AUTO OSINT

Research Document

LIAM MOORE (C00196503)
Supervisor – Richard Butler

Abstract

This document records the research I have undertaken in order to develop an OSINT extension that will allow Cyber Security Professionals to speed up their OSINT investigations and enable them to spend their time on more important tasks. In this document I talk about Open-Source Intelligence and what it is, including the Intelligence Cycle. My list of research objectives including existing tools, data reliability, choosing a suitable browser and coding languages as well as the API's I will be implementing.

Contents

Introduction

This document showcases the research I have undertaken to understand how my project will function and how it evolves from an idea to a working product. As cyber attacks are a growing threat to any business or corporation in today's world, it is important to be able to trust that cybersecurity professionals can gather correct information in a timely manner. It is doubly important that the information collected from Open-Source Intel (OSINT) is correct and reliable. Utilising different Application Programming Interfaces (API's), Auto OSINT will be able to provide results and information almost instantly on specific IP addresses and hash values. The first section of this document will cover the problem statement and the reason why I felt the need to create my Auto OSINT extension.

Problem Statement

In a world where cyber security threats are on the rise, it is of utmost importance to ensure that cyber security businesses and organisations are investigating these threats to the best of their ability. Open-Source Intelligence (OSINT) investigations have become more important for threat detection. Despite the existence of various well received OSINT sites such as VirusTotal, AbuseIPDB and WhoIs, there continues to be a lack of browser extensions that aid in investigations. Currently, the procedure requires cybersecurity professionals to manually navigate to these OSINT sources which can become very time consuming, and this can result in a delayed response to security breaches.

To address this problem within the industry, I have proposed the idea of the Auto OSINT extension. The aim of this browser extension is to reduce the overall time spent conducting investigations and enable a prompt response to cybersecurity alerts allowing for cybersecurity professionals to concentrate on other areas. Aswell as eliminating the need for cyber security professionals to navigate to the sites manually, it also reduces time by allowing users to operate from a centralised hub that will be able to offer an export function, which will create a document with the most relevant information needed by cyber security professionals to record their information for future investigations and reduce the need for this to be done manually. This extension should seamlessly integrate with browsers, offering a centralised hub where cybersecurity professionals can submit information from their own investigations and receive results from popular and reliable OSINT sites.

What is Open-Source Intelligence (OSINT)

SANS Institute defines OSINT as “intelligence produced by collecting, evaluating and analysing publicly available information with the purpose of answering a specific intelligence question” (Gill, 2023). Open Source content can be found from a variety of places including:

- Social media platforms
- Images, Videos
- Websites
- The Dark web

All of the above locations share a common theme, they all house information that can be accessed by any member of the public. This information can be analysed and looked at from a critical thinking mindset, and once this happens, it becomes intelligence. OSINT is used in many different areas of work, including, Government, Military, Cyber Security Professionals and Social Engineers.

Intelligence Cycle

The intelligence cycle describes the five steps that is undertaken when carrying out OSINT. Those 5 steps are:

1. Preparation
2. Collection
3. Processing
4. Analysis
5. Dissemination

Preparation

This step is when the needs of the request are assessed. This includes understanding the objectives and identifying the best sources to use and find the relevant information that you are looking for.

Collection

The most important step in the Intelligence Cycle is collection. This involves collecting data and information from as many relevant sources as possible.

Processing

After collection, the data must be organised or collated to better make sense of what has been returned from collection step.

Analysis

The collected information needs to be analysed to make sense of what was collected. This step can be used to identify patterns and develop timelines. This is the step where reports can be produced, conclusions can be drawn and what steps come next in the whole process.

Dissemination

Dissemination is where the findings are presented and delivered, ultimately answering the question we were presented with at the beginning of the cycle and why we are carrying out this OSINT investigation. (Gill, 2023)

Below is an example of the intelligence cycle, with all five different steps visible.

