INSTITUTE of TECHNOLOGY CARLOW
Institiúid Teicneolaíochta Cheatharlach

Project Research Document

Student Name:	**Artur Katelovic**
Student Number:	**C00242207**
Module:	**Project**
Stage/Year:	**4**
Date:	**??/??/2021-2022**

Table of Contents

Introduction

In this report I have researched many topics in order to help me with the development of my project. My project is a Google Chrome extension that will scan a website and check if it's a fake/fraudulent website and prevent the user form accessing it and inform them of this.

The first thing I researched is other add-ons that do the same or similar things that my add-on needs to do. The first add-on that I researched is "Avira Browser Safety", I researched this one first because I use it and already familiar with it. The second add-on that I researched was "WebTitan" which gave me a lot of ideas on how to do my add-on. The third add-on that I researched was "Tessian" which also gave me lot of ideas and also provided me with a lot of information on how to spot a cloned website. I also researched some other add-ons that were open source on "github" which gave a lot of ideas and information.

I researched all the languages technologies and tools that I will need to use in order to make this add-on. The languages that I researched were HTML, CSS and JavaScript, I researched these specific ones because I chose to make a chrome extension and those are the standard languages used to make a chrome extension.

The first tool/technology that I researched was the NER (name entity recognition) the first one looked at was a python library called spaCy NER, I researched this first because I was unaware that using python to make a chrome extension would be too much unnecessary work, so I had a look at JavaScript NER's and the one I chose was the Wink NER which is a JavaScript NER which is what I needed in order to make my extension. The next technology that researched was the API, I looked for one for a very long time however I was recommended to use an API called Axio, so I researched that one and decided to use it as I felt it was the best one to use.

I also did research into Google Apps Scripting, since I was making a google chrome extension, I thought that I would need to use this so I did as much research as I could and decided on using it. The finale tool/technology that I researched for my project was the Local Web SQL, I wanted a local database that would store website that the extension has marked as malicious and prevent the user from accessing it.

Project Description

The project that I am working on is a chrome extension that will use various tools and technologies as shown in this document to check for fraudulent cloned websites.

This extension when turned on will check the website that the user is trying to access and check if it's a fraudulent cloned or the real website and if the website is a fraudulent clone it will warn the user and prevent them from going to that website, if the user still wishes to visit that website, they will then need to confirm that they want to access that website or simply turn it off.

The extension will check if the website is a fraudulent clone by using a text processing system to extract phrases and check if those phrases are unique or if they are used in another webpage, it will also check for spelling errors, if there are any images it will also check if they appear on any other webpage.

It will also take the URL and using an API it will connect to a website/webapp like visrustotal and check for viruses and any other malware that might be on the website that the user is trying to access and then it warns them that the website is fraudulent and fake and contains malware, it advises them to leave that website and not to put in any personal information into the input fields.

Abstract

This research document is an insight into what I researched in order to help me with my project. I did a lot of research into the add-ons that are similar to what I need to do, I researched all the languages, technologies and tools that I will need to use in order to achieve my goal, and this research document will show all that I have researched to help me with the making of my project.

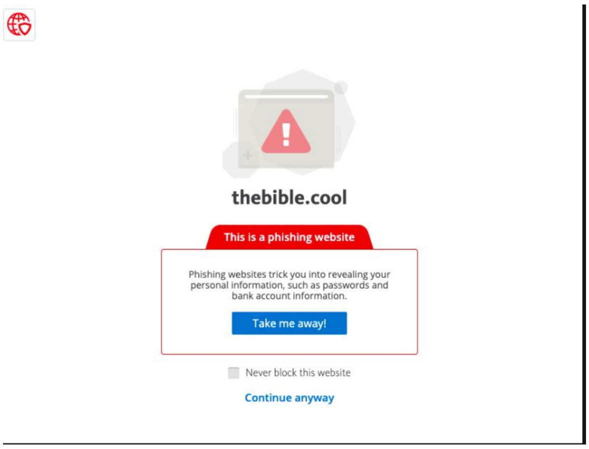
Research into Similar Add-ons

In order to do my add-on, I searched for similar add-ons and researched them to understand how they work in order to help me with making my own add-on. There were quite a few of add-ons that were similar to what I needed to do.

Avira Browser Safety

The first add-on that was similar to what I needed to do was "Avira Browser Safety" I was familiar with this add-on since I have been using it myself for quite some time now and it is very similar to what I have to do for my add-on.

