Malicious Chrome extension

Amit Waizman, Rivka Buskila

November 2022

1 Abstract

In this study we perform a large analysis of Chrome extensions by investigating different extension versions. We use this data to find out whether Chrome extensions are malicious or not.
We analyze the data collected by articles, records and current information to identify various anomalies and accordingly build our model. The model is based on integrated information of properties of crx files. Each file contains JavaScript, CSS, HTML and a manifest file.

2 Introduction to your work and related background

Browser extensions are a separate type of module. The main difference is that extensions are usually just source code, but plug-ins are always executables (i.e. object code). extensions are no longer supported by the major browsers, but extensions are widely used.
The most popular browser, Google Chrome has over 100,000 extensions available. Google Chrome extensions are programs that you can install in your Chrome browser to change its functionality. These extensions can help automate certain functions in your browser, modify existing behaviors, and improve the convenience of your software. There are even Chrome extensions that can improve your SEO. Chrome extensions are built with HTML, JavaScript, and CSS scripts and are essentially small websites uploaded to the Chrome store.
The only difference between a Chrome extension and a regular website is that extensions contain a manifest file, which gives them a specific function to execute. You can think of extensions as a piece of code that changes your browser experience.
Google Chrome extensions are meant to make your life easier. With Chrome browser extensions that help you get discounts, correct your grammar, take screenshots and watch shows with friends, downloading an extension can be very tempting. However, The extensive help of Chrome extensions has caused a lot of interest in attackers trying to misuse the extensions. Malicious plug-ins

mimic the appearance of popular plug-ins to compromise your privacy.
Malicious plug-ins redirect users to phishing sites and insert partner IDs into cookies of e-commerce sites, plug-ins also track user browsing activity, and every site visit is sent to servers owned by the plug-in creator.
The creator of the plugin can insert code into e-commerce sites the user visits, and receive affiliate payment for each item the user purchases. There is an implicit privacy violation for the consumer when browsing data is shared. If an attacker exploits vulnerabilities presented in browser extensions, it is likely that the attacker will gain full privilege access to the victim's browser and actually infiltrate the computer system. In this paper, we present a machine learning-based approach to detect malicious Chrome extensions This is by analyzing HTML, JavaScript and CSS scripts and manifest.

3 Related Works

The article "You've Changed:: Detecting Malicious Browser Extensions Through Their Update Deltas" investigated malicious versions of plugins. They investigated 922,684 different extension versions and used these data to discover malicious versions of extensions. They proposed a two-stage system that first identifies malicious extensions based on anomalous extension ratings and locates the code that was added to a benign extension to make it malicious. they encode these code deltas accordingly the APIs that they abuse and search our historical dataset for others similar deltas of extensions which have not yet been flagged, neither by users nor by Chrome's Web Store. They found 143 malicious extensions belonging to 21 malicious clusters, exhibiting a wide range of abuse, from history stealing and ad injection to the hijacking of new tabs and search engines. This work demonstrates that clustering extensions based on the similarity of their code deltas is a step in the right direction and can detect malicious extensions in an abuse-agnostic way. Current systems that aim to limit the abuse from malicious extensions can benefit greatly by our proposed extension-analysis techniques to identify extensions that are currently evading detection.
The article "Hulk: Eliciting Malicious Behavior in Browser Extensions" proposed Hulk, a dynamic analysis system that exposed extensions to honeypot-like content (a network-attached system set up as a decoy to lure cyber attackers and detect, deflect, and study hacking attempts to gain unauthorized access to information systems) and monitored whether these extensions exfiltrated that content. Using these techniques, Hulk discovered 130 malicious browser extensions that had evaded prior detection systems and were installed by millions of users.
The article "Chrome Extensions: Threat Analysis and Countermeasures" The widely popular browser extensions now become one of the most commonly used malware attack vectors. The Google Chrome browser, which implements the principles of least privileges and privilege separation by design, offers a strong security mechanism to protect malicious websites from damaging the entire

browser system via extensions. In this study, we however reveal that Chrome's extension security model is not a panacea for all possible attacks with browser extensions. Through a series of practical bot-based attacks that can be performed even under typical settings, we demonstrate that malicious Chrome extensions. Using a prototype developed on the latest Chrome browser, we show that they can effectively mitigate the threats posed by malicious Chrome extensions with little effect on the normal browsing experience. In this study, we have conducted an experiment-based study on the security of the extension support in Google Chrome browsers. We have shown that under the existing security model for extensions in Chrome, it is not difficult to launch large-scale bot attacks. Through in-depth analysis, we find that the problems are rooted from coarse-grained privilege management for the extension components and undifferentiated access permissions for DOM elements in web pages. Accordingly, we propose new policies to enforce micro-privilege management and differentiate DOM elements, both of which have been implemented in our prototype. In particular, considering the compatibility with existing web applications, we develop an extension to automate the sensitivity assignment for different DOM elements. Our experiments show that our design can effectively mitigate security threats without affecting the user's browsing experience.
in the article "Trends and lessons from three years of fighting malicious extensions" report on three years of detecting malicious extensions in the official Chrome-Web Store, using a combination of dynamic analysis, permission analysis, developer reputation, and static analysis.
in the article "Security Analysis of Chrome Extensions" focuses on the security of Chrome extensions and what enables malicious attacks through Chrome extensions to occur. We take a look at past attacks and organize them into 2 categories - attacks from intentionally malicious code within the extension and attacks made possible by the insecurity of benign extensions.
In the article "An integrated approach to static and dynamic analysis for detecting malicious browser extensions" They apply static and dynamic techniques to analyze an extension for feature extraction. The parsing process extracts features from source codes including JavaScript codes, HTML pages and CSS files and extension execution activities. To ensure the robustness of the features, a feature selection method is then applied to retain the most relevant features while discarding features with low correlation.