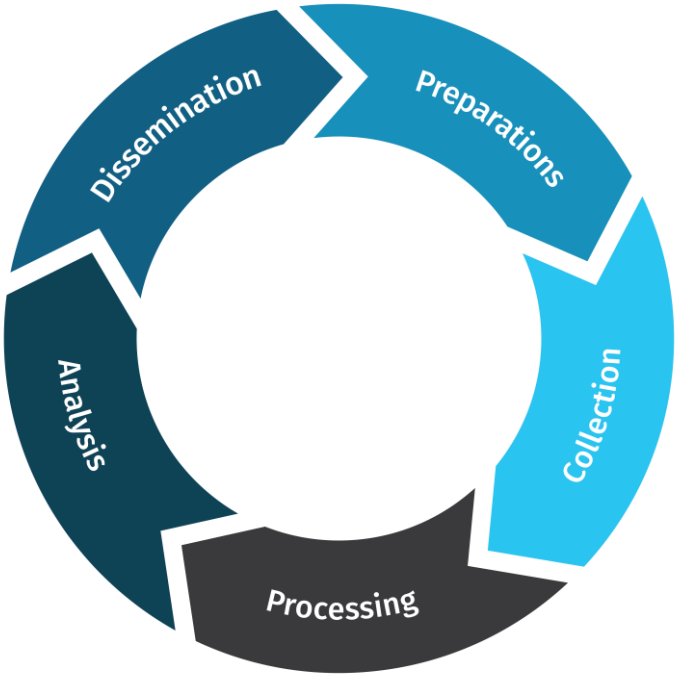

Figure 1 - Intelligence Cycle (Gill, 2023)

Research Objectives

Existing Tools and Methods

As part of my research, I wanted to discover what already exists in the industry and how it compared to my tool. Although I have not found an exact match for my OSINT tool, there are variations that exist. I decided to research what tools already exist to gauge how other developers approached the same scenario. Understanding how existing tools operate could help me anticipate and avoid future potential obstacles. Furthermore, I feel that it could help me in my thought process and identify some possible additional features that would benefit and enhance the tool which I could then incorporate into my project without straying too far from the main objective.

The first comparison I have found lies in the CrowdStrike suite. During my work experience I was lucky enough to have the opportunity to use CrowdStrike. “CrowdStrike is a cloud-delivered platform that unifies next-generation antivirus, endpoint detection and response, threat intelligence, managed threat hunting and security hygiene.” My understanding of how CrowdStrike alert queue interface worked was that the CrowdStrike admin would configure the rules that would generate alerts. These rules would consist of certain actions that a device would carry out that would be deemed to be suspicious and thus creating an alert to be investigated. After clicking into the alert, you are met with the alert details screen it is here where I discovered the comparison.

CrowdStrike has a built in VirusTotal (an open-source intel website that is widely used in the industry) button that appears on the alert details page and when clicked, will take the associated file hash value that is located underneath and open a new window displaying the results of the search in VirusTotal.

Another similar feature is found within the DarkTrace Endpoint platform. Like CrowdStrike, most of my experience with this product came from my time on work experience. In DarkTrace’s case, this feature is not in the form of a button and is less likely to be obvious to users. Like the built-in feature in the CrowdStrike platform, it does open in a new tab.

Since one of the OSINT sites I am going to use for my project will be VirusTotal, my research led me to discover that they, themselves have their own browser extension called VT4Browsers. From my experience in using this extension it appears to work differently to how my Auto OSINT browser extension will work. I noticed that it was mainly verifying a user’s file that had just been downloaded against their already existing database of malicious files. It did this to determine whether the downloaded file was malicious and how severe it is.

Ensuring Data Reliability

As this extension is mostly aimed at cybersecurity professionals working in the industry it is imperative that it can guarantee the data it collects and provides back to the user is reliable. This is the reason I chose for my extension to use the APIs from the OSINT sites: VirusTotal and AbuseIPDB. Both sites are well known and are widely used amongst cybersecurity professionals and are trusted within the industry, thus my reasoning for choosing these sites.

Choosing a Browser

Why Chrome?

Since the moment I decided I was going to build an OSINT extension I was instantly convinced that it would be a Google Chrome extension. Why was this? Well, it is a known fact that Google Chrome is by far one of the most popular browsers amongst users. In 2021, it was observed that Google Chrome was the leading browser with 63.84% of devices around the world running it (itpro, 2023). With it being the most popular browser, it means that it is compatible with almost all webpages. If features are not available natively within Google Chrome, you can be almost sure that someone has developed one. Google Chrome also has integration with all Google services which contributes to its high usage and popularity. One of the advantages of Chrome is the speed at which it renders web pages. Although one drawback of the Chrome browser is the Random Access Memory (RAM). Chrome is notorious for its high RAM usage. Adding an extension on top will add to the usage of resources and that will have to be considered when developing the extension. In a paper titled "A Study on the Systematic Review of Security Vulnerabilities of Popular Web Browsers", it was deduced that Google Chrome was in fact the most secure browser when it comes to popular web browser attacks like SQL Injection, Cross-site scripting, and cross-site request forgery. This paper carried out various tests against the most popular and well-known browsers. (Tewari & Datt, 2021)

