Password Managers
Exposing Passwords Everywhere

Marc Blanchou — marc[at]isecpartners[dot]com
Paul Youn — paul[at]isecpartners[dot]com

iSEC Partners, Inc
123 Mission Street, Suite 1020
San Francisco, CA 94105
https://www.isecpartners.com

November 13, 2013

Abstract

Advancements in password cracking and frequent theft of password databases endanger single-factor password authentication systems. Password managers are one of the only tools available that can help users remember unique high-entropy passwords, and other secrets such as credit card numbers, for a large number of applications. Can password managers deliver on security promises, or do they introduce their own security vulnerabilities? This paper examines popular browser-based password managers and presents common security flaws that could be exploited to remotely extract a user's password.

1 Introduction

People regularly use dozens, if not hundreds, of web applications. Savvy users know that the best security practice is to choose unique and complex passwords for every web application. Passwords are chosen to resist both online and offline brute-force attacks that might occur after a password database has been stolen. Offline attacks get better and better as password dictionaries get published[1] (and are used as baseline guesses against passwords) and computing power improves.[2]

Even users who have a system for creating passwords that may be more difficult[3] to guess[4] will have trouble remembering the exact password for a web application that is only rarely used. The solution is some type of password management system.

Password management systems can range from using the integrated browser auto-fill functionality, to a spreadsheet of username/passwords, to a memorized system for modifying passwords between applications, to actual password management software. Actual password management software is becoming increasingly popular because of usability and affordability of the products.

Previous research on password managers has focused on the cryptographic protections of the passwords themselves in particular environments such as mobile devices.[5] This research instead focuses on browser specific integrations

[1] http://www.theregister.co.uk/2010/01/21/lame_passwords_exposed_by_rockyou_hack/
[2] http://arstechnica.com/security/2013/03/how-i-became-a-password-cracker/
[3] http://xkcd.com/936/
[4] https://www.schneier.com/blog/archives/2007/01/choosing_secure.html
[5] http://media.blackhat.com/bh-eu-12/Belenko/bh-eu-12-Belenko-Password_Encryption-Slides.pdf

and mechanisms to remotely compromise credentials. Four of the most popular password managers were examined[6]:

- LastPass Chrome and FireFox Add-On, version 2.0.20: https://lastpass.com/
- OneLastPass Chrome Extension, version 2.6.7: https://www.onelastpass.com/
- 1Password Chrome and FireFox Add-On, version 3.9.19: https://agilebits.com/
- MaskMe Chrome and FireFox Add-On, version 1.27.318: https://www.abine.com/maskme/

Password managers have a difficult goal: provide a password management system that is both easy to use and also protects passwords from unauthorized parties. In the context of a web browser, password managers should make it easy to log into web applications, but also ensure that passwords are only submitted to the intended party.

Making sure that passwords are only sent to the intended party is actually more complicated than it may seem. Password managers must answer difficult questions such as:

- Which login form is correct?
- When should the password be auto-filled?
- Is the password being submitted to the intended party?

This research shows that most password managers made design decisions that greatly increase the chance of users unknowingly exposing their passwords through application-level flaws. Many of the flaws relate to the browser-integrated password managers that don't follow the same-origin policy that is crucial to browser security. In the case of password managers, this means that passwords could be filled into unintended credential forms, making password theft easier.

2 Vulnerabilities in Browser Integration

The most popular password managers have integrated browser extensions or plug-ins that can automatically manage your passwords. The extensions attempt to automatically detect credential fields and fill out detected forms with the appropriate password. If the integration isn't performed properly, passwords could be filled into an attacker-controlled password form or siphoned off to unintended parties.

We tested the above password managers to see if they could properly protect against multiple attacks described below.

2.1 HTTP vs HTTPS

Perhaps the worst type of vulnerability discovered was in the MaskMe password manager. MaskMe failed to distinguish between HTTPS and HTTP schemes, and violated the same-origin policy concept. That means if MaskMe is configured to auto-fill a credential on an HTTPS domain such as https://www.google.com, but encountered a login form on http://www.google.com, the form would still be populated.

A man-in-the-middle attacker, say on a public wireless network, could simply redirect victims to fake HTTP versions of popular websites with login forms and JavaScript that auto-submits after they are automatically filled in by MaskMe. Anyone using MaskMe with auto-fill enabled (this is the default behavior) could very quickly have their passwords stolen by simply connecting to a malicious access point, and victims would never know.

[6] All password managers discussed in this paper have been informed of the discussed weaknesses and were given at least sixty days to address issues prior to the publishing of this whitepaper.

