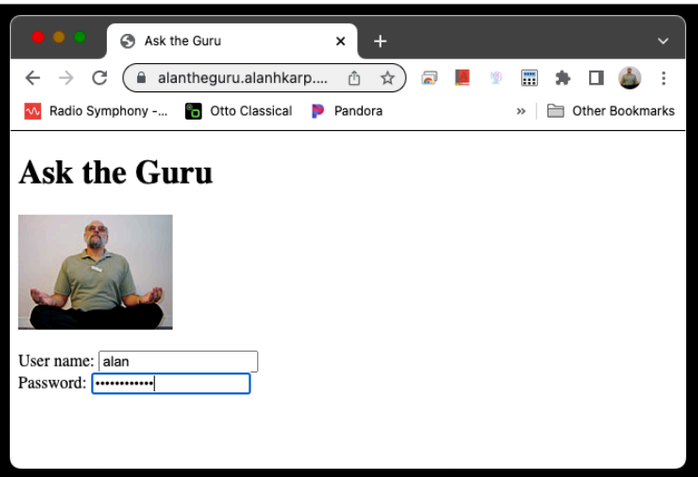
Click on the SitePassword icon, and see the form to complete.



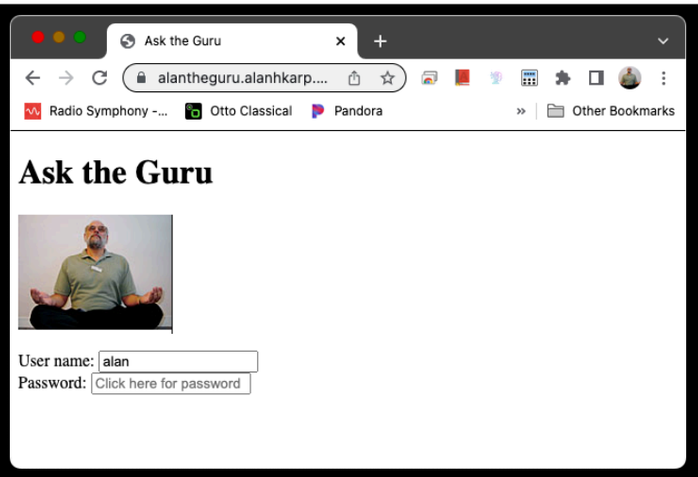
Fill in the form. Note that your site password starts with to3X.



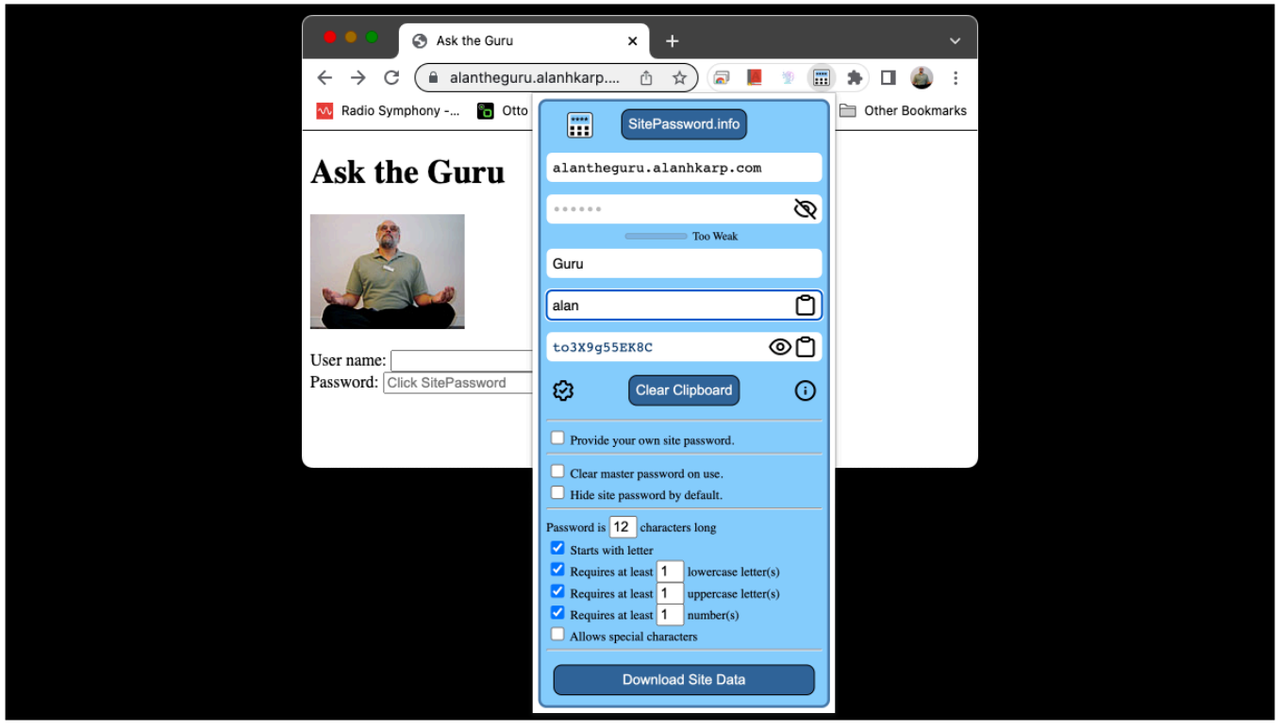
After you mouse out, you see your user name filled in and the password field says, “Click here for password.”



Click on the password field, and you’re ready to log in.

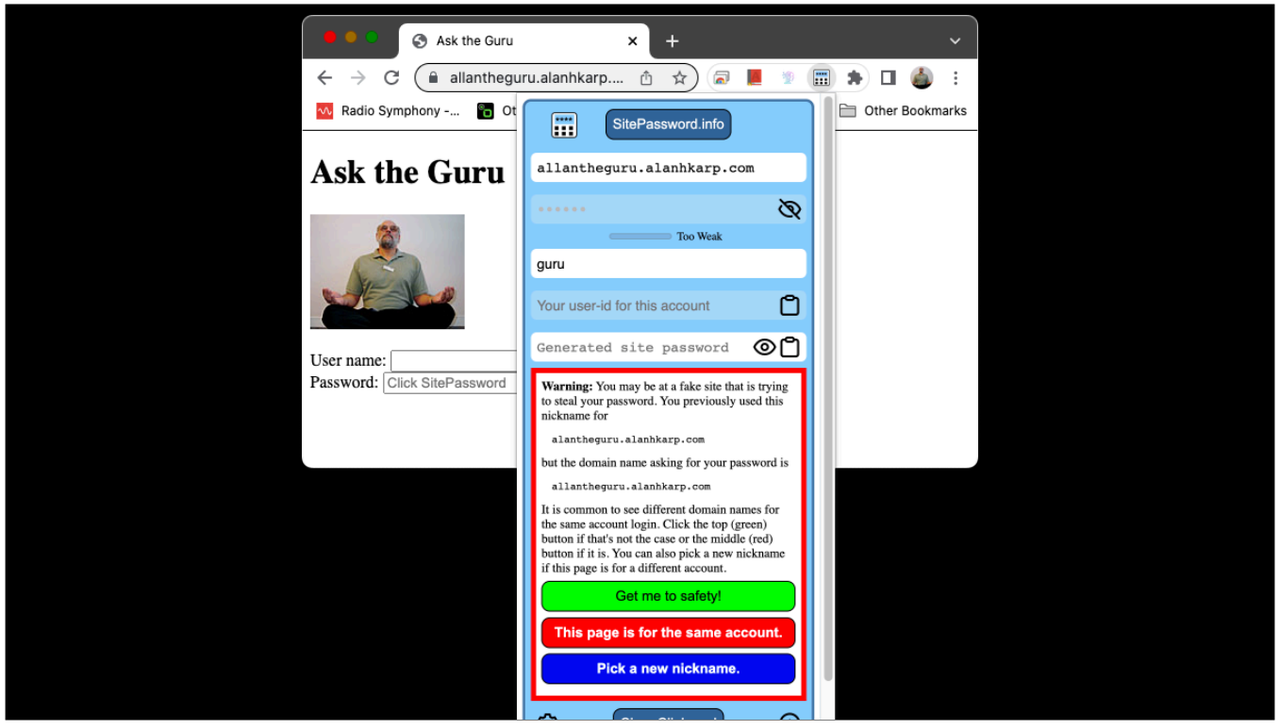


Now for the best part. When you return to the page on any machine that syncs your bookmarks and has the extension installed, all you have to do is click the login field.

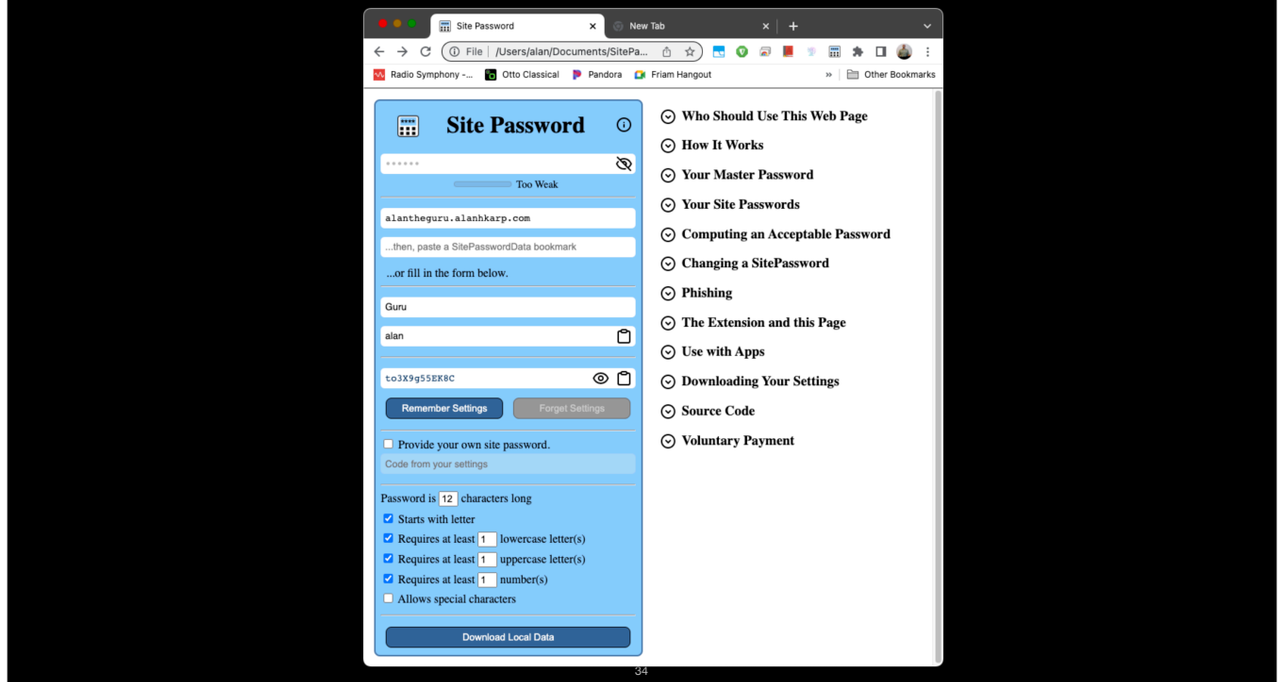


Remember that piece of paper?