4 feature extraction

manifest file:

We issued permissions to the manifest file and used three types of feature:
1. We counted a number of permissions
2. We extracted the permissions and used Wors2Vec to weight them
3. We used a blacklist of common permissions for malicious use

css file:

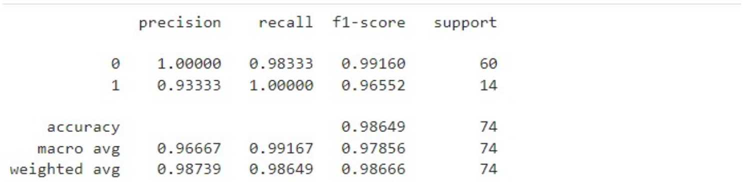
1. We extracted urls and checked whether they are suspicious by a regular expression
2. We checked whether it contains a script.

js file:

1. We checked whether it contains obfuscated code

5 Result

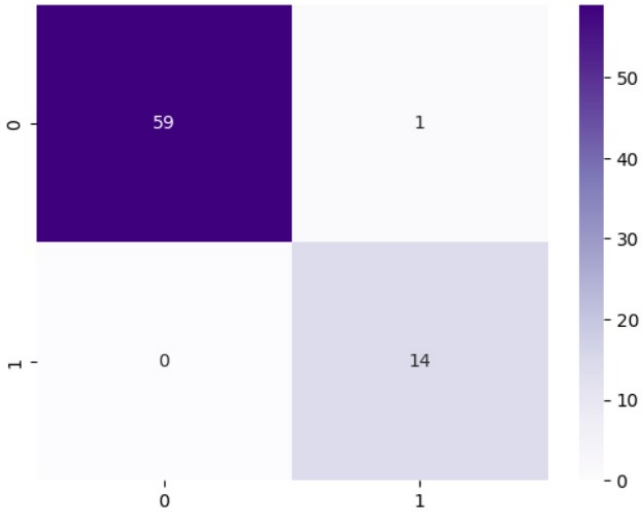
The results of the model show that
our model identified about 99% of malicious chrome additives, through in-depth analysis and preliminary research in which we highlight the structure of the CRX files, and extract relevant features to create a model that will cooperate reliably. We used the supervisor method and the AdabostClassifier model In which we achieved good results and high accuracy described in the figure1 Unlike previous works where they focused on a specific blacklist of permissions and suspicious words.
Another study we started using the Unsupervised method we tried different models but the results are not significant so we went for a combination of several models which are: ellipticenvelope, oneclassssvm And we reached much more satisfactory results described in the table above



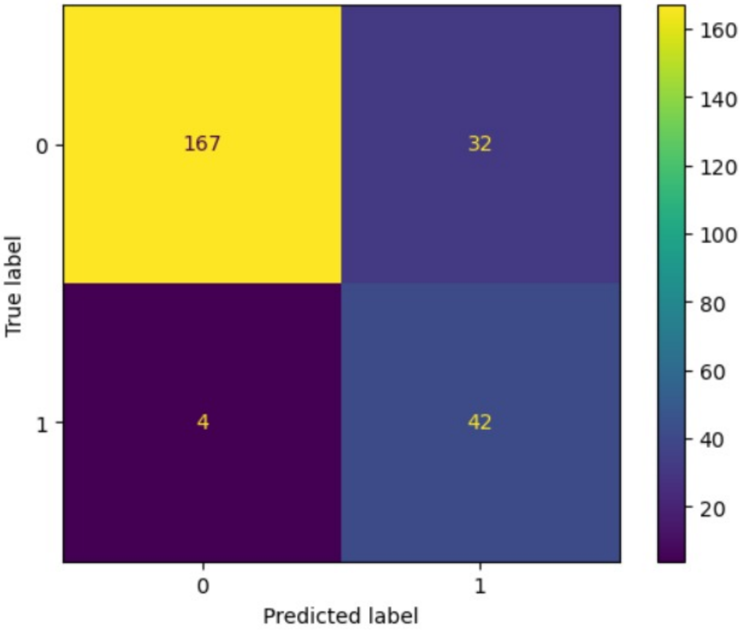
```
              precision    recall  f1-score   support

           0    1.00000   0.98333   0.99160        60
           1    0.93333   1.00000   0.96552        14

    accuracy                        0.98649        74
   macro avg    0.96667   0.99167   0.97856        74
weighted avg    0.98739   0.98649   0.98666        74
```



supervise



Unsupervised accuracy=0.85%, recall=0.91%

6 Summary

Google Chrome extensions can be a convenient and powerful tool for users Extensions offer a large subset of browser add-ons. At the same time, there are many attackers who abuse its power.
To combat this, we developed a learning machine that knows how to deal with malicious chrome plugins.
When the machine receives a malicious chrome plugin, it knows how to analyze and determine if it is malicious or not malicious.