2.2 Cross-Origin Password Submissions

Three browser-based password managers (LastPass, OneLastPass, and MaskMe) were found to submit passwords across origins. In simple terms, that means if a login form is encountered on `https://www.google.com` and sends the password to `https://www.isecpartners.com`, the password manager will fill in the user's `https://www.google.com` credentials and send them to `https://www.isecpartners.com`. If an attacker is able to create a login form on a victim website that redirects credentials to a malicious web server or a compromised application, the attacker could steal a victim's password even when JavaScript code cannot be inserted or executed.[7]

Although the ability to create a malicious login form on someone else's website seems difficult, it could still be done relatively trivially because of additional vulnerabilities that are described in subsequent sections.

2.3 Subdomain Equivalence

OneLastPass, LastPass, MaskMe and 1Password ignored subdomains when comparing origins. That means that a login form encountered on `https://forum.example.com` will still be treated as equivalent to a login form encountered on `https://example.com/log_in` — violating the same-origin policy.[8] Subdomain equivalence is quite dangerous because some subdomains — such as user discussion forums, blogs, or mail subdomains — can often be manipulated by an attacker. For example, a forum that allows for HTML formatted comments could be exploited by an attacker to add a login form on a domain, and thus steal credentials from unsuspecting users. In addition, an application with multiple subdomains is likely to have weaker ones that could be vulnerable to Cross-Site Scripting (XSS) attacks — and could effectively allow an attacker to retrieve credentials for the parent domain when the password is auto-filled on a fake login form.

2.4 Which Login Page?

None of the examined password managers appear to verify the login page for a remembered password on a given domain. For example, although Vimeo's login page is hosted at `https://vimeo.com/log_in`, all of the examined password managers will detect login forms anywhere on the `https://vimeo.com/` domain.[9] That means that if an attacker is able to inject a login form anywhere on the Vimeo domain, a victim's credentials could be stolen.

2.5 Amplifying Risk: Auto-Fill and Auto-Submit

In order to make password managers even more usable, LastPass and MaskMe can be configured to auto-fill a user's credentials into an encountered login form. LastPass also allows users to configure the manager to auto-submit credentials. Due to the identified issues, auto-fill and auto-submit functionality increase the risk of a victim leaking passwords, because a login form could be hidden by an attacker within an expected form. If a user submitted the expected form, they would be unaware that their password had also been filled into hidden form fields and submitted to the attacker.

2.6 Putting it Together: Stealing Passwords

Because of subdomain equivalence, it would be relatively easy for an attacker to inject a phishing login form into any popular domain. In fact, many domains explicitly allow any user to create HTML content that is then rendered;

[7] It should be noted that a script can retrieve and exfiltrate any data auto-filled on a page
[8] Browsers treat these as separate domains and limit the interaction allowed between the two subdomains.
[9] This behavior is also true with password managers built into modern browsers — see Section 2.7 on page 5.

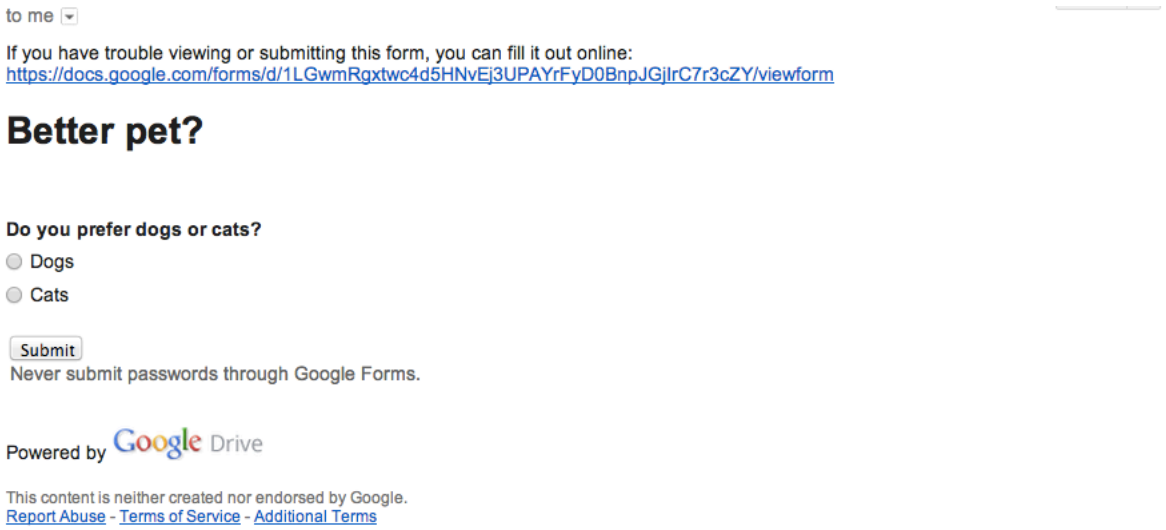
for example, wiki pages, forums, and perhaps most terrifying: most web-based email clients that render arbitrary HTML-formatted email.

