Google Chrome Browser Extensions: Another Malicious Portal In

Olivia MacDougal

13 December 2017

Abstract

Since the time that Chrome developers first implemented browser extensions in 2009 up to present day, Google Chrome browser extensions continue to be a security threat vector with very real consequences for unsuspecting users. Because of lax and ineffective Google Chrome policies, very little progress has been made in preventing the effect of malicious browser extensions. The focus of this paper is to identify how hackers are exploiting Chrome extensions, discuss the issues surrounding the prevention of malicious extensions and analyze the effectiveness of the current prevention measures. To aid in the reader's understanding of these issues, I will also present an example browser extension, which though seemingly harmless at first, is in fact malicious. Finally, this paper provides a series of best practices for developers and Chrome users alike in regards to browser extensions.

Contents

1 Introduction

In the 2016 Pwn2Own hacking competition, Google Chrome fared best out of all of the major web browsers in a series of cybersecurity attacks to find security vulnerabilities, solidifying the browser's established reputation as being the most secure web browser [Pro16]. Yet there is a component in the Chrome web browser that hackers are capitalizing on: browser extensions. Browser extensions are small programs that add new features to extend a web browser, modify web pages, personalize a user's browsing experience and integrate the user's browser with the other services they use when surfing the web. However, extensions can be exploited to carry out malicious actions on a user's computer without their consent. The lack of security in Chrome extensions is a glaring security threat vector. Not only have many actually malicious extensions been uploaded into the Chrome Web Store, many of these extensions have been hijacked since due to Google's relatively lax policy regarding browser extensions. Every few months some new malicious browser extension is exposed or exploited and yet, very little is being done to remedy this issue. Users need to be educated about exactly what is happening when they download browser extensions in Chrome and instructed on how best to protect themselves to avoid exploitation.

2 To the Community

In a 2015 research paper written by Google software engineers about malicious browser extensions, the authors reported that between 2012 and 2015, they had "detected 9,523 malicious extensions: nearly 10% of every extension submitted to the store. Despite a short window of operation—we removed 50% of malware within 25 minutes of creation— a handful of under 100 extensions escaped immediate detection and infected over 50 million Chrome users. . . ." Their evaluation covered roughly 100,000 unique extensions submitted to the Chrome Web Store and they found nearly one in ten to be malicious [Jag15].

There is no shortage of reports about recent malicious extensions, ranging from extensions that steal all posted data (Catch-All) [cat], steal bank data in Brazil [Mar], or mine for cryptocurrency (Shorten-URL) [Mem17]. One of the most recent, widespread examples occurred in October 2017, when over 37,000 Chrome users downloaded a fake version of the popular extension AdBlock Plus. The fake version forcibly opened new tabs to show ads to users, and though it is no longer on the web store, Google is still working on removing the phony extension off users' devices [Har17] [@Sw]. This is all to say that browser extensions have been and continue to be a major security threat vector in the Google Chrome browser. Moreover, most users do not realize what they are and how powerful they can be. The intent of this paper is to bring greater awareness of this issue.

Personally, this topic also holds personal significance for me. I have been a long-time Chrome user and have encouraged family and friends to switch to it as it has been shown to be very secure and reliable. Recently, when I went home for the holidays, my mother complained about strange behavior with her browser and when I inspected her Chrome browser I found a multitude of browser extensions that had changed her default search bar and altered the look of the new tab page, among other things. When I asked her why she had installed these extensions, she told me she did not remember installing them - a clear example of social engineering and user ignorance.

3 Background

3.1 Browser Extensions 101

Browser extensions are software programs that can modify or enhance the functionality of a web browser [def]. Although they are often very small in size with minimal user-interface, they can be incredibly powerful, often providing useful and desirable tools for a user. Typically they are written using HTML, CSS, and/or Javascript. In the Chrome browser, extensions are downloaded from the Chrome Web Store and are updated automatically. They can be enabled, disabled, deleted and generally accessed from the chrome://extensions page. When a browser extension is installed, the extension icon will automatically appear next to the Chrome omnibox unless a user chooses

to hide it; this functionality was implemented by Chrome developers to increase the visibility of installed extensions on the browser.

At its core, a browser extension is simply a manifest file, `manifest.json`, which is a metadata file in JSON format that contains all relevant properties of an extension [Kja16]. The manifest file is essentially the declaration of what the extension does and the related permissions that it requires to function. Permissions are arguably the most important aspect of Chrome browser extensions in terms of security threat vectors. As most browser extensions add non-trivial functionalities, many require high levels of permissions that users are prompted to agree to only on installation. Though extensions do update automatically, if the permissions change, users must agree to the new permissions before the update is installed. Overall, browser extensions are small programs, which often require additional permissions, installed by users, that are added to the Chrome browser to add functionality.