The Download Site Data button creates a file with a nicely formatted table of your settings for each site. You can print it out and carry it in your pocket. However, if you store that file in the cloud, you don’t even have to type in your settings. You can copy the bookmark from a link stored included in the file.



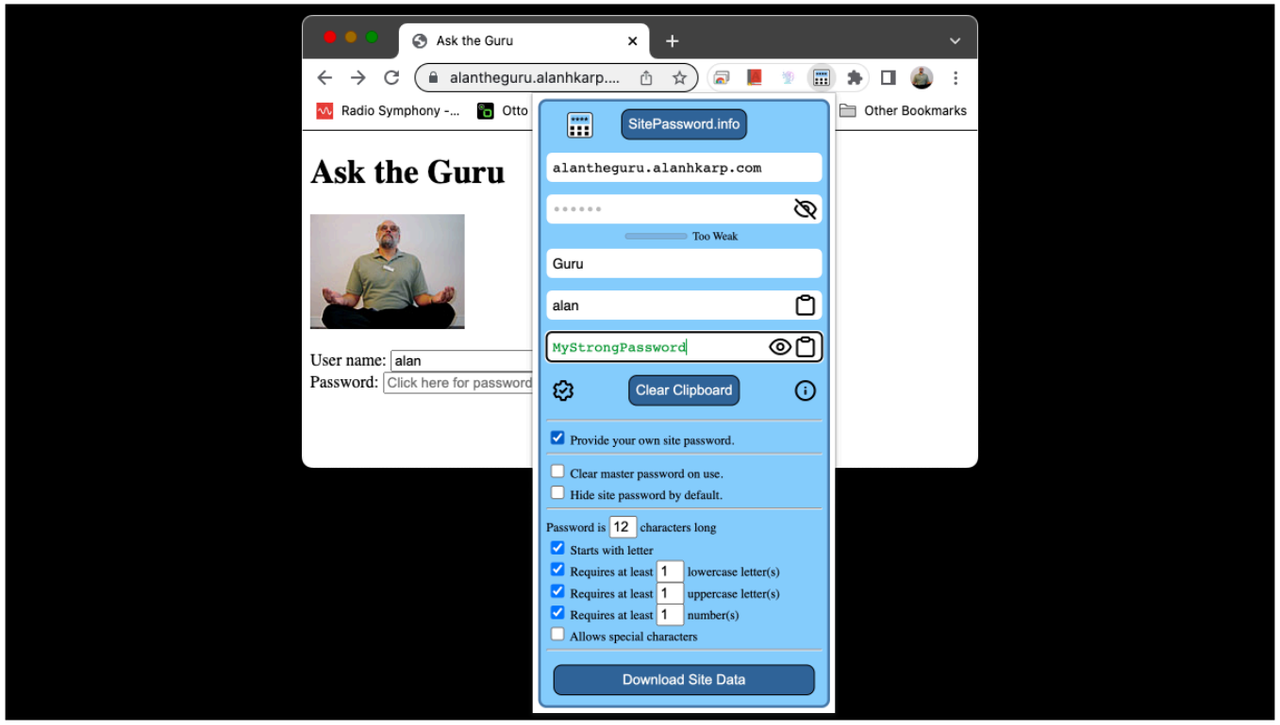
SitePassword started as an anti-phishing toolbar, so it warns you if you might be at a phishing site. If you open SitePassword and try to fill in the site name you used for a different domain, you get this big, scary warning. I'm sure you can't read the warning, but it says that the domain name with 2 ells is not the one you set up for. Clicking on "Get me to safety!" will take you to your home page. Clicking the red button remembers your settings so you don't see this warning again for this domain.



What if you're at friend's house? Well, there's a web page for that.

It's basically the same form, and you don't even have to remember your settings. You can use your bookmarks. That's really nice when you're on your phone, which doesn't allow browser extensions but does sync bookmarks. Note that you get the same site password, the one that starts to3X.

SitePassword has extensive instructions, but my hope is that nobody will have to read them.



There may be times when you'll have to provide your own password. Maybe you can't satisfy the password rules, although that's never happened in 10 years of use and testing on some 300 sites. Maybe you can't figure out how to change your password, although the Forgot Password option should get you over that hump. Maybe you've been given a password that you're not allowed to change, as some companies do.

SitePassword stores the password you provide encrypted with the computed one in the bookmark for the domain, so you can retrieve the password you provide using that same piece of paper. That feature makes SitePassword a hybrid password manager.

Usability

I noted earlier how important usability is to adoption.

Usable Security

No system can be secure if users don’t understand the implications of their actions.

but the real issue is usable security, which recognizes that no system can be secure if user’s don’t understand the implications of their actions. That statement, while true, isn’t very helpful in designing the user experience.

Ping's 10 Princples for a Secure UX

1. Path of Least Resistance: Make the easy way the secure way.
2. Appropriate Boundaries: Make distinctions matter to the user.
3. Explicit Authorization: Authorize only by explicit user action.
4. Visibility: All security related decisions must be viewable.
5. Revocability: Make it easy to revoke granted permissions.
6. Expected Ability: Don't make user think the tool will do what it can't
7. Trusted Path: It should not be possible to spoof the UI.
8. Identifiability: Give each object an identifer that the user understands.
9. Expressiveness: Allow users to express policies that they want.
10. Clarity. Make effects of actions apparent to the user before acting.

Fortunately, there's Ping's 10 principles. (Yes, it's the same Ping.) Many of these don't apply to SitePassword but several do.

Ping's 10 Princples for a Secure UX

1. Path of Least Resistance: Make the easy way the secure way.
2. Appropriate Boundaries: Make distinctions matter to the user.
3. Explicit Authorization: Authorize only by explicit user action.
4. Visibility: All security related decisions must be viewable.
5. Revocability: Make it easy to revoke granted permissions.
6. Expected Ability: Don't make user think the tool will do what it can't
7. Trusted Path: It should not be possible to spoof the UI.
8. Identifiability: Give each object an identifer that the user understands.
9. Expressiveness: Allow users to express policies that they want.
10. Clarity. Make effects of actions apparent to the user before acting.

Make the easy way the secure way: It's easier to login with SitePassword than it is to type even a simple password.

Ping's 10 Princples for a Secure UX

1. Path of Least Resistance: Make the easy way the secure way.
2. Appropriate Boundaries: Make distinctions matter to the user.
3. Explicit Authorization: Authorize only by explicit user action.
4. Visibility: All security related decisions must be viewable.
5. Revocability: Make it easy to revoke granted permissions.
6. Expected Ability: Don't make user think the tool will do what it can't
7. Trusted Path: It should not be possible to spoof the UI.
8. Identifiability: Give each object an identifer that the user understands.
9. Expressiveness: Allow users to express policies that they want.
10. Clarity. Make effects of actions apparent to the user before acting.

Explicit Authorization: SitePassword won't insert your password until you click the password field. That has 2 effects. It avoids a particular family of attacks, but maybe equally important it gives you control over where your password goes.

Ping's 10 Princples for a Secure UX

1. Path of Least Resistance: Make the easy way the secure way.
2. Appropriate Boundaries: Make distinctions matter to the user.
3. Explicit Authorization: Authorize only by explicit user action.
4. Visibility: All security related decisions must be viewable.
5. Revocability: Make it easy to revoke granted permissions.
6. Expected Ability: Don't make user think the tool will do what it can't
7. Trusted Path: It should not be possible to spoof the UI.
8. Identifiability: Give each object an identifer that the user understands.
9. Expressiveness: Allow users to express policies that they want.
10. Clarity. Make effects of actions apparent to the user before acting.

SitePassword isn't about granting permissions that you might want to revoke, but you can easily forget the settings for a site by deleting its bookmark.

Ping's 10 Princples for a Secure UX

1. Path of Least Resistance: Make the easy way the secure way.
2. Appropriate Boundaries: Make distinctions matter to the user.
3. Explicit Authorization: Authorize only by explicit user action.
4. Visibility: All security related decisions must be viewable.
5. Revocability: Make it easy to revoke granted permissions.
6. Expected Ability: Don't make user think the tool will do what it can't
7. Trusted Path: It should not be possible to spoof the UI.
8. Identifiability: Give each object an identifer that the user understands.
9. Expressiveness: Allow users to express policies that they want.
10. Clarity. Make effects of actions apparent to the user before acting.

Expressiveness: Here expressiveness is the part of the menu that adapts the computed passwords to the site's rules. I believe that SitePassword is the only password calculator expressive enough to cover nearly all of the sites in dumbpasswordrules.com.

My User Experience Goals

Make it as easy to use as Candy Crush

- Can’t break anything by experimenting
- A Help button is a crutch for developers
- Encourage good practices
  - Warn on weak master/site password
- Choose a readable font, particularly for site password
- Focus the field the user will be needing next

In addition, I have my own guidelines. I want the tools I build to be as clear to use as a game. You don’t read the instructions for Candy Crush on your phone; you just play with no worries that you might break something.

I also believe that a help button is a crutch for developers. I’ve heard more than one say, “Oh, we’ll just explain that in the Help.”

You want to encourage good practices. That’s why SitePassword has a strength meter for your master password and will let you know if a site password is weak.

A readable font is important, which a hard earned lesson for me. You don’t want to have to enter your Netflix password again because you thought that character was a lower case ell when it really was an upper case eye. Using the earlier version, it took me 3 tries to login to Netflix on my new smart TV.

Users often are uncertain about what to do next. One hint you can give them is to focus on the field they need to fill in.

The last item is also a pet peeve of mine. Did you ever have a code texted to you as a second factor, you type in the numbers, and then see that nothing happened because the field wasn’t focused for you? Aaarrrggghhh!!!

My User Experience Goals

Where I failed

- Can’t always make “Click here for password” show up
  - Resorted to tooltip, but it takes a second or so to appear
- Forgetting settings for a domain
  - Delete the corresponding bookmark
- Closing the popup

There are still some things I’m not happy with.

The placeholder doesn’t show up on all pages in spite of my best effort. I spent the better part of a week trying to make it show up on one site and never figured out why it didn’t. I resorted to a tooltip, but how do you tell the user to hold the mouse over the password field for a second or two?

There are times when you want to forget the settings for a domain name. I haven’t figured out a way to make it obvious that all you need to do is delete the corresponding bookmark.

