

Forum:[Hack Forums](#)

Thread: [\[Guide\] How To Make Sure Your Chrome Extensions Are Secure](#)

King of Hearts - August 11, 2015, 10:34 am

Hello,

A couple of days ago, I decided to code a Google Chrome [extension](#) for [HF Live](#). Having never coded an extension before, I was a bit surprised with how easy it is to get one up in the Chrome store. To upload an extension to the Chrome store, all you need is a gmail account and \$5. The reason I'm telling you this is to illustrate the ease of which one can upload an illegitimate product on the app store. One can easily upload an extension that listens to all the traffic and commit "affiliate fraud, credential theft, advertising fraud and social network abuse." Extensions make it extremely easy to inject javascript into pages and one can easily add code that listens to the sign in button on the yahoo mail page and upon clicking that button, the user is logged in to their yahoo mail but unknown to him, his info is also sent to the hacker's server.

Today, I will be showing you how to easily check if the extensions you are using are backdoored.

Before I do, I would like to explain a little bit about how chrome extensions work.

Chrome extensions have three places where javascript that can be injected.

Background - Code that is injected into the background is constantly running from the start of Google Chrome till the time you close. Only one instance of this runs at a time (with the exception of when Chrome is open in Incognito mode and regular mode, in which case two instances of the script are running, one in incognito mode and one in regular mode). Nonetheless, the biggest issue with this is that every network request made isn't easily visible (you'll understand what I'm talking about in a second).

Content Scripts - These scripts are only instantiated (run) when they match certain urls. The code in these scripts are injected directly into the tab the url is loaded in. These can prove most dangerous. However, because these scripts run in the current tab, it's also much easier to check what the script is doing.

Popups - Popup scripts are only run when you click on the extension icon. They are also run independently of the active tab but because they are run don't have as much control as content scripts and background scripts do.

Now that you know exactly how and where code can be executed, there's one important thing you should know: the content script can easily communicate with the background script and vice versa. You will see why this is important in a second.

How do you actually check if you're extension is running a backdoor?

Well, first of all, you need to check what you are installing. Be cautious of extensions that "read (and/or change) all your data on the website you visit. These extensions can easily listen for post requests (for example when you login to a forum or purchase an item from ebay) and they can intercept the data being sent and along with sending it to the original requeter (e.g hackforums or

ebay), they can also send it to themselves.

The first thing you should do is visit "chrome://extensions" in a new tab. This window will bring up all the extensions that you currently have installed on your computer.

On the top right corner of the screen, you'll see a checkbox that says "Developer Mode". Make sure that checkmark is enabled (checked).

Image [\(Click to View\)](#)

The first thing you should do, is check if the extension in question has a background script. If it does, it will look like this:

Image [\(Click to View\)](#)

If there is a background script:

Click on "background page"

Continue to step 6 to analyze requests being sent.

If there isn't a background script:

Open a page where the script is supposed to run.

Right click anywhere on the screen and select "Inspect Element"

Continue to step 6 to analyze requests being sent.

If the network tab is already selected on the popup (Chrome Developer Tools), select the network tab.

Image [\(Click to View\)](#)

Here you will be able to see all requests that are being sent from the page. All you have to do now is select "Preserve Log" and start browsing as you normally do. After a couple of minutes scroll through the requests and look for anything suspicious. Here are some examples of what to look for:

Image [\(Click to View\)](#)

Image [\(Click to View\)](#)

As you can see, if an extension is listening to your requests, the request has to go from your computer to the hacker's server. If you see a request to a suspicious website or if you see something being sent that you don't think should be sent, then you should uninstall the application

right away.

Also, one last note, if I were a malicious coder and I knew that my requests could be sent, I wouldn't be sending the requests right away. Rather, I'd wait a random amount of time or better yet, wait until the user closes the browser and then send the login/stolen information then. The best way around this is to take a look at the source code of the extension. You can take a look at where Chrome stores extensions [here](#)

-KoH

King of Hearts - August 29, 2015, 8:39 pm

Reserved for future use.

-KoH

Fish - August 29, 2015, 8:45 pm

nice post! im gonna see if my extensions are good. thanks for sharing

Hamware - August 31, 2015, 9:21 am

Awesome post mate, revamping the browser section. <3

bonjours_bonjours2015 - August 31, 2015, 9:30 am

thank you bro for your post

thank

King of Hearts - September 7, 2015, 10:14 am

(Aug 31, 2015 11:27 PM)xadamxk Wrote: [\[CITE\]](#) Wow, this thread needs far more attention than it is getting.

Thanks a lot. Hopefully, this will receive some hits from the next news edition.

thefunstash - September 7, 2015, 6:21 pm

Thanks man. Well writ, solid advice.

King of Hearts - September 29, 2015, 5:14 pm

(Sep 7, 2015 06:21 PM)thefunstash Wrote: [\[CITE\]](#) Thanks man. Well writ, solid advice.

Thanks. I'm glad you've found this useful.