3.2 Google Chrome Policies, Effectiveness and Failures

On May 27, 2014, Google Chrome began blocking local browser extensions [blo14]; that is, they began enforcing a policy that dictates that extensions can be installed only if they are hosted on the Chrome Web Store. This announcement came after many Windows users were infected with malicious browser extensions downloaded from sources all over the web. Though this was definitely a step in the right direction as far as preventative measures go, Chrome does not do enough to protect its users from malicious extensions. The process of putting a browser extension onto the Google Web Store is as simple as uploading a compressed version of the files after paying a $5 registration fee to become a Google Developer. The fee according to the website is "to prevent fraud in the Chrome Web Store."

After publishing a browser extension, it takes a few hours for it to "go through an automated review process and in most cases, [the extension] will be published without further manual review." [faq] The automated review process does include a scan for malicious code, and the extension might be flagged and reviewed by a Google developer if it violates any part of Google's Developer Program Policies [pol]. This review process is easily avoided: since the extension cannot ship with the payload if it wants to pass through the security checks, the extension simply needs to download it after having been installed [Kja16]. In a few lines of Javascript and the right permissions, any extension can be malicious. With the volume of apps, themes, and extensions uploaded to the Chrome Web Store every day, the automated review process makes some sense; however, like many other automated processes (see the malware-ridden Android app store), their process for reviewing extensions leaves users at the mercy of hackers.

3.3 Social Engineering and Phishing

An average user will not just find and download random malicious extensions on the Chrome Web Store. Most users affected by malicious browser extensions are victims of phishing attacks or instances of social engineering. For instance, one hacker used a link on Facebook that directed users to a fake Youtube video [Mer14]. After playing a few seconds of the video, a pop-up appeared that insisted users install a browser extension to be able to continue using the fake Youtube and viewing the video. Once installed, the browser extension could post statuses and comments on the user's Facebook account, as well as send messages and links via the chat function, thereby continuing the spread of the extension. As users are highly susceptible to these kinds of attacks and cannot be expected to always be able to identify them, the onus should be on Google Chrome to protect their users by not allowing extensions onto the Web Store in the first place.

4 Exploiting Chrome Browser Extensions

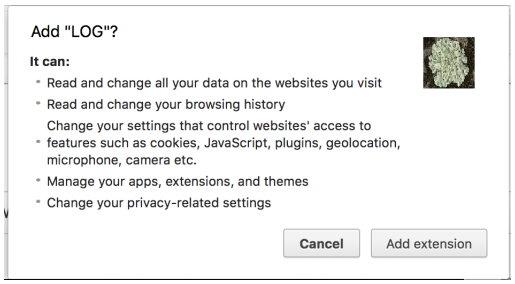
4.1 Lichens Overthrowing Google (LOG)

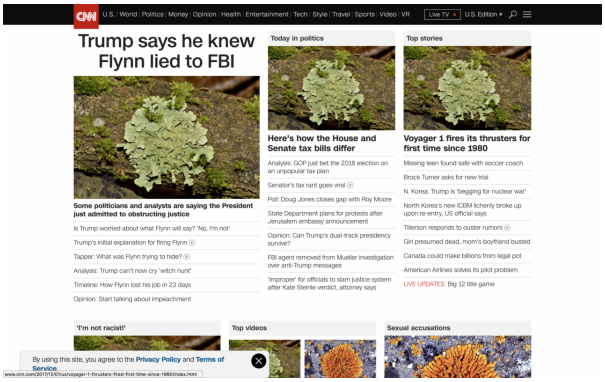
While doing researching for this paper, I read a lot of Google Chrome policies and many articles with examples of malicious browser extensions, and kept asking myself the question, "How do they all get in there in the first place?" To really understand the process of uploading a browser extension to the Chrome Web Store, I decided to make one myself.