Mozilla Firefox

Another option I could have chosen was Mozilla Firefox. This is my personal browser of choice and I have been using it for many years. There are many benefits to using Firefox such as the security features it has to offer. In 2021, Firefox became the first browser to offer the ability to block cross site tracking. It also implemented DNS over HTTP (DoH), making it difficult for malicious users to monitor your web traffic and has features to block scripts that mine cryptocurrency. Despite these advantages, Firefox has seen its popularity plummet from almost 30% at one stage to 3.9% in 2021 which is the main reason for my decision not to select it. (itpro, 2023)

Microsoft Edge

The third browser I gave thought to was Microsoft Edge. Microsoft Edge is the successor to Internet Explorer and comes pre-installed with Windows 10 and 11 machines. Despite this, Microsoft Edge gained only 3.99% popularity in 2021. (itpro, 2023) Comparing it with Google Chrome and Firefox it severely lacks any competitiveness. The main factor being that Microsoft can decide what extensions Edge is compatible with although it is supposedly purely for security reasons. Ultimately, users of Edge report many problems with using extensions and therefore, that is the reason I decided on not using this particular browser.

Coding Languages

Coding languages are how developers communicate instructions to a computer and allow them to get the computer to perform how the user wants to, (Garon, 2023). There are many reasons to choose a specific coding language. Many factors come in to play such as difficulty, integration, and execution time. I needed to decide which programming languages would be suitable for my project. I have some experience and a decent understanding using certain languages already such as Java and C++. Although I have previous experience with these languages. Upon further research I ultimately settled on using Python because the API's I will be using work with Python.

HTML

For my project I plan on developing my extension using a variety of different coding languages. The first of these is HyperText Markup Language (HTML). HTML is a markup language which differ from normal programming language as they are used to determine how elements are displayed within or on a webpage (Codecademy, 2021). I will be using HTML to design a simple and user-friendly interface that will enhance the user experience.

CSS

In addition to HTML, I will also be utilising Cascading Style Sheets (CSS) alongside HMTL to further enhance the user experience. In conjunction with HTML, CSS allows for style to be added to HTML or web pages such as colours, font and spacing.

Python

My choice of coding language will be Python. Although I have had minimal exposure to Python, the reason I have chosen Python is because I want to utilise the Python API's from VirusTotal and AbusedIPDB. As these API's play a large factor in the function of my project, I think it may be beneficial for me to have a seamless integration and reduce the number of problems I may encounter to a minimum. Python also has access to many different API library tools that I may find useful throughout the development stage of my project.

One of these helpful libraries is called 'virustotal-api'. This is the API associated with the OSINT site VirusTotal and provides a method for me to scan URLs, hashes, and IP addresses for potential threats. During my research I found that Python, although being a relatively new language is quite beginner friendly, so it would make the task of learning it easier.

Alternative Coding Languages

Although I have chosen Python as my coding language of choice it was not the only option. I also gave some thought to Java and C++. The reason these coding languages were an option to me is because I have more experience with them. Upon further research, I concluded that these languages were not going to be suitable for use.

Java

I have had previous exposure to Java on my course and it is probably the coding language I am most comfortable with and strongest at. While researching which programming language I should commit to for my project, I wanted to compare how different Java and Python are and what benefits one has over the other. This led me to discover a paper entitled "Comparison of Python and Java for Use in Instruction in First Course in Computer Programming". This paper compares how Python and Java perform against each other under certain criteria including memory consumption, code size robustness and execution time. In all but one of these criteria, Python comes out on top. (Ogbuokiri, 2016)

The first criteria show Python succeeds at memory consumption. The following image taken from the above paper indicates how much more memory is consumed by Java than Python.