It blocks malicious adds, stops malicious and phishing websites, prevents companies from tracking you, prevents browser hijacking and detects unwanted apps in the downloads. However, I was mainly interested only in the "stops malicious and phishing websites" and when looked for how they did it all I could find on their website was that it scans the webpage source code and looks for malware and trackers and if it finds any it will block the website and warn the user like so.



thebible.cool

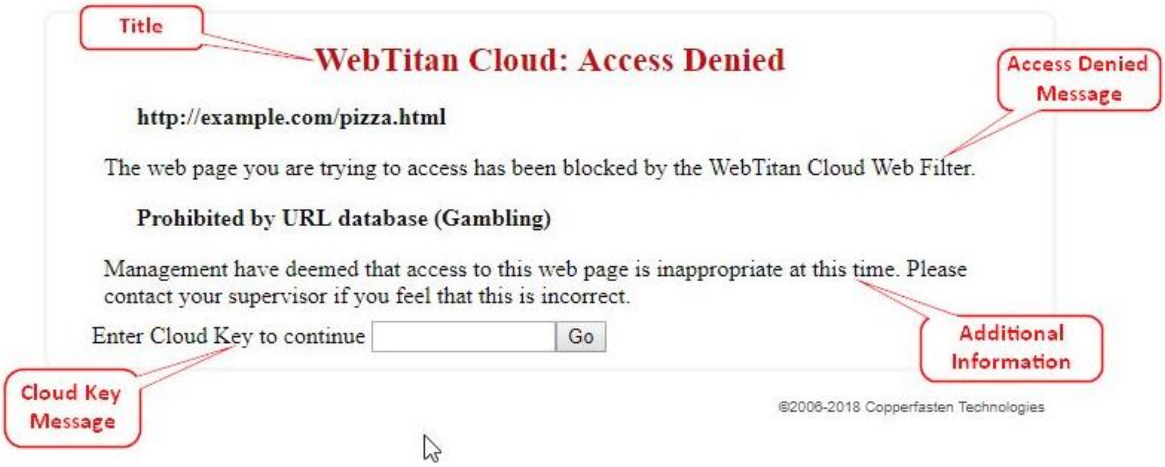
This is a phishing website

Phishing websites trick you into revealing your personal information, such as passwords and bank account information.

Which I could also use that method to detect if a website fake/fraudulent I can user different tools to scan the webpages source code as well as other methods in order to find out if a website if fake/fraudulent and warn the user if there is any malware or tracking on the webpages source code. So, it was very helpful in order to give me an idea on how I can make my own add-on.

WebTitan

Another useful add-on that I found that will help me with my project and give me a better idea on how to do my add-on is WebTitan. WebTitan gave a good insight into how you can spot a malicious website such as they will ask you to download software, save a file or run a program, when you visit a malicious website it automatically launches a download window or your asked to download an invoice or receipt such as a PDF file or an executable, they will also tell you that your computer is already infected or your plugins or add-ons are out of date or you have won a competition and other things like that. They also mention something quite useful on how to block users from accessing a fake/fraudulent website, they mentioned to use a powerful web filtering solution in order to achieve this.



WebTitan Cloud: Access Denied

http://example.com/pizza.html

The web page you are trying to access has been blocked by the WebTitan Cloud Web Filter.

Prohibited by URL database (Gambling)

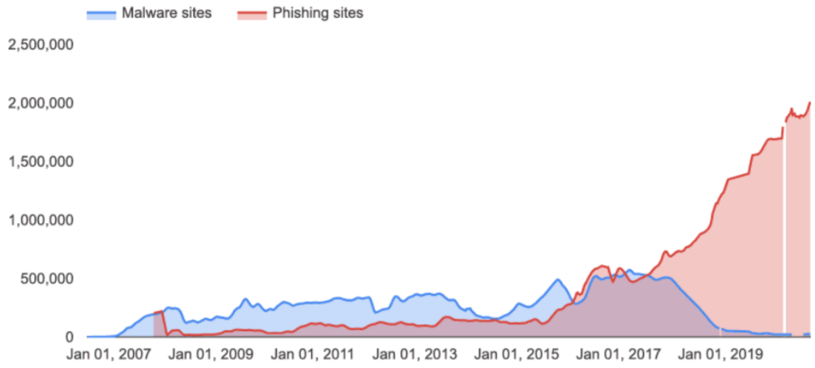
Management have deemed that access to this web page is inappropriate at this time. Please contact your supervisor if you feel that this is incorrect.

Enter Cloud Key to continue

From researching this add-on, I learned that I would need to create or find something similar to the WebTitan web filtering solution in order to prevent user form accessing fake/fraudulent websites.