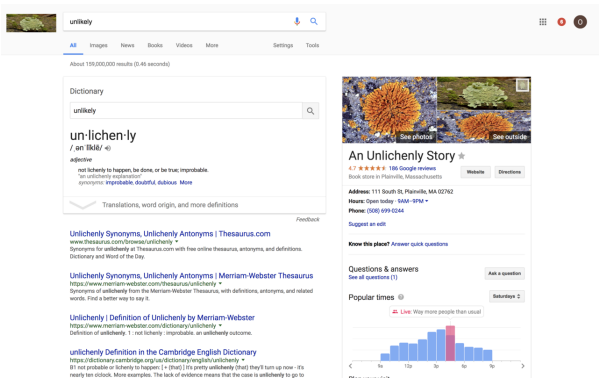
Following a few online tutorials, including the Google Chrome developer tutorial [**?**], I had a working extension up and running in fifteen minutes or less. The example that the Chrome developer tutorial walks through is used to change the background color of the active tab open in Chrome. For my purposes, this was too insignificant and did not require any high level permissions. Instead, my browser extension was going to replace all photos on any tab with photos of lichen and change any and all instances of the word "like" to lichen. Though a bit comical, the choice of the subject matter was deliberate. Similar to browser extensions in the Chrome environment, "lichens are the most overlooked of the conspicuous organisms in the natural landscape." [Sha] Thus, in a little under two hours, I created the browser extension Lichens Overthrowing Google, LOG.

Not only does LOG require the permission to read and write all data on every single page a user visits (which is in fact necessary for the extension to be able to replace words and pictures), I also added a bunch of other permissions that a user must agree to only once when they add my extension to their browser. Though none of these permissions are actually used, I wanted to see if the Chrome Web Store would reject my application because I required too many superfluous permissions or at least alert me that I was requesting too many permissions.

Beyond high permissions, I wanted to create an extension that does something nefarious - that is, something that the user did not explicitly agree to nor necessarily want to happen. After a few annoying minutes with LOG installed, inevitably the user will want to rid themselves of all the lichen, because it literally replaces all images with one of three pictures of lichen. When a user clicks on the LOG icon located near the omnibox, a plain HTML page pops up in a new tap. When the user clicks the button "Disable LOG", to the user, it seems like nothing happens, and in fact, all of the front-facing functionality is still in-place. However, the user has just unwillingly and unknowingly erased all cookies associated with the websites they visited while LOG was installed and enabled.







The process of uploading LOG to the Chrome Web Store was unbelievably simple. After logging in with my regular Google account and paying the $5 membership fee, I simply had to upload a zip file of the extension along with some screenshots and fill out a few trivial questions about the purpose of my extension. It did not show up immediately, and I was worried that the automated security process would alert me of some issue with my extension, but within a few hours, my extension was live on the Chrome Web Store with no correspondence from the Web Store. Literally as easy as uploading a zip file.

In reality, replacing images and words, secretly wiping some cookies, and requesting high levels of permissions does not make this extension incredibly malicious. Creating this extension took a little over two hours and most of that time was spent reading tutorials, taking screenshots and filling out the Google Web Store form. The point of this whole activity was to show that anyone can upload a malicious browser extension and that it requires very little work or knowledge. Imagine that rather than replacing all images with pictures of lichen, all or some of your images on all or specific pages the extension displayed targeted ads for a specific product or company. Now remember all of those permissions you, the user, agreed to and imagine the extension collecting your browsing history and cookie information and selling those to third-party buyers thirsty for data about you. Possibly even more damning, the creator of the extension can change and update the extension as much as they want without you ever being aware of it. All that is necessary is for you to fall into a social-phishing trap and download LOG.

Currently LOG has one user and is still active on the Chrome Web Store as of 10 December 2017. The extension itself can be downloaded from Chrome Web Store at https://chrome.google.com/webstore/detail/log/ipnjofbojbpifmbajkmpbkiegcpgfogg. The extension source code can be accessed on Github at https://github.com/olivaceously/LOG.

4.2 Methods of Exploitation

Following the examples from above, there is no one definitive way that malicious extensions are exploited because they have been used for many different and diverse purposes, though there are some commonalities between them. Most malicious browser extensions have high permissions and run some Javascript file to download a malicious payload, inject content into a user's browser tab, or do other nefarious activity. Most browser extensions get downloaded as a result of social engineering or phishing scams.

5 Action Items

5.1 For Chrome Users

Though the following action items might seem a bit intuitive or general, they should be effective in helping Chrome users avoid malicious browser extensions in the future.

1. Only install extensions that you have personally sought out on the Chrome Web Store. Do research about the extension by reading reviews and the developer's profile. Extensions should be well-reviewed, with many downloads and plenty of screenshots.

2. If a website or link directs you to the Chrome Web Store, insisting that you must download an extension before you can view content or perform some action, do not download the extension and navigate away from the origin website. It is extremely likely that the extension is malicious.

3. Before agreeing to install an extension, read all of the permissions in the pop-up that appears when you are asked if you want to add the extension to Chrome. Ask yourself if the permissions seem reasonable. For example, should a browser extension claiming to be a calculator be able to read and write all the data on your webpage? (The answer is obviously no; this is was a malicious extension that actually existed).