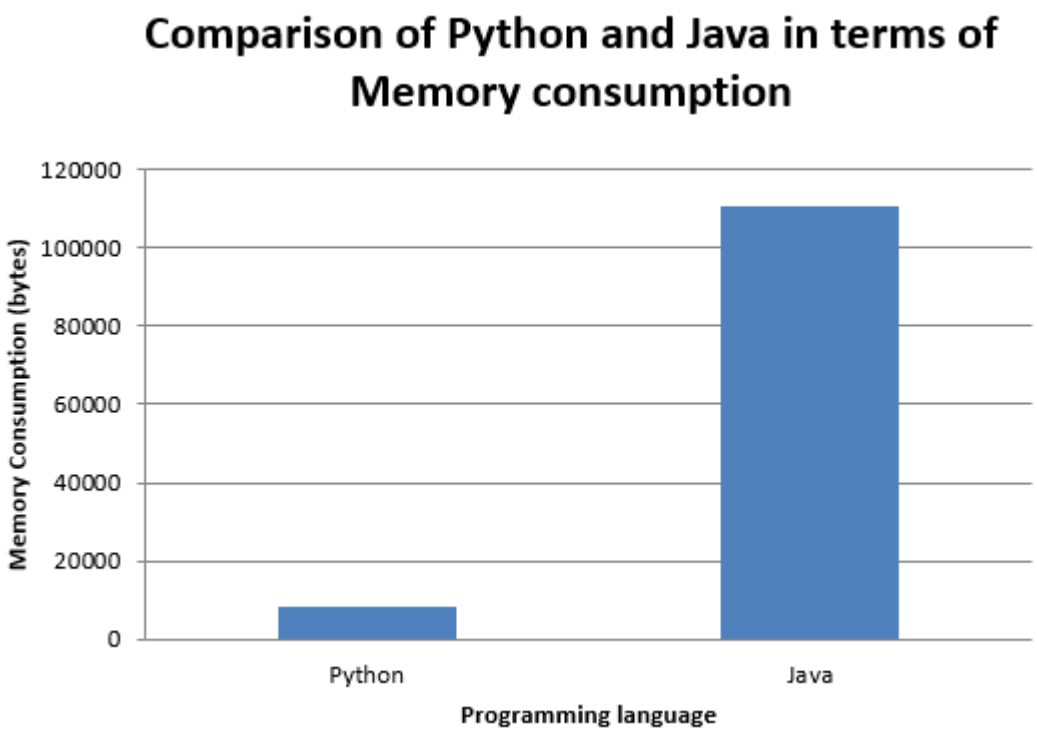

Figure 2 - Memory Consumption Comparison (Ogbuokiri, 2016)

The second comparison that was made compared the code size of each language. It was done using an identical program for each language. The following table shows the difference in size between two identical programs. (Ogbuokiri, 2016)

Programming Language	Code size (bytes)
Python	6,356
Java	72,320

Figure 3 - Code Size Comparison (Ogubuokiri, 2016)

The next test observed in the paper identified the execution time of the languages. This was done using the same program as the authors used for the previous comparison. It was concluded that in this test, Python also had the faster execution time with almost half the length of time that Java had taken. (Ogbuokiri, 2016)

Programming Language	Time (seconds)
Python	11
Java	19

Figure 4 - Execution Time Comparison (Ogubuokiri 2016)

Lastly, the final comparison was how robust the language was. Robust code refers to how the code deals with invalid inputs or how it handles errors. A "robust program should anticipate where problems might arise and compensate for them" (Bishop & Elliott, 2013). It was determined in the paper using a sample piece of code that Java was more robust the Python. It is with these results that we can see Python is much better than Java thus the reason I have chose it as my coding language.

C++

Another language I could have chosen is C++. C++ is a low-level language that is mostly used in software infrastructure and in applications that run on limited resources. This is because C++ is fast and can directly manipulate the hardware of the machine it is running on.(Xiao, 2021). C++ has the ability to utilise pointers, which can help to manage memory and free up resources (Python, n.d.). This makes C++ ideal for developing operating systems as they need to be efficient with system resources. Another area where C++ is used is in game development. As gaming requires a lot of resource intensive functions, C++ allows developers to adjust how memory allocation is handled (Xiao, 2021). Although C++ can be used in association with web browser, I feel like my research on this particular language convinced me that its strengths were better suited to a different application as they didn't really benefit me and what I want to achieve with this project.

API Utilisation

For my project to function as intended, I will be implementing some API's into my extension. API's will allow for communication from my extension to other products or services, simplifying application integration (IBM, 2023). As my project includes scraping OSINT sites for relevant information, the API's must be able to provide that functionality. The OSINT sites I have chosen (VirusTotal and AbusedIPDB) already have API's available for me to use and incorporate into my project.