We tested a password field containing phishing email on three popular webmail providers: https://mail.live.com, https://mail.google.com, and https://mail.yahoo.com. The following proof of concept was sent as an HTML-formatted email:

```
<html><body>Thanks for taking our Survey!
<form action="https://www.isecpartners.com" method="post" style="font-size:medium;
    margin:0px;font-family:Times;border:0px;padding:0px" target="_blank">
  Do you like cats?: <input type="text" name="cats"><br>
  Do you like dogs?: <input type="text" name="dogs"><br>
<input type="email" name="Email" value="" style="max-height:0px;padding:0px;border
    -width:0px;width:0px">
<input type="password" name="Passwd" style="max-height:0px;padding:0px;border-
    width:0px;width:0px">
<input type="submit" name="signIn" value="Submit"></form></body></html>
```

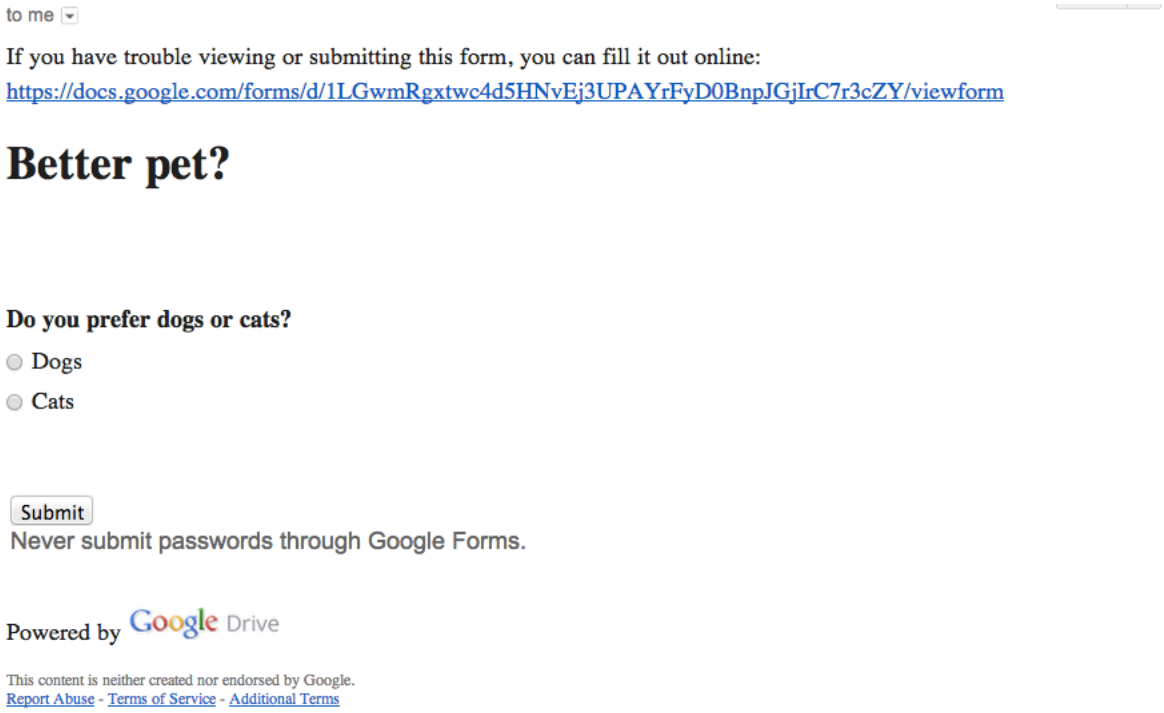
Yahoo! Mail users running LastPass are the most vulnerable to credential theft. Any Yahoo! Mail user who has LastPass with auto-login enabled for the yahoo.com domain and views emails over HTTPS could have their username/password stolen just by opening the phishing email. When the email opens, LastPass will automatically "log in" and send the credentials to https://www.isecpartners.com. If a user only has "auto-fill" enabled, the credentials will still be stolen if the survey is submitted.