Closing the popup is an odd problem. If I close it when the popup loses focus, you have to click twice, once to close the popup and once to insert the password. Yuck! If I close the popup when you mouse out, it’s too easy to do that accidentally. Double Yuck! Instead, I wait 3/4 of a second before closing the popup when you mouse out and keep it open if you mouse back in before it closes. I hope that’s long enough so you can get your mouse back and short enough that you don’t have to click twice to get your password.

Password Manager Security

Another aspect of using a password manager is the security piece.

Mobile Devices
Situation is Really Bad

- Too easy to spoof app identity
- iframes inside Android WebView can spoof messages
- One password manager pulled their app

The situation on mobile devices is really bad. You can build an app that will fill in the password field of another app. That's great, but it's too easy to spoof app identity, so you might be giving away your password. There's a similar problem with WebView on Android.

The situation is so bad that one password manager pulled their app.

I won't be talking about mobile further.

Built into the Browser

- Pluses
  - Better than not using a password manager
  - Never resorts to the clipboard
  - Always finds password field
- Minuses
  - Not as secure
  - Some don’t generate passwords, only store them
  - Many don’t consider the password policy

All browsers have a built-in password manager. Typically, they’re not great, but they are far better than not using one at all. As far as I can tell, they always find the password field, so you don’t have to use the clipboard. That’s dangerous because other programs can read what you put there.

The big problem is that the articles I read say they are not as secure as standalone password managers except …

Any password manager that asks you to provide passwords has a usable security issue by making it too easy to just keep using weak/reused passwords. Also, not all password managers that generate passwords take into account the password rules for the site.

Chrome Browser Password Manager

As Bad as it Gets

- Autofills with no user action — Makes it easy to steal passwords
- Can’t customize generated password —Might not be able to use password
- Passwords not encrypted by default
- Warns on weak passwords only after you check
- Assumes any field of type “password” holds a password
- Uses same password for smile.amazon.com and amazon.com
  - but not amazon.co.uk

I decided to see how bad it was and set up some passwords with the Chrome browser password manager. It wasn’t good.

It autofills passwords without any user action. That means your password can be stolen if a bad guy can inject a password field at any page in the domain of the login page. It could be an FAQ or a “Contact Us” page, one that nobody thinks has any security implications.

There’s even an attack that only takes a few minutes to steal all the password you password manager has stored.

There’s actually a way the browser can block these attacks that isn’t available to extensions, but I verified that Chrome doesn’t do it.

I couldn’t figure out how to tell the browser about the funky password rules at my power company’s web site.

You can encrypt your passwords, but it’s not the default, and you get a scary warning if you do.

The browser does warn you if you’ve created a weak password but only after you explicitly check.

It also assumes that any input element of type “password” holds a password, but many don’t. If the site doesn’t think it’s been given a password, it might not put as much effort into protecting it.

It uses the same password for smile.amazon.com and amazon.com but not amazon.co.uk. In general, it’s very hard to know if two domain names are for logging into the same account. Now that might be fine if they’re checking the TLS certificate, but I can’t be sure. Unfortunately, Google doesn’t make the TLS certificate available to

extensions.

I think the minuses of using the built-in password managers outweigh the plusses, so I won't talk about built-in password managers further.

It’s More than Just Passwords Strength

- Autofill with no user action
  - Put the password into a form controlled by an attacker
- Storage Security
  - Stolen, didn’t get the passwords but got the metadata
- User Communication
  - Inform on weak/reused passwords
  - Generated passwords sometimes weak (oMMMMMMT?m*m)

Getting a different, strong password for every site is just the beginning of password manager security.

I’ve already mentioned autofill, and we’ve seen that the databases holding your passwords can be stolen, but warning the user about weak and reused passwords is also important. The example given here comes from a person who had a password manager generate more than 140,000,000 random passwords. About a dozen were easy to guess. For example, the one shown here can be guessed in about 10 seconds at 1 M guesses/sec.

User communication is handled in SitePassword with the strength meter for you master password. Also, your site password will be in a noticeable color should SitePassword accidentally computes a weak one. That’s why the provided site password in an earlier slide was green indicating that it was good but not great.

It’s More than Just the Passwords

- Phishing sites and phishing the password manager password
- Domain name errors — google.evil.com vs google.com
- Accessibility features
- Code size - one password manager is over 250,000 lines of code
  - SitePassword 1,500 lines of JavaScript, 750 HTML, 250 CSS
- jQuery on the login page
- Worst of all - Not using a password manager due to lack of trust

I think your password manager should warn you if you’re at a possible phishing site, but you also have to be worried about being phished for your password manager password. After all, a site could show you a form that looks just like your password manager’s.

I mentioned the problem of amazon.com and amazon.co.uk. It sure would be nice if I could tell you that they were both for logging into the same account, but I can't. For example code from a highly up-voted answer on StackOverflow tells you that you should use the same password for mail.google.com and maps.google.com, but it also says to use that same password at google.evil.com, probably not a good idea.

Accessibility features are interesting. For example, you don’t want a screen reader to say your passwords out loud when you’re in a public place.

Some of the open source tools I’ve looked at are HUGE. One is over a quarter million lines of code. SitePassword is only 1% of that, but that means I produced about 10 LOC/work day. I don’t know how happy you’d be with that if I was on your team.

I think a big problem is the login pages that rely on jQuery. Now, don’t get me wrong. I love jQuery. It reads your mind. It does want you want not what you said. Still, it’s a big dependency to have on a security critical page. SitePassword is self-contained; it never talks on the network once it’s loaded.

Your password manager must convey that you can trust it. A lot of people report giving up on a password manager because they didn’t trust it.

SitePassword Security

So, how does SitePassword stack up?

The Good

- Require click on password field
- Callback registered only on visible password fields
- Use iframe domain name if its password field clicked
- Use zxcvbn() from Dropbox for password strength meter
- Site name and user name act as salt to defeat pre-computation attacks
- Warn if password might still be on the clipboard

I think I've done a pretty good job.

You have to click on the password field, blocking an entire family of attacks.

I only register a callback for visible password fields, which defeats some clickjacking attacks.

I use the iframe domain name when looking up settings. The only downside is that users will see a bookmark with a strange domain name.

It used to be that knowing how strong a password is was difficult, but we now have this nice tool from Dropbox. It's so good that I'm using it even though it takes the SitePassword distribution from <100KB to >1MB.

Attackers will precompute the hash of all possible passwords up to some size, 8, 12, even 14 characters to create rainbow tables. That might work for a site password, but your master password is safe since it's combined with your user and site names.

Of course, it's not a good idea to leave a password on the clipboard any longer than necessary, so SitePassword lets you know if that might be the case.

The Not So Good
But for Good Reasons

- Settings in bookmarks
  - Settings not encrypted
  - Stored on disk and in Google cloud
- Site password visible by default
  - Gives you a sense of how random looking the site passwords are
  - Don’t often open the popup
- 12-character site passwords by default

I did make some compromises in the interest of usability.

Bookmarks are very well protected from code running in the browser but not from malware on your machine. Of course, if you’ve got malware on your machine you’ve got bigger problems than losing your settings, and if Google gets hacked, life as we know it ends.

I chose to make the site password visible by default. Watching it change as you type gives you a sense of how uncorrelated your passwords are. Besides, you rarely have to open the popup, just to set up a new site and to enter your master password at the start of a browser session, or when you have to put a password on the clipboard.

I originally chose 12-characters as the default because many sites didn’t take longer ones.

The Ugly

An offline attack against master password

1. You create an account at a bad guy's site
2. Bad guy knows site password and username and can guess site name
3. Bad guy starts guessing master passwords

- Mitigations
  - Strong master password
  - Hash a minimum of 100 times to get site password
  - Multiple master passwords

Unfortunately, there is a serious attack, but it's no worse than one against a stolen password database.

There are two kinds of guessing attacks, online and offline.

In an online attack, the bad guy goes to a bank login page, enters your userid, and starts guessing passwords. That's about all an attacker who just gets your bookmarks can do. Fortunately, this attack is slow and likely to be detected.

Offline attacks are more serious. The attacker might steal a password database, which stores the hashes of users' passwords, and start guessing. That can be fast and is undetectable.

The attack is still there with SitePassword but it works differently. Say that you create an account at a rogue site. The bad guy now knows your userid and site password, and can probably guess your nickname for the site. Now he can start guessing master passwords.

You can't eliminate that threat with SitePassword, but you can mitigate it with a strong master password. It would take 3 days to guess a 12-character password at 1 M guess/sec. Guessing would take centuries if you used 15 characters.

SitePassword also increases the work needed for a guess by hashing 100 times before showing a site password. That means 1M guesses/sec would take almost a year to guess a 12-character master password.

You can also use different master passwords, one for banking an health, another for subscriptions, and a third for sketchy sites. You don't have that choice with the

leading password managers.

War Stories

One of things I miss from not going to the office is hearing other people’s war stories, the crazy stuff they ran into and how they dealt with it. I hope you enjoy mine.

Finding the Password Field

Websites do some weird #%^@

- Put password field in an iframe with a different domain name than page
- Add password field dynamically
- Add password field with type=text and change to type=password
- Many password fields on page but only one visible
- Some sites add CSS at runtime that make the password field visible
- Make password field visible only after you click a button

As I said, it took me 6 months to find the password field on all my 120+ test sites.

It's quite common to put a password field in an iframe. That's the most common reason for the earlier version's failure to find the password field. Fixing that got me from finding the password field on 50 or so of my test sites to over 80.

Several sites build the page when you load the URL; if you View Source, you only see a few lines of HTML. Fortunately, there's something called a MutationObserver that lets me know when things change. This one got me 20 or so more.

The rest of these changes got me 1-5 each.

A few sites create the password field as a text field and later change it to type password, but that change doesn't count as a mutation. I had to add special code.

Dropbox used to have 6 password fields on the page with only one visible when you log in, but determining visibility is tricky as I'll show later.

Two sites add CSS at runtime that makes the password field visible. Unfortunately, that doesn't count as a mutation, so more special code.

Several sites don't make the password field visible until after you fill in your userid. That means I can't fill it in for you, but once you do, I find the password field every time.

Finding the Password Field

Websites do some weird #%^@

- Change contents of page based on the fragment
- Password field in a shadow root (shadow DOM)
- Clears the field after I set it (Who the %#^@ knows why)
- Don’t want to update password field at sitepassword.info

One site changes the look of the login form depending on how you got there, yet more special code.

One site puts the login form in a shadow root. Do you know what that is? I’d never heard of it. It was designed to completely isolate a complete document inside it from the main page, but I’ve been told by someone who knows that the design is flawed. More special code.

Then there’s the bank clears the user name and password fields 2 seconds after I fill them in. I have no idea what that’s for. I couldn’t even figure out where in their code they were doing it even though I single stepped through thousands of lines of code. The workaround — more special code.