VirusTotal API

VirusTotal's API allows a user to scan files, submit URLs for scanning and allows the creation of simple scripts to access the information that would normally be generated by VirusTotal (VirusTotal, API Overview, 2023). VirusTotal provide a multitude of resources on how to use the API and how to integrate it correctly. VirusTotal have two separate versions of their API. A public API and a premium version. My research has shown me that the public API is available to anyone who has a VirusTotal account, while the premium version is a paid feature. The biggest difference I have found between the two API's is that the premium version offers to gather more in-depth information and also provides the user with a higher daily number of searches that can be run. The public API, for example, is limited to only 500 requests per day and only 4 per minute. The premium API however does not have any daily request limit and is indicated more towards professionals (VirusTotal, n.d.).

For my project I will be using the public version of the API key as per my research I do not see how it would hinder the general operation and function that I want from my extension.

AbusedIPDB

The other API I plan on using is the AbusedIPDB API. Just like the site itself, this will allow me to utilise their database programmatically for reporting and checking IP addresses (www.abuseipdb.com, n.d.). This API works mostly through Fail2Ban which is included in the configuration. Like the VirusTotal there are many different API versions which all have their own specific target audience and daily limit. Again, here, I am utilising the standard version which will give me 1000 IP checks per day which will be more than enough for my project (docs.abuseipdb.com, 2023).

Conclusion

Through my research I have found the need for a tool like mine can be greatly utilised by cyber security professionals to speed up their investigations. Although there are many other options available at the moment, I feel that mine can offer the same functionality and not require the implementation of CrowdStrike or DarkTrace. I also feel that those I have stated in this report are more focused on large companies whereas my extension can be more focused on one specific user and their needs.

I have deduced that I will be using HTML, CSS and Python to develop my extension as through my research I found that these languages best suit the task at hand and offer me a wide variety of options so that I can have a multitude of ways to implement features. I will be utilising the VirusTotal and AbusedIPDB API's for this as they come from well trusted OSINT sites and provide the information that I think is most necessary for investigations. As there are more OISINT sites out there, if time is not an issue, there is also the option of utilising maybe another API on top but that depends on how the development of my extension goes and if it is worth it.

References

Bishop, M., & Elliott, C. (2013). *Robust Programming by Example.* Springer, Berlin, Heidelberg.

Codecademy. (2021). *What is HTML: Common uses and defining features*. Retrieved from Codecademy News: https://www.codecademy.com/resources/blog/what-is-html/

docs.abuseipdb.com. (2023). *AbuseIPDB APIv2 Documentation*. Retrieved from https://docs.abuseipdb.com/#introduction

Garon, C. (2023). *Why Different Programming Languages Matter and their Pros and Cons*. Retrieved from https://christophegaron.com/articles/mind/why-different-programming-languages-matter-and-their-pros-and-cons/

Gill, R. (2023). *What is OSINT?* Retrieved from SANS: https://www.sans.org/blog/what-is-open-source-intelligence/

IBM. (2023). *What is an Application Programming Interface (API)*. Retrieved from IBM: https://www.ibm.com/topics/api

itpro. (2023). Retrieved from https://www.itpro.com/web-browsers/24796/best-browser-chrome-vs-edge-vs-firefox

Ogbuokiri, A. O. (2016). *Comparison of python and java for use in instruction in first course in computer programming.* Transylvanian Review.

Python, R. (. (n.d.). *Python vs C++: Selecting the Right Tool for the Job*. Retrieved from realpython.com: https://realpython.com/python-vs-cpp/#comparing-languages-python-vs-c

Tewari, N., & Datt, G. (2021). A study on the Systematic Review of Security Velnerabilities of Popular Web Browsers. *International Conference on Technological Advancements and Innovations (ICTAI)*, 314-418.

VirusTotal. (2023). *API Overview*. Retrieved from VirusTotal: https://docs.virustotal.com/docs/api-overview

VirusTotal. (n.d.). *Public vs Premium API*. Retrieved from VirusTotal: https://docs.virustotal.com/reference/public-vs-premium-api

www.abuseipdb.com. (n.d.). *API Documentation - AbuseIPDB*. Retrieved from https://www.abuseipdb.com/api

Xiao, L. (2021). *What is C++ used for?* Retrieved from Codecademy News: https://www.codecademy.com/resources/blog/what-is-c-plus-plus-used-for/

Table of Figures