4. Frequently navigate to the chrome://extensions page and verify that you still need all of the installed browser extensions and uninstall any extensions you don't remember explicitly downloading or no longer are using.

5.2 For Developers

As a browser extension developer, the most important "best practice" is to not request too many permissions and to follow general best coding practices.

Google Chrome developers have some major improvements they need to make to the Chrome Web Store and the process of allowing browser extensions online before the it can be considered truly safe. One of the most notable areas of improvement I discovered while writing LOG was how Chrome deals with extension permissions. Chrome permissions are divided into three different potential risk levels: high, medium, and low alert. The highest alert is for permissions such that the extension can access "**all data on your computer and the websites you visit**, it means that the app or extension can access almost anything. This could be your webcam or personal files, inside or outside of your browser." [per] The Chrome Web Store documentation states that a warning pops-up when a user attempts to install a browser extension displaying one of these alerts to help prevent users from agreeing to install "high alert" extensions, but this is false. When a user tries to install an extension from the Chrome Web Store, these alerts and warnings are nowhere to be found: a pop-up appears confirming that the user wants to add the extension with a list of what the extension can do, that is, the permissions that are required. This distinction is important. The average user will agree without a second thought, often without reading the permissions at all. If the user were to read the permissions, they may not realize the extent of what they are agreeing to.

If Chrome developers were to add information about these permissions using the alert system they already supposedly have in place, or require users to be more cognizant of what they are agreeing to, developers could help protect users from themselves. Even better, Chrome does not allow for any granularity in required permissions for extension developers essentially using an all-or-nothing approach. Nonetheless, as a browser extension developer, the most important "best practice" is to not request too many permissions and to follow general best coding practices.

6 Conclusion

Malicious browser extensions will continue to be another portal in for hackers in the Google Chrome browser until more attention is paid to the platform and relevant policies. The biggest weakness at this point, which Apple's App Store has managed to mostly avoid, is the amount of malicious browser extensions that get uploaded to the Chrome Web Store as a result of the relatively lax requirements. They have been an issue since extensions were first added to the browser in 2009 and they will continue to be, with ever great consquences into the future unless users and Google developers alike take preventative and offensive measures.

References

[blo14] Protecting chrome users from malicious extensions. https://chrome.googleblog.com/2014/05/protecting-chrome-users-from-malicious.html, 2014.

[cat] "catch-all" google chrome malicious extension steals all posted data. https://isc.sans.edu/forums/diary/CatchAll+Google+Chrome+Malicious+Extension+Steals+All+Posted+Data/22976/https:/threatpost.com/malicious-chrome-extension-steals-data-posted-to-any-website/128680/.

[def] What is a browser extension? https://developer.chrome.com/extensions.

[faq] Faq: Is there an approval process for apps in the store? https://developer.chrome.com/webstore/faq#faq-gen-08.

[Har17] Tim Hardwick. Fake chrome web browser extension unwittingly installed by 37,000 users. https://www.macrumors.com/2017/10/10/fake-chrome-extension-google-web-store/, 2017.

[Jag15] Nav Jagpal. Trends and lessons from three years fighting malicious extensions. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43824.pdf, 2015.

[Kja16] Maxime Kjaer. Malware in the browser: how you might get hacked by a chrome extension. https://kjaer.io/extension-malware/, July 2016.

[Mar] Renato Marinho. https://isc.sans.edu/forums/diary/BankerGoogleChromeExtensiontargetingBrazil/22722/.

[Mem17] Francisco Memoria. Google chrome extension caught mining monero via user's cpu. https://www.cryptocoinsnews.com/google-chrome-extension-caught-silently-mining-monero-with-visitor-cpu-power/, October 2017.

[Mer14] Fernando Mercês. Uncovering malicious browser extensions in chrome web store. http://blog.trendmicro.com/trendlabs-security-intelligence/uncovering-malicious-browser-extensions-in-chrome-web-store/, September 2014.

[per] Permissions requested by apps and extensions. https://support.google.com/chrome_webstore/answer/186213?hl=en&rd=1.

[pol] Developer program policies. https://developer.chrome.com/webstore/program_policies.

[Pro16] Emil Protalinski. Pwn2own 2016: Chrome, edge, and safari hacked, 460,000 awarded in total. https://venturebeat.com/2016/03/18/pwn2own-2016-chrome-edge-and-safari-hacked-460k-awarded-in-total/, 2016.

[Sha] Stephen Sharnoff. Lichens of north america. http://www.lichen.com.

[@Sw] @SwiftOnSecurity. Google allows 37,000 chrome users to be tricked with a fake extension by fraudulent developer who clones popular name and spams keywords. https://twitter.com/SwiftOnSecurity/status/917446126382526464.