Finally, I loaded up sitepassword.info for testing and the password field said, “Click SitePassword,” which I found annoying, so I added a special case for that URL.

Is the Password Field Visible?

Harder to figure out than you may think.

- Does window.computedStyle(element) say it's visible?
  - Correct most of the time but not always
- Is parent visible?
  - offsetParent != null if position != 'fixed'
- One clickjacking trick
  - element.style.opacity = "0" reported visible

Finding out if an element is visible is a challenge.

window.computedStyle gets it right except when it doesn't.

The offsetParent is another test, but it took me 2 days to learn that it doesn't work if position==fixed. Unfortunately for me, that piece didn't show up on the documentation page I was looking at until I accidentally scrolled down.

There's many ways to make a field invisible, but a common one is to set the opacity to 0 or a small value. This one's easy, since there's no good reason to change the visibility of a login form.

Finding the UserID Field

Heuristic Used by Many Password Managers

- UserID field is the one immediately before the password field
  - Except when it's not
- Sometimes to the left
  - Arabic, Hebrew to the right?
- Sometimes intervening fields
  - Some of them not visible

Finding the user ID field has its own challenges. Most password managers us the simple heuristic of choosing the element immediately preceding the password element. You do have to deal with intervening fields, but they are almost never visible. SitePassword gets this wrong on only 1 of 120+ sites. At least one other password manager gets it right by having a special case just for that website.

Crazy #%^@

Done by big companies with professional programmers

- Multiple domain names for the same login form
- Login form at a completely different domain
- Username and password fields have the same id
- A bunch of errors when the page loads
  - Took 30 seconds to time out on 4 bad GETs

Then there's the really crazy stuff, things you might expect from your corner bagel shop who hired the neighbor's teenager to build their web site, but not major banks and retail companies.

As we've seen there are multiple domain named for the login page; one bank has secure01 … Those show up as phishing warnings the first time you encounter a new one.

My health insurance company has me log in at a completely different domain. Are they training me to be phished? How are user's supposed to know what bookmark to use?

An HTML element can be given an id. Almost anything will do. The only rule is that ids must be unique on a page. Good rule, but don't count on it. A major store uses the same id for the user name and password elements, which meant I had to change the way I kept track of the password field.

Companies work really hard to improve the performance of their web pages, but you'd never know it by looking at the developer's console. There's often a gazillion errors. A particularly egregious example took 30 seconds for the page to finish loading, which meant SitePassword couldn't show you your user name for that long. Of course, the first reaction is that there was something wrong with SitePassword. The cause was 4 failed GETs done sequentially, each one timing out after 7.5 seconds. I went back to show someone, but they now use a different logon page.

How Bad Is It?

One password manager has
special cases for over 200 sites.

How bad is it? One password manager has special-cased over 200 sites, and I guess they've got people tracking those sites in case they change how they do things.

Summary

Ok. Let’s wrap up.

Future Work

- User studies
- Security review
- Other browsers
  - Works on Safari, Firefox and most Chromium browsers, e.g., Brave, Chrome, Edge
- IoS and Android apps — Monitoring security issues

It’s been a year, and I still have work to do.

I’d love to do user studies, but that’s really hard for someone working alone, so I’m asking for your help. Try it out, and give me your feedback.

A security review would be nice, but paying for one is out of my price range.

SitePassword works in all the major browsers, Firefox, Safari, and most Chromium browsers, such as Brave, Edge, and Opera, but I need to do a lot more testing.

And I’m keeping my eye on mobile, but I won’t be building a SitePassword app until those security issues I talked about get fixed.

References

Slides at https://alanhkarp.github.io/SitePassword/SitePasswordTalk.pdf

- Ping's 10 Principles - http://zesty.ca/pubs/icics-2002-uidss.pdf
- Oesch Thesis - https://trace.tennessee.edu/cgi/viewcontent.cgi?article=7785&context=utk_graddiss
- Oesch Paper - https://www.usenix.org/system/files/sec20-oesch_0.pdf
- https://eprints.whiterose.ac.uk/158056/8/Revisiting_Security_Vulnerabilities_in_Commercial_Password_Managers_2.pdf
- https://www.usenix.org/system/files/soups2019-pearman.pdf
- https://crypto.stanford.edu/~dabo/pubs/papers/pwdmgrBrowser.pdf

Here's where I got most of my information. Don't bother copying them down. These slides are at alanhkarp.github.io/SitePassword/SitePasswordTalk.pdf.

The paper on Ping's 10 principles is well worth reading.

The Oesch thesis is amazingly comprehensive, but the paper is a good summary.

Takeaways

- Use a password manager - a bad one is better than none at all
- Turn off any setting that fills in the password without a user action
- Prepare for the worst - Some have gone out of business
- Use it wisely - use strong passwords even if you don’t have to

OR

Take control and use SitePassword

Use a password manager. I can’t stress that enough. My son uses the one built into Chrome. In spite of all the problems I pointed out, I haven’t told him not to use it out of fear he won’t use a password manager at all.

Some autofill without a user action. Turn that off, or be very careful about using public WiFi. If you find a replacement, don’t use that tool.

But be ready in case they raise their prices too much or go out of business.

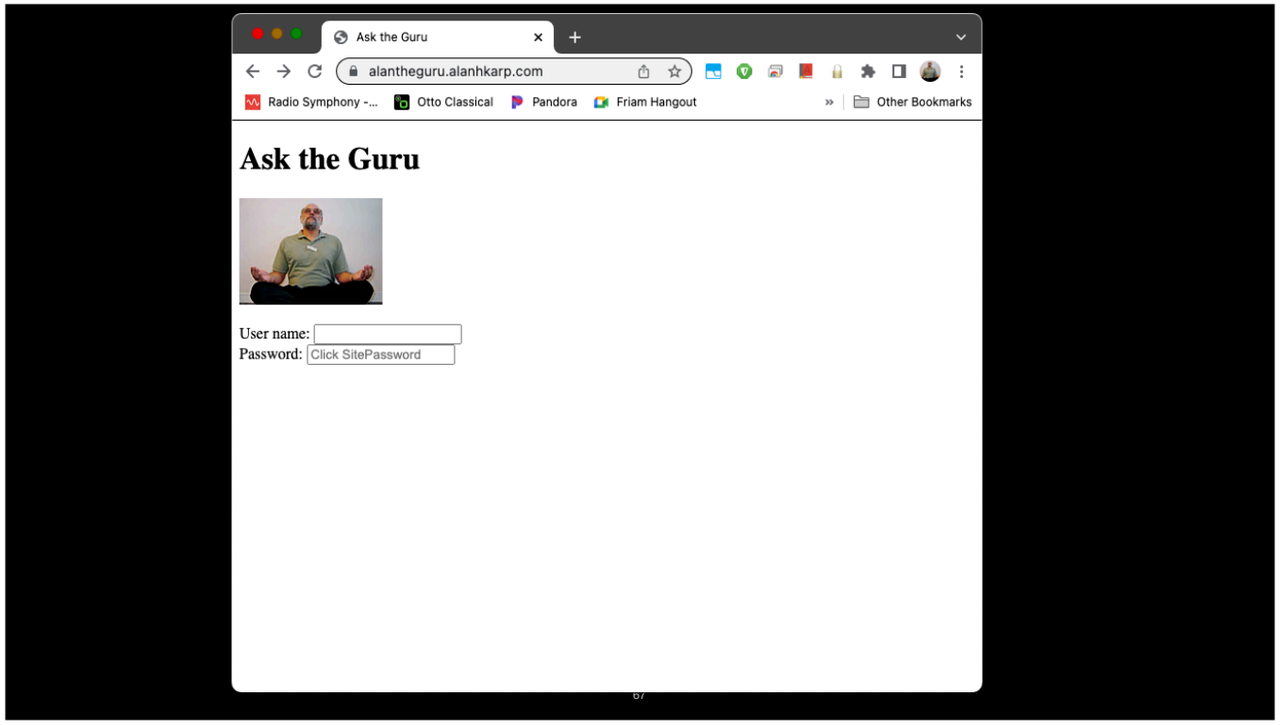
Make sure you use strong passwords even if the manager lets you use weak ones.

Or use SitePassword and be in control.

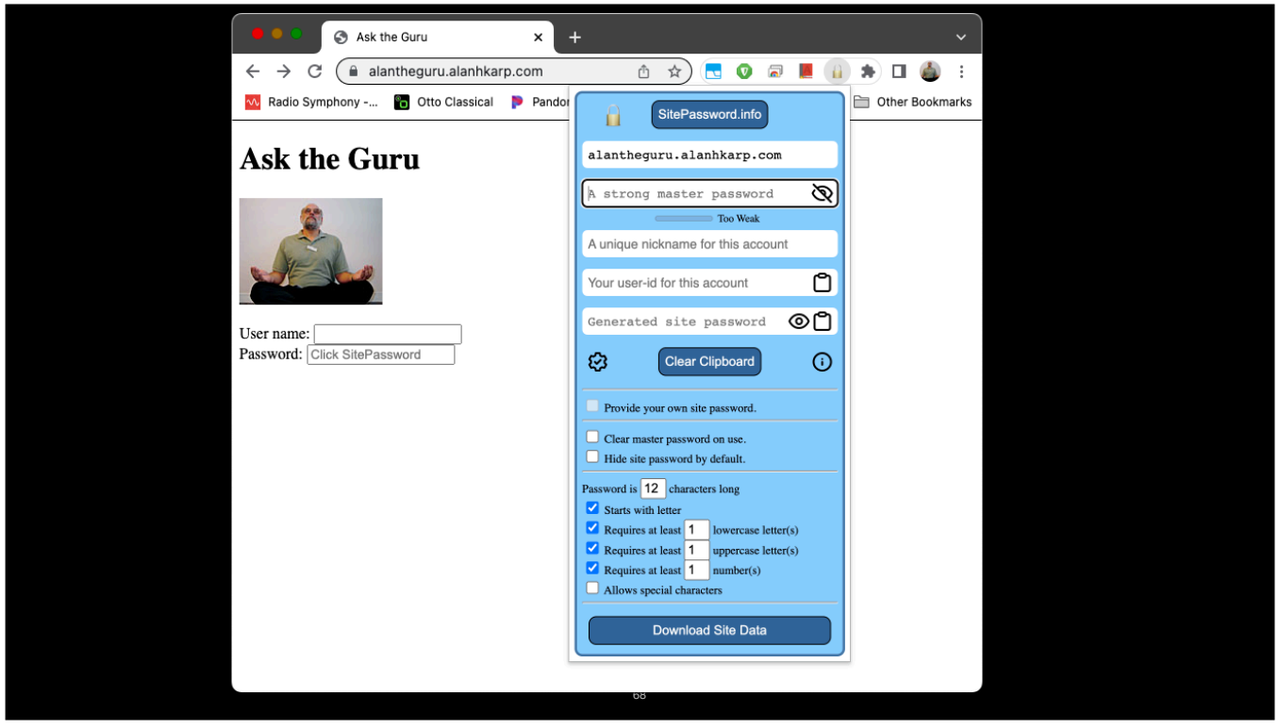
Thanks, I’ll take questions now.