Tessian

Researching Tessian was also helpful since it gave more insight into phishing/fake/fraudulent websites, like how to know if you're on one however it was the same as the others as its common knowledge, it also informed on how common these phishing websites are however from what they stated phishing website are becoming more common than malicious website as cybercriminals prefer to use phishing website over malicious ones. As can be seen on this Google's Safe Browsing reports



They also gave a Real-World example where a group called BAHAMUT targeted consumers, business and government officials via phishing email, fake mobile apps and a huge network of malicious websites. However, they didn't provide much information on how they combat this, all they said was that they try to stop Spear phishing emails since that's how most users end up on a phishing website however i might use this as a way to prevent a user from reaching a fake website.

Technologies, Tools and Languages

Languages

For the languages that I will be using for this project I am limited in which languages that I can use since I chose to do a chrome extension. In order to make a chrome extension the only languages that I will be using is HTML, JavaScript and CSS, since those are the main languages that are used in order to make an extension for chrome and using any other languages will result in a lot more work and very poor efficiency therefor, I will only use those languages.

What is HTML?

HTML or Hyper Text Markup Language is the standard language used for documents designed to be used on a webpage it will also be assisted by CSS and JavaScript.

What is CSS?

CSS or Cascading Style Sheet is the standard language used for making the presentation of a webpage and used to make the webpage look all nice and add a style to it.

What is JavaScript?

JavaScript or JS is a text-based language used in both the client side and server side that will make the webpage interactive. The other two languages that will be using (HTML, CSS) give the extension structure and style to it, where as the JavaScript it gives it the interactive elements that engage the user.

spaCy NER

My project will need to be able to find and warn the user of a fraudulent websites so it will need to be able to detect a fake website, to do this I will use spaCy NER which is a python library used for name entity recognition.

Name entity recognition is a tool which locates and identifies the named entities in a unstructured text into standard categories like person's name, locations, organizations, time, expressions, quantities, monetary values, percentage, codes and some others. spaCy also allows for an option to add arbitrary classes to entity recognition systems and update the model to also include the new examples from the already defined entities within the model.

As mention before this extension will need to able to identify a fake/fraudulent website and there are a few ways of doing so, like checking the domain name, check the domain age, check for poor grammar and spelling mistakes and check the websites privacy policy. Using spaCy I can check for poor grammar and speeling mistakes I can also use it to check the privacy policy on the website and check if it is correct or check for grammar mistakes in it also, I can use it to check the domain name and its age to verify if it is a legitimate website. However, I am unable to use this NER because in order to make a chrome extension I need use HTML, JavaScript and CSS and if I want to use spaCy NER it will complicate the extension and make a lot more work then I need to do so I decided on using another NER and this time I will need it to be JavaScript NER.

Wink-NER

Wink NER is a JavaScript NER which is what I decided to use as my main NER since its JavaScript NER it should be easy to implement into a chrome extension.

Wink-NER is a smart Gazetteer-based NER which can be trained quite easily to suit specific needs such as it can differentiate between Manchester United and Manchester in a sentence and tag them as a football club or city. As I mentioned before in the spaCy NER section I can use this NER to scan a webpage and check for spelling errors and check if the URL seems legit and I can use the information that I get back and then decide whether or not the website is fake/fraudulent

If I need to assign a specific pos tag to an entity, this can be achieved by including a property pos in the entity definition and assigning it to a desired pos tag. Another great thing about this NER is that it is an open-source packages for such things like Statistical Analysis, Natural Language Processing and Machine Learning in Nodejs. On their website they also state that the code is thoroughly documented for easy human comprehension and has a test coverage of ~100% for reliability to build production grade solutions.

API'S (rework a little)

If the website is done well the NER might not be able to find anything wrong with the website so there is another way of checking if a website is fake/fraudulent and that is to run it through a virus scan. I can use a API such as Axios.js to connect to a web app like virus total and perform a full scan and if it return a certain result it will flag it as a fake/fraudulent website.

Axios is a promise-based HTTP client that uses an easy-to-use API, and it can also be used in both the browser and Node.js which is what I will need for my extension to connect to a virus scan web app. My extension will get the domain name of the website that the user is on and if it get a green light from the NER and there is nothing that seems off then I will pass the domain to the "virus total" using the axios API and run a virus scan and then I will get the results and if they are over an certain mark I will warn the user that the website that they are on is fraudulent and it is best if they leave that website.