Gmail users are a bit better off, because Google will warn you that a form is about to be submitted before fulfilling the request, even if LastPass auto-login functionality is enabled. For Gmail users, a victim would still be vulnerable if they actually respond to the survey and have auto-fill enabled, or if they have auto-login enabled and click through the warning. Many victims will unwittingly submit their username and password to https://www.isecpartners.com. To give an idea of how successful a phishing campaign may be, compare the two screenshots of survey emails sent to a Gmail address[10]:



The above graphic is a legitimate survey that anyone can create and send via email. Below is a malicious form that will steal a person's password if they have LastPass with autofill enabled:

[10]The pictured survey was customized to look like a standard Google Drive form and differs from the proof-of-concept HTML above.


to me

If you have trouble viewing or submitting this form, you can fill it out online:
https://docs.google.com/forms/d/1LGwmRgxtwc4d5HNvEj3UPAYrFyD0BnpJGjIrC7r3cZY/viewform

Better pet?

Do you prefer dogs or cats?

- Dogs
- Cats

Submit

Never submit passwords through Google Forms.

Powered by Google Drive

This content is neither created nor endorsed by Google.
Report Abuse - Terms of Service - Additional Terms

Note that there is no obvious login form in the above screenshot,[11] but any victim who clicks "Submit" and was using a password manager that auto-filled in their credentials would send their username and password to `https://isecpartners.com`.[12]

Outlook.com (`https://mail.live.com`) users were best off because the mail application uses mitigations to prevent cross-origin submissions of any kind to prevent this attack.

2.7 How password manager extensions compare with modern browsers

We determined that browser auto-fill mechanisms were far more secure than the extensions tested. Both Firefox and Chrome respect the same-origin policy when filling passwords and do not auto-fill passwords when the URI scheme (http/https) or subdomains of the form target differs from the current page. In addition, browsers refuse to auto-fill passwords when a login form is sent to a domain different than the domain it is displayed on. However passwords are auto-filled on any page of a web application as long as it is within the same domain.

3 Native Application Flaws

Native application password managers can also be attacked just like any other software. We examined one such application: 1Password. 1Password performed automatic updates in an insecure manner by reaching out to an unprotected endpoint: `http://updates.agilebits.com/check?...`. If an update was discovered, the software would be automatically installed using admin privileges.

[11] The warning about submitting passwords is inserted by default into every Google Drive form and is unrelated to the described attack.

[12] Although Google does warn that a form is about to be submitted, the warning appears for any in-line form submission. For example, this warning will appear when a Google docs-based survey is filled out in an email. A user who is expecting to be submitting a survey about pets will likely click through the warning.

Because the update was performed over HTTP, a man-in-the-middle attacker could purport to be the legitimate update server and serve the 1Password application an arbitrary piece of malware that would be installed with administrator privileges and completely compromise the victim's machine.

Note that AgileBits has reportedly patched this vulnerability.

4 Conclusions

Password managers can still be a huge asset to users when used properly. Unfortunately, it appears that many popular password managers are insecure by default, but there are simple actions that users can take to safely use a password manager. There are also fairly simple improvements that password managers could introduce which would help improve their security.

4.1 Recommendations for Users

Most of the tested password managers are designed to detect login credential forms. Although auto-fill and auto-login functionality can make password managers more user friendly, those features greatly increase the risk of password theft using techniques described above. iSEC highly recommends disabling any auto-fill or auto-submit functionality in password managers. Without auto-fill or auto-submit functionality enabled, users will have to manually indicate that the password form should be filled with the saved credentials and a phishing attack such as the one described in section 2.6 will be much more difficult to mount.

Other general recommendations that are not specific to this research include:

- Use the password manager to generate a random password instead of picking one yourself if possible. Random passwords are much more difficult to guess, and one of the benefits of a password manager is that you don't have to memorize it.
- Register a unique password for every site so that one password compromise will not affect others. Password managers are designed to make this easy to do.
- Only submit passwords on pages that are entirely HTTPS.
- Choose a strong master password to protect your individual passwords as it could still be potentially brute-forced on a stolen device.

Although imperfect, a properly used password manager can still have a large positive impact on an individual's security.

4.2 Recommendations for Password Manager Software Developers

Password managers have some serious weaknesses that can make it easy for an attacker to remotely steal a user's password. Password managers configured to auto-fill login forms can be exploited through a simple survey-based phishing attack that a victim views through a web browser. The MaskMe password manager could be exploited directly with a simple network attack to harvest a large number of a victim's credentials at once.

We highly recommend that password managers respect the same-origin policy concept, specifically in regards to subdomains and protocol scheme. Additionally, users should be given the option of configuring cross-origin credential submission and it should be disabled by default.