Demo

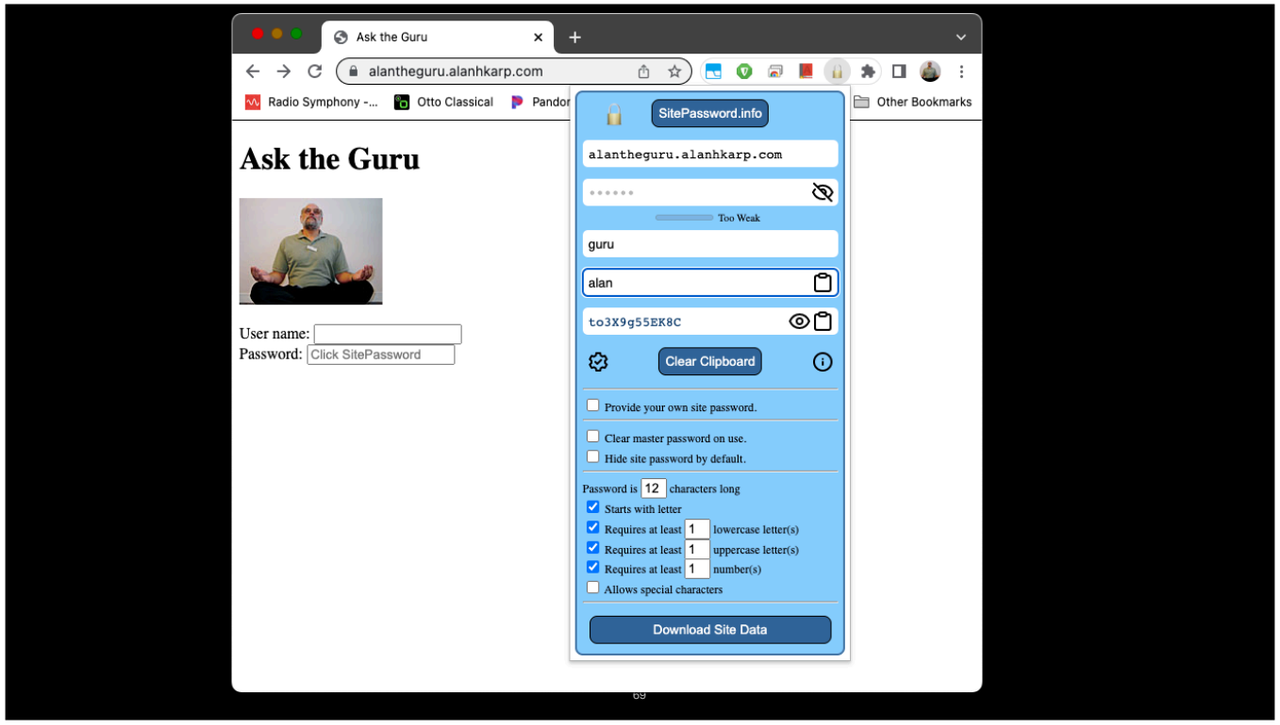
I want to show you how it works before I go into more detail.



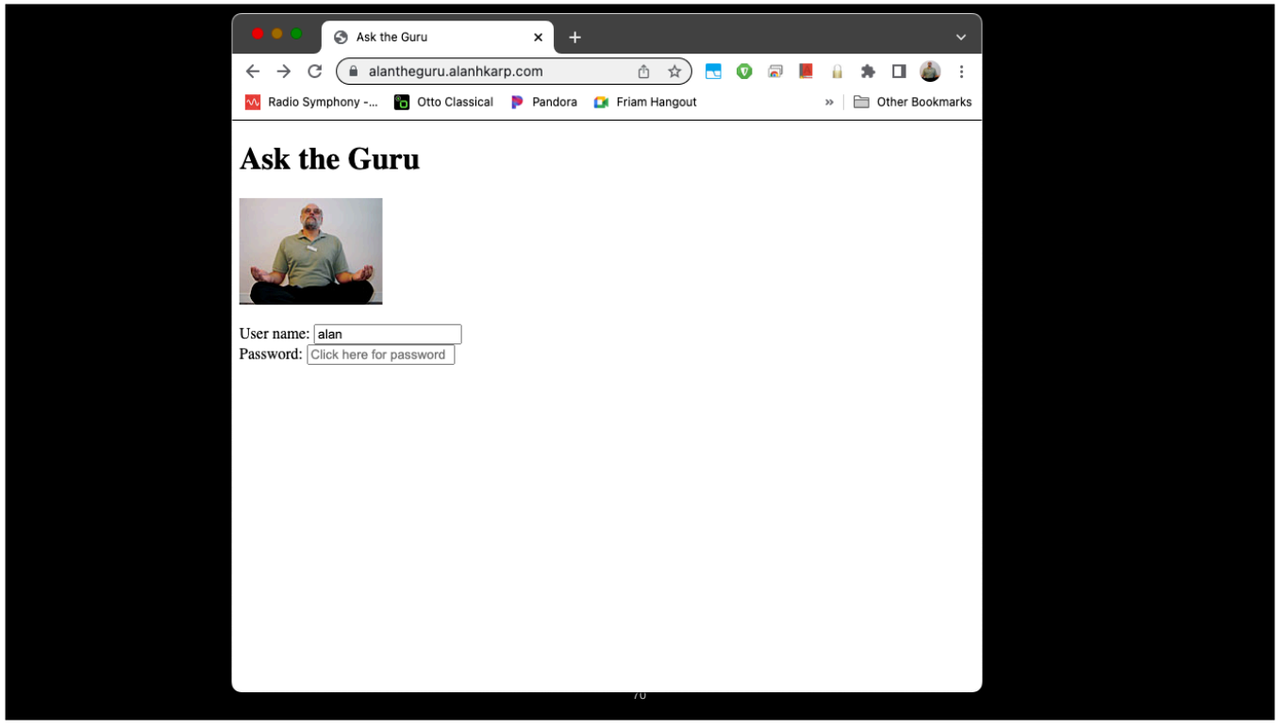
After you install the extension, you will see something like this the first time you visit a login page.



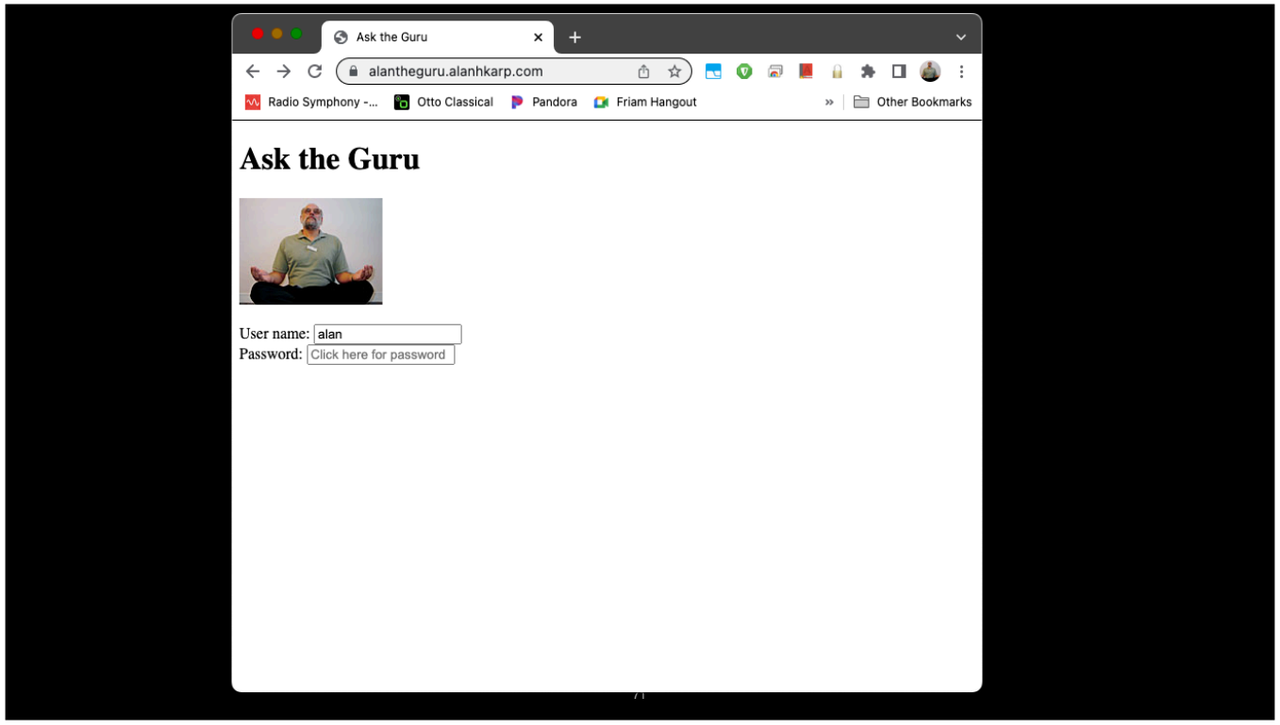
Click on the SitePassword icon, and fill in the form.



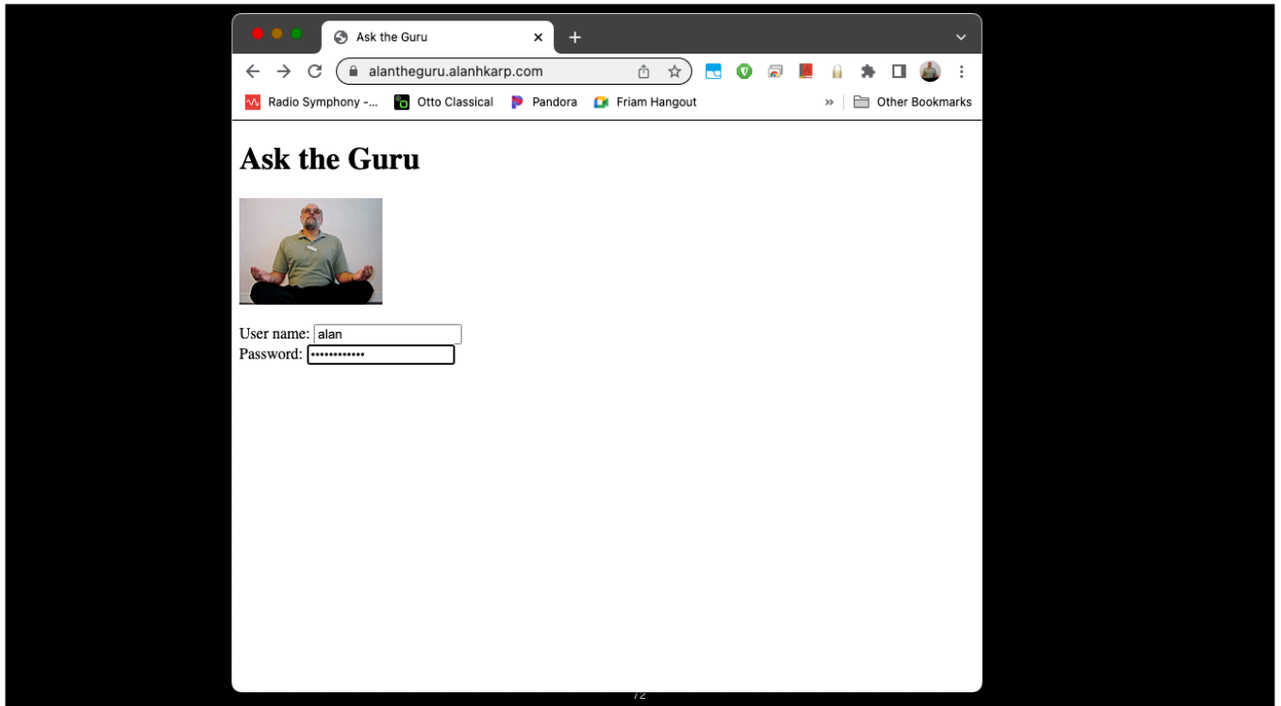
Notice your site password starts with to3X. I'll come back to that later.