Chrome Extensions

For my add-on I will be making it as a chrome extension since chrome is the most popularly used browser. A Chrome extension works by using either the page actions or a browser action. A page action is only used for that specific page while a browser action is used no matter where you are in the browser. The chrome excision will need to have a good user-friendly user interface and easy to use and set up. The main languages used for a chrome extension are HTML, CSS and JavaScript which are the main languages I will be using.

Google Apps Scripting

Google Apps Scripting is a JavaScript cloud scripting language that allows for an easy way to automate tasks across Google products and other third-party services and also the build web applications. You write code in modern JavaScript and you will access to all the built-in libraries for Google Workspace applications such as Gmail, Calendar, Drive and others.

What is used for?

Google Apps Scripting is used for many things such as, adding custom menus, dialogs and sidebars to all the Google Applications, it is also used for writing custom functions and macros for Google sheets. You can also use it for publishing web applications either standalone or embedded in Google site. You also use it for interacting with other Google services such as AdSense, Analytics, Calendar, Drive, Gmail and others, you can use it to convert a Android application and make it into a Android add-ons, you can also streamline Google Chat workflows by building a chat bot however the main thing I am interested in from the Google Apps Scripting is the fact that you can build add-ons to extend Google Docs, Sheets, Sliders and Forms and you will be able to publish them to the add-on store.

Local Web SQL

I will take advantage of the Local Web SQL in the chrome web browser to store website that my extension will mark as fake/fraudulent in the local web SQL database and if the user ends up on the same fake/fraudulent website it will already know that it is a malicious website and it will block it faster since all it will have to do is look through the database and if it finds that database them will prevent the user from entering that website. However, I can also use it to store a huge range of legitimate website and if the user ends up on one that isn't in that database it will scan that website and check if its fake/fraudulent it isn't then it will add it to the Local Web SQL however if it's a malicious website then it will warn the user and blacklist it and it will make sure that the user is blocked from accessing it later.

Summary and Conclusion

So, in conclusion, I believe that I did all the right research into the correct topics in order to help me make my add-on. I researched similar add-on like "Avira Browser Safety", "WebTitan" and "Tessian" which gave me a rough idea on what I need to do and how to get started with my add-on which was a huge help to getting started. The languages, technologies and tools that researched were the one I believe would be the easiest to work with and help me make my add-on. The tools and technologies that I researched were spaCy NER and Wink NER, I looked into Google Apps Scripting, Local Web SQL, Axios API and chrome extension. The research that I that I did, I believe that its sufficient and was really helpful, it gave me an idea on how to do and start my extension.

Appendix

Glossary

Bibliography

Tripathi, A. (2020, September 15). *Named Entity Recognition NER using spaCy | NLP | Part 4*. Medium. Retrieved October 27, 2021, from https://towardsdatascience.com/named-entity-recognition-ner-using-spacy-nlp-part-4-28da2ece57c6

Knepper, J. (2021, June 24). *11 Ways to Check if a Website is Legit or Trying to Scam You*. Home Bank of California. Retrieved October 15, 2021, from https://www.hbc.bank/11-ways-to-check-if-a-website-is-legit-or-trying-to-scam-you/

Eschweiler, S. (2018, June 18). *Getting Started With Axios - CodingTheSmartWay.com Blog*. Medium. Retrieved October 30, 2021, from https://medium.com/codingthesmartway-com-blog/getting-started-with-axios-166cb0035237

Jacques, N. (2021, May 24). *Axios Beginner's Guide: A Handy Promise-based HTTP Client - SitePoint*. Axios-Beginner-Guide. Retrieved October 30, 2021, from https://www.sitepoint.com/axios-beginner-guide/

spaCy 101: Everything you need to know · spaCy Usage Documentation. (n.d.). SpaCy 101: Everything You Need to Know. Retrieved November 2, 2021, from https://spacy.io/usage/spacy-101/

Security check. (n.d.). Avira-Browser-Safety. Retrieved November 3, 2021, from https://support.avira.com/hc/en-us/articles/360001180425-Characteristics-of-Avira-Browser-Safety

Avira Browser Safety - Free security add-on with anti tracking. (n.d.). Avira. Retrieved November 3, 2021, from https://www.avira.com/en/avira-browser-safety

Overview of Google Apps Script |. (n.d.). Google Developers. Retrieved November 11, 2021, from https://developers.google.com/apps-script/overview

wink-ner - Wink JS - Summary. (n.d.). Winkjs. Retrieved November 22, 2021, from https://winkjs.org/wink-ner/

What is Web SQL - javatpoint. (n.d.). Www.Javatpoint.Com. Retrieved November 20, 2021, from https://www.javatpoint.com/what-is-web-sql