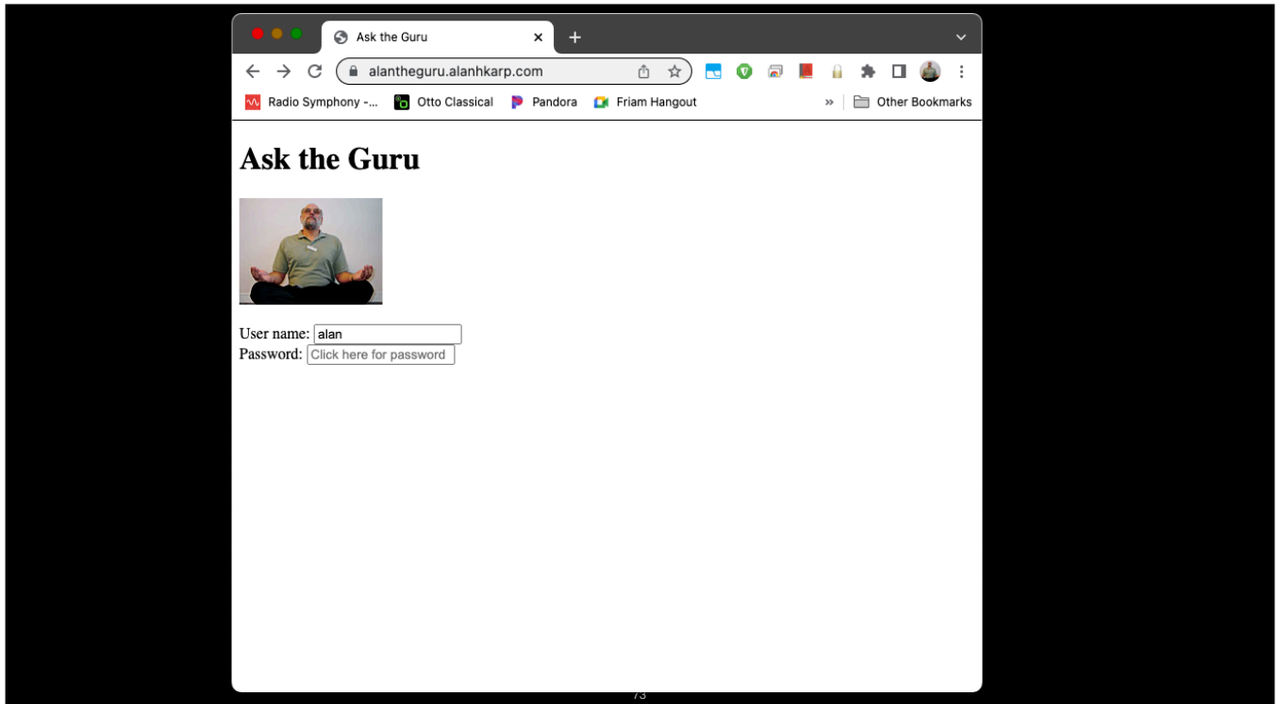
After you mouse out, you see your user name filled in.



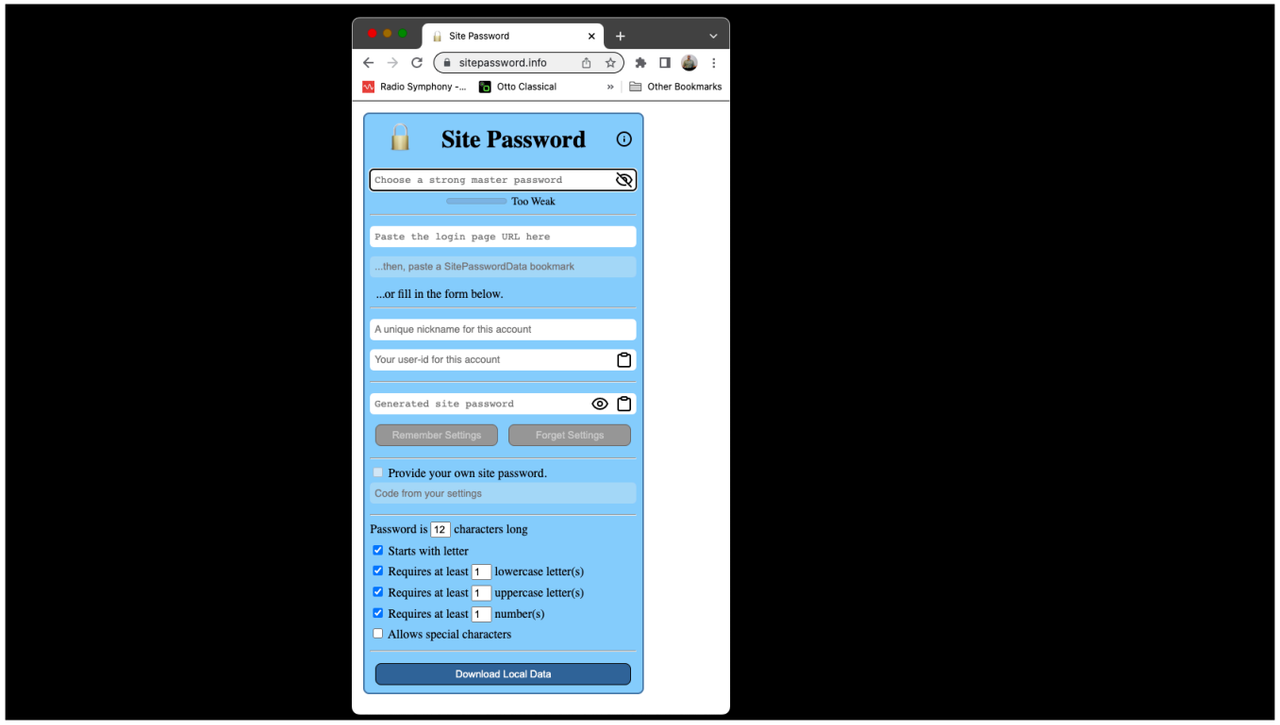
After you mouse out, you see your user name filled in.



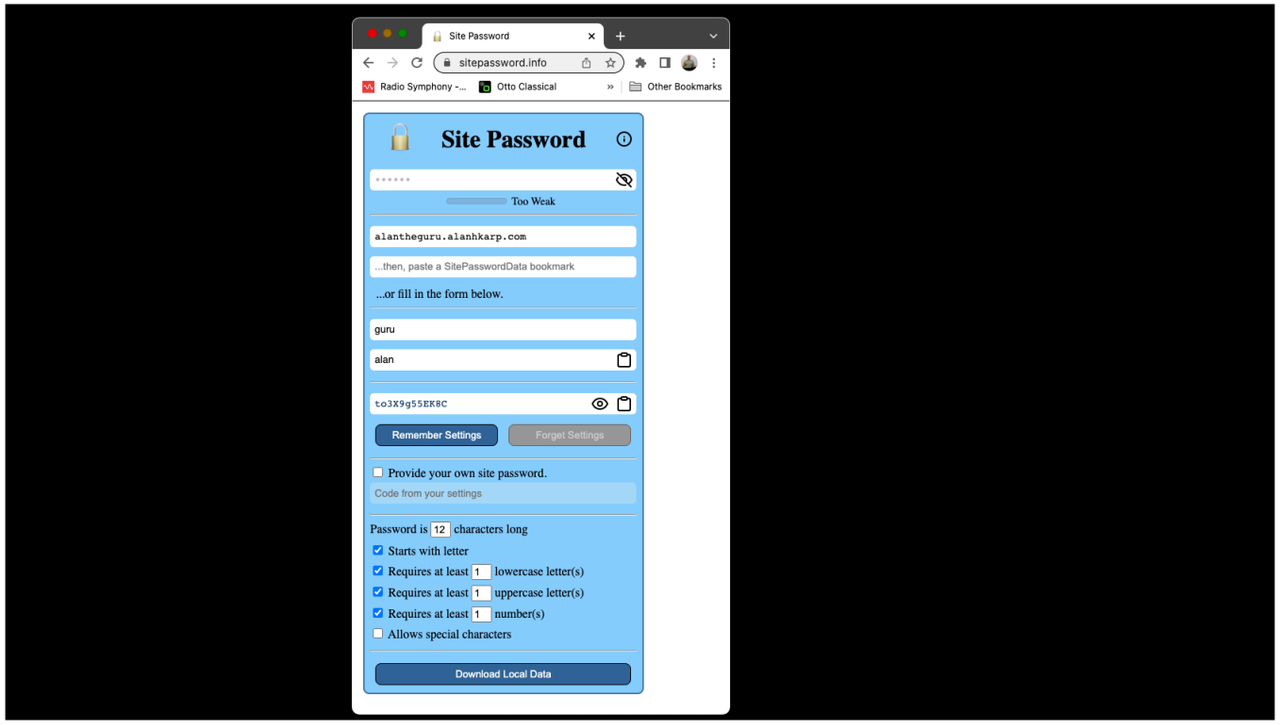
Click on the password field, and you’re ready to log in.



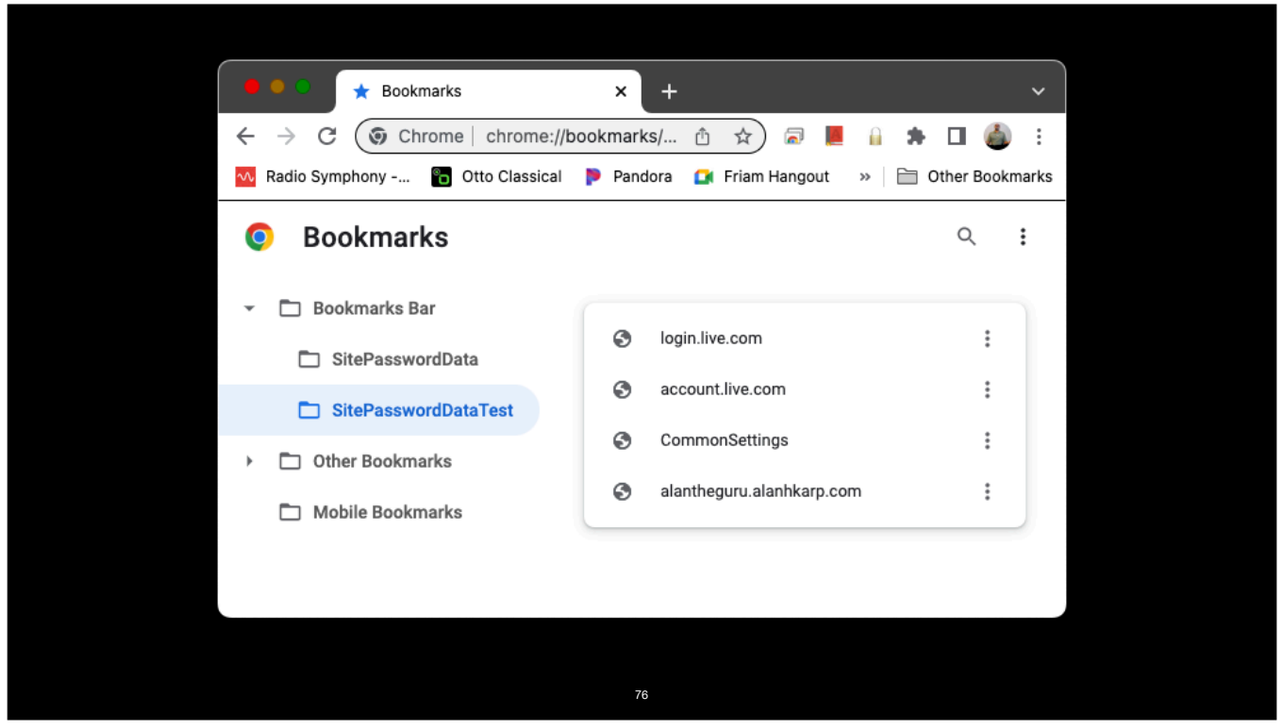
After you mouse out, you see your user name filled in.



What if you’re not at a machine that has the extension installed? There’s a web page for that.

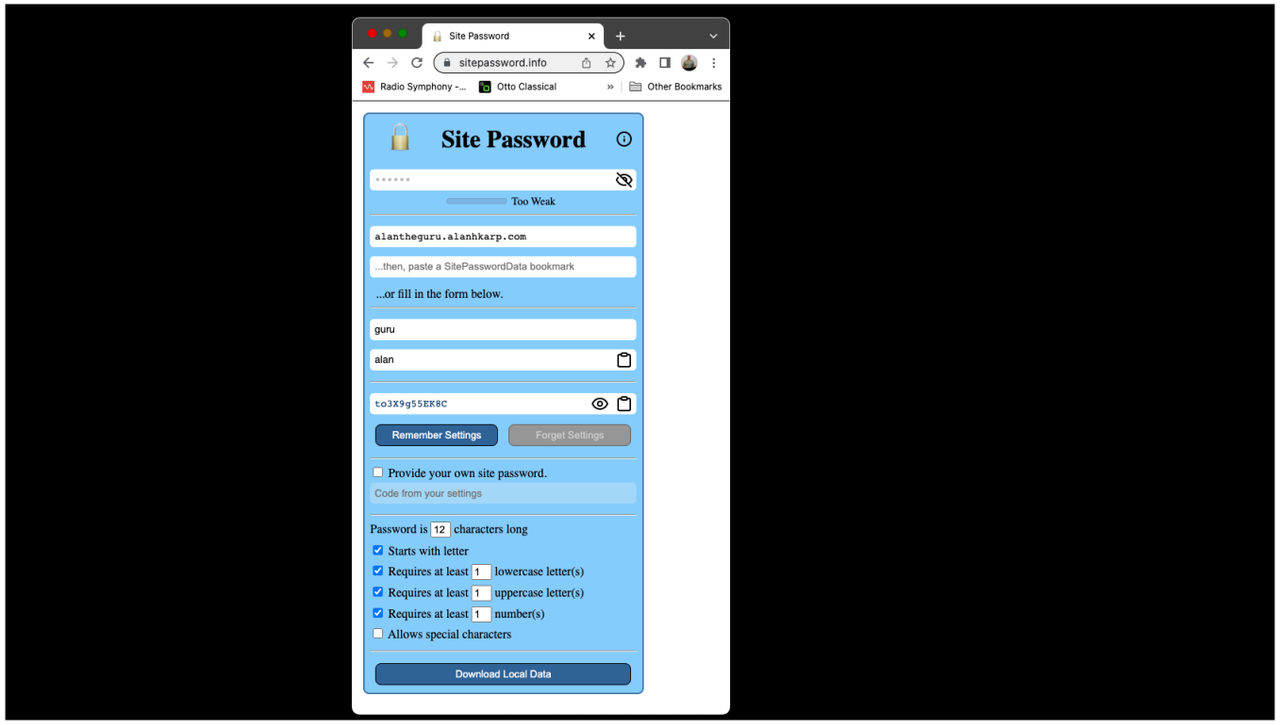


Fill in the form, and you get the same site password.

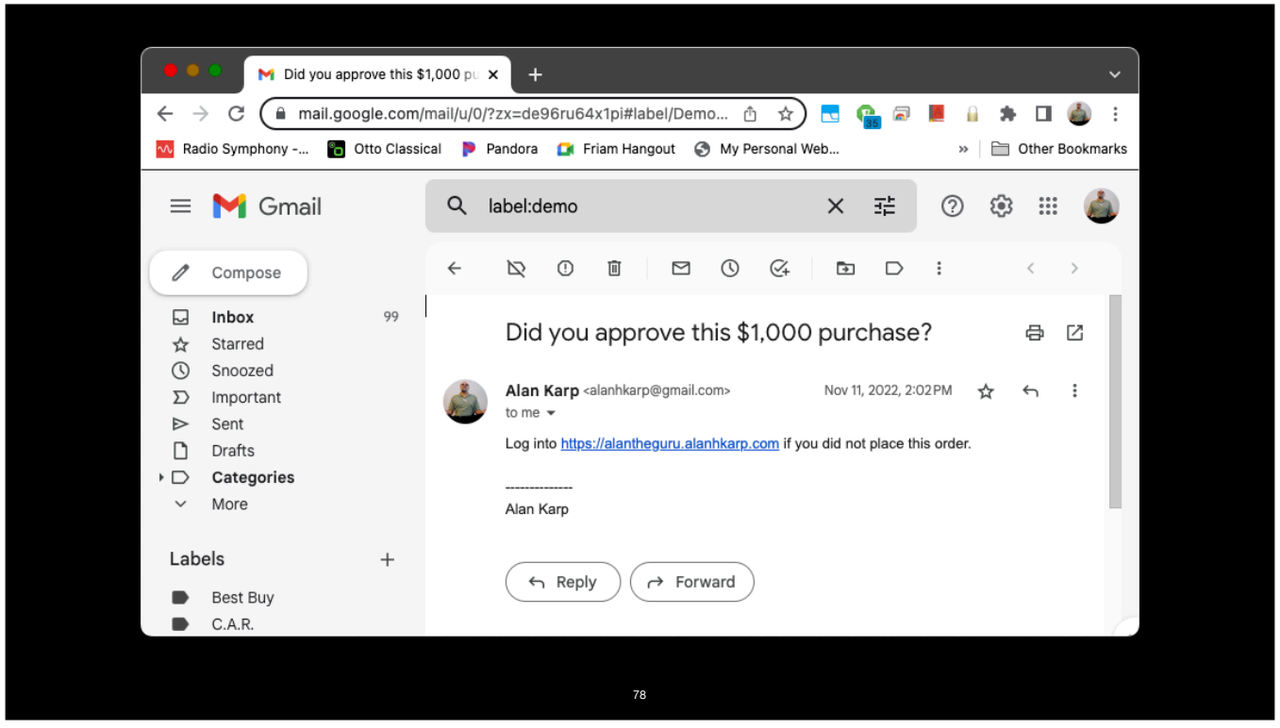


However, if you’re on a device you sync bookmarks with, such as your phone, you can copy the appropriate bookmark.

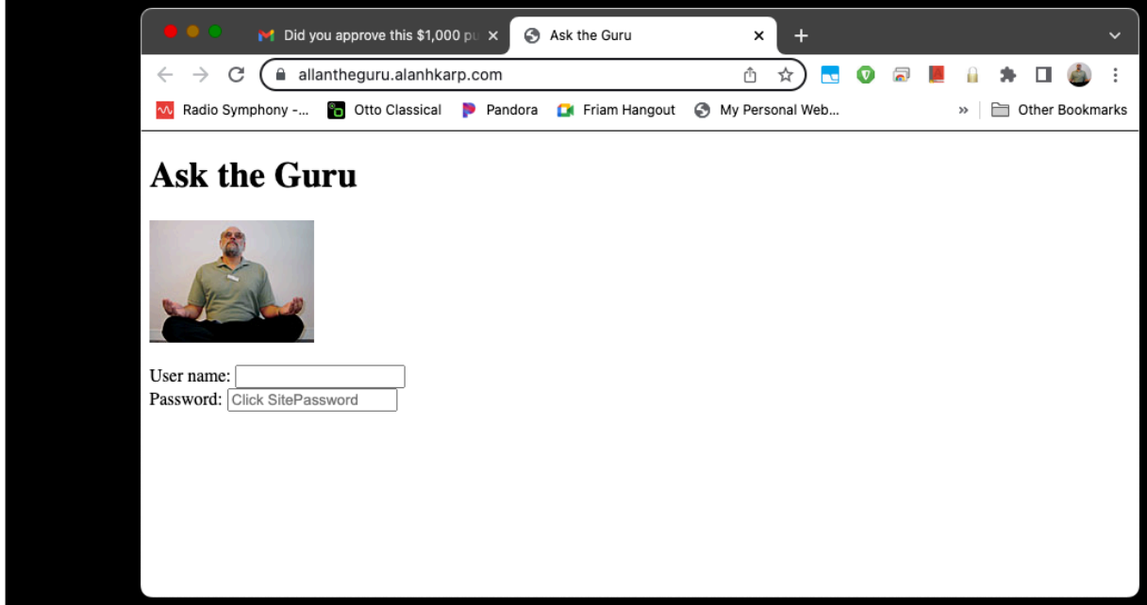
(Notice the two live.com entries. I’ll explain them shortly.)



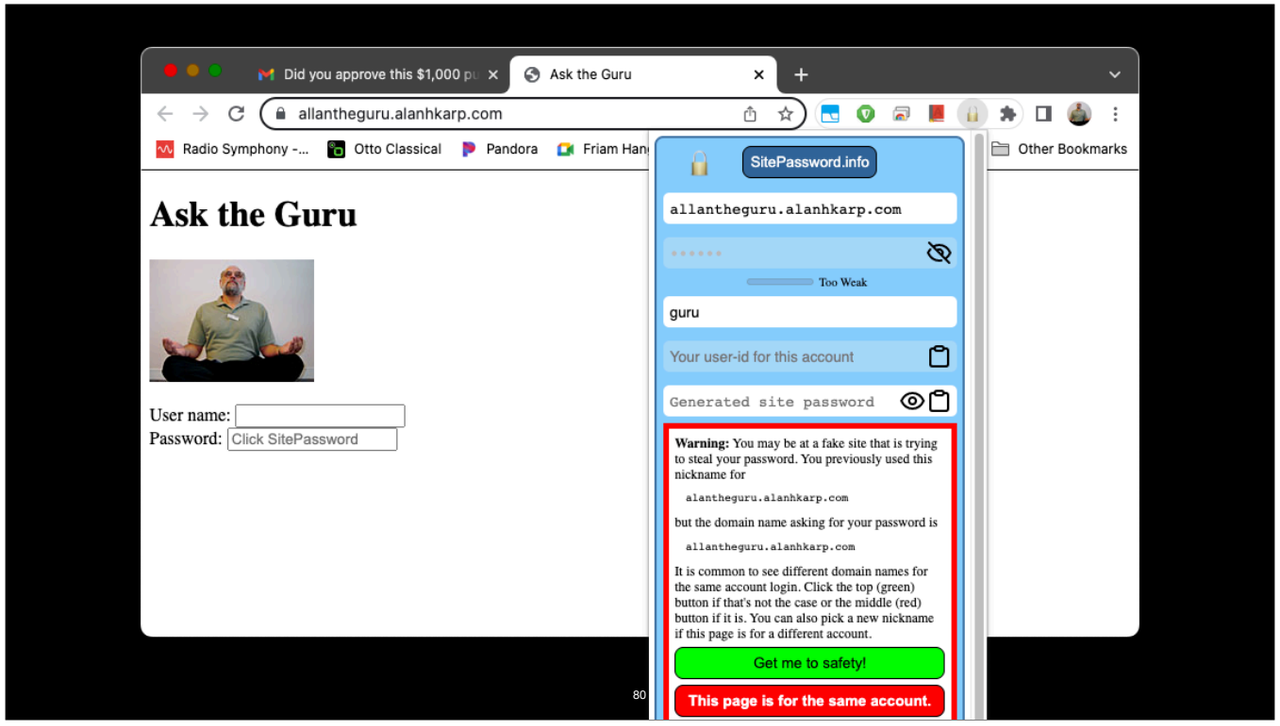
Fill in the form, and you get the same site password.



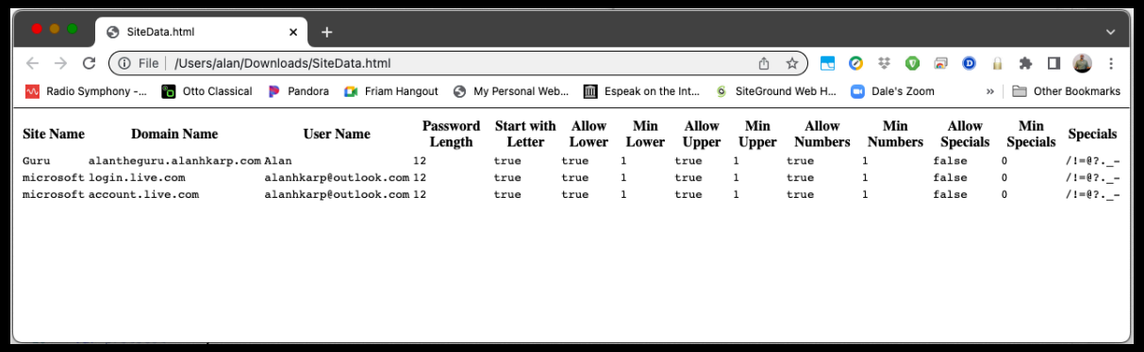
Have you ever gotten an email like this? Let’s say you click on the link.



The first thing you notice is that your user name isn’t filled in.



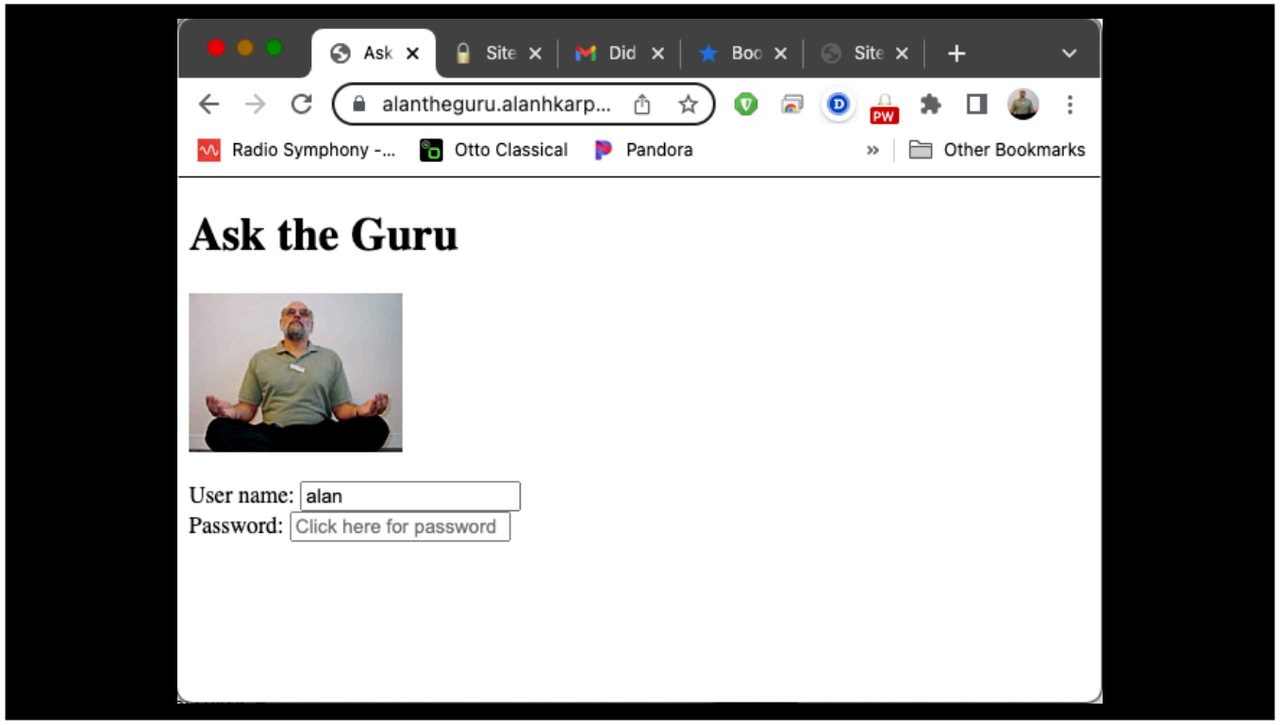
If you open SitePassword and try to fill in the settings, you get this big, scary warning. I'm sure you can't read the warning, but it says that the domain name with 2 ells is not the one you set up for. Clicking on "Get me to safety!" will take you to your home page.



Site Name	Domain Name	User Name	Password Length	Start with Letter	Allow Lower	Min Lower	Allow Upper	Min Upper	Allow Numbers	Min Numbers	Allow Specials	Min Specials	Specials
Guru	alantheguru.alanhkarp.com	Alan	12	true	true	1	true	1	true	1	false	0	/!=@?._-
microsoft	login.live.com	alanhkarp@outlook.com	12	true	true	1	true	1	true	1	false	0	/!=@?._-
microsoft	account.live.com	alanhkarp@outlook.com	12	true	true	1	true	1	true	1	false	0	/!=@?._-

You can now print this page in case you need to retrieve your settings.

Note that the settings are tied to the site name. So any change you make for login.live.com will be applied to account.live.com.



When you put your site password on the clipboard, SitePassword will remind you to remove it.

I don’t know about you, but that red box is annoying enough to me that I do clear the clipboard.

Backup Slides

The Evil Coffee Shop Attack

Autofill with no user action

1. The user sets up several sites with a password manager.
2. User connects to a rogue router in a coffee shop.

   The attacker can inject, block, and change network packets.
3. Attacker directs user's browser to a vulnerable page at the target site.
4. Attacker injects login form into the vulnerable page by modifying packets.
5. Your password manager fills it in.
6. Repeats for another site with a vulnerable page.

I just want to convey the importance of requiring a user action before filling in the password.

The vulnerable page at the victim site doesn't need to be the login page.
The attacker can extract all the user's stored passwords in a few seconds.
The user might notice the address bar changing, but the pages look normal.

How to Inject a Password Field

- HTTP login page - submits forms with HTTPS
  - Doesn’t happen much now but (Schwab, eBay)
- Broken HTTPS
  - Usually an expired certificate (HP)
- HTTPS with active content fetched over HTTP
- XSS (Cross Site Scripting) on any page at the victim site
- Unnecessary dependencies
  - Many use jQuery on their login pages

All the issues on this page allow a bad guy to inject a password field onto the page.

Schwab story: Many years ago, schwab.com had a login form on an HTTP page. I got in touch with the security team, but they kept saying, “But your password is sent over HTTPS so it’s safe.” After a while they got tired of me and transferred me to Identity Theft. I explained the problem to a nice woman who listened politely and then said, “I’m sorry. We only deal with actual cases of identity theft. I don’t know much about security, but that problem sounds really serious. I’m going to escalate it.” Two weeks later the problem was fixed.

eBay story: I was at a conference where one of the speakers was the CSO of eBay. After his talk, I told him that they had an HTTP login page. He said, “Impossible!” To his credit, he got on the phone to double check. He found me at the break and told me I was wrong. I opened my laptop to the eBay login form, and sure enough it was HTTP. He contacted me about a week later. It turns out it was HTTPS, but only inside the eBay firewall.

You’re in trouble any time there’s broken HTTPS or a site fetches content over HTTP. Then there’s the normal kinds of attacks via Cross Site Scripting and clickjacking. Any of these makes it easier for an attacker to get a malicious password form on a page you’re viewing.