

**פרוייקט מעשי לתואר B.Sc. במדעים**

**עדכוני סייבר שליליים**

**Negative Cyber News**

<תמונה הקשורה לפרויקט>

**מגישים:** ניב עמוס –

רועי עבודי –

מתן צופי - 312539539

**בהנחיית:** <שמות המנחים> **חתימה: <**חתימות המנחים>

**תאריך הגשה: <**תאריך הגשת הספר לחתימת המנחה>

**Project Summary**

**Introduction**

**Background**

In these days' organizations are more vulnerable to outside and inside cyber-attacks, leaks and exploits. Most organizations are not ready to handle a cyber-attack. For example, a bug in the system can leave access to valuable information open for the public, or hackers hack into the system and acquire personal information of clients and employees. Leaks can include personal data of existing and past clients, which makes the organization uncredible for new clients. Some of the threats are not always well known and exposed to the public, and some organizations even try to cover-up their vulnerabilities, in the purpose of hiding it from their clients or potential clients and business partners. In some cases, the organization may not prioritize or capable of fixing the vulnerability, leaving it open for repeating attacks.

Additionally, in recent years, cyber-attacks became more frequent, and encompass broader range of targets that are not from the hi-tech industry. For example, banks are one of the most common cyber-attack targets. As more products start connecting to the cyber space, they became more vulnerable to cyber-attacks. Products such as smart houses, cars and TV's can become targets for cyber-attacks as they become more common.

Therefore, since people start connecting to the cyber space through means other than their computers and cellphones, they are exposed to direct attacks from other products. Furthermore, as organizations get bigger, having more and more clients, they start providing services outside of their scope, and therefore they become more vulnerable to cyber-attacks and exploits. Big companies, like Microsoft or Google, have array of different services and products, directed to different clients. Each one of these products can be a target to cyber-attack.

**Targets**

Since cyber-attacks are a lot more common than what people think and can occur in wide variety of products, it's easy to lose track of all of them. People sometimes don't fully know all of the cyber-attacks that occur against their products. For instance, data breach against an organization can leave behind personal data of hundreds or even thousands of clients. These can include email addresses and passwords, which their owners must be aware of. Our project is designed to help people in this kind of situations.

While they are not our primary audience, business partners and shareholders can also use our tool to know better about the organization they are associated with. It can help them make better business decisions based on the organization security and credibility.

Our project condenses the latest news about cyber-attacks on organizations, using a simple interface that allows to search for an organization name. We also incorporate configurable keywords that help narrowing down results and display only the relevant sections. Only registered users can access the tool. Our goal is to help clients stay informed about all of the ongoing cyber-attacks that occur against the organizations they mainly consume from. With our tool, clients can look up any organization name and read about the latest data breaches and exploits that occur in the products they use. For example, an Adobe Photoshop user can look up on Adobe and see if they had any data breach or a company that uses Microsoft Azure service can see if the service had any exploits.

**Project Structure**

Our project is a web application written in JavaScript. We use React for our front-end with components from Material-UI, to achieve a modern feel and look to the application. We also use MongoDB database to manage users. The main page asks the user to log in or register and then loads the search page. The application is built as Single-page application, meaning the search results will load up on the current page instead of on an entire new page.

For our search algorithm, we use script written in Python to scrap the latest news.

|  |  |
| --- | --- |
| Usage | Library/Package/Language |
| Framework used to write, review and edit our code. | Visual Studio Code |
| The entire web application is written in JavaScript. | JavaScript |
| Handles the front-end; user interface. | React |
| Database for managing registered users. | MongoDB |
| Handles HTTP requests. | Axios |
| Used to scrap relevant articles from the web. | Python (BeautifulSoup) |

**OSINT Project**

**OSINT project preview**

OSINT (open-source intelligence) is the collection and analysis of information that is gathered from public, or open, sources. Our OSINT tool helps the user to identify breaches and leaks without compromising and revealing sensitive information. The tool uses Peer API which provides historical data of downloading and sharing content using the BitTorrent protocol. The data is then stored in an external database.

The tool called "ips to torrents", is built with Node.js as the server-side, and React.js for the client-side.

The purpose of the tool is to extract data from external API with IP addresses and will return an indication about malicious torrents, with an algorithm to reduce false positives.

**The base tool**

First, before the creation of the main tool, we created an environment (base tool) that the main tool will use for its implementation. The base tool is a web application that uses Peer API to store and provide the information from the API. The information is stored in an external database that was built in MongoDB. The web application is divided to 2 parts, client-side and server-side. The client-side includes 3 main pages, for every aspect of the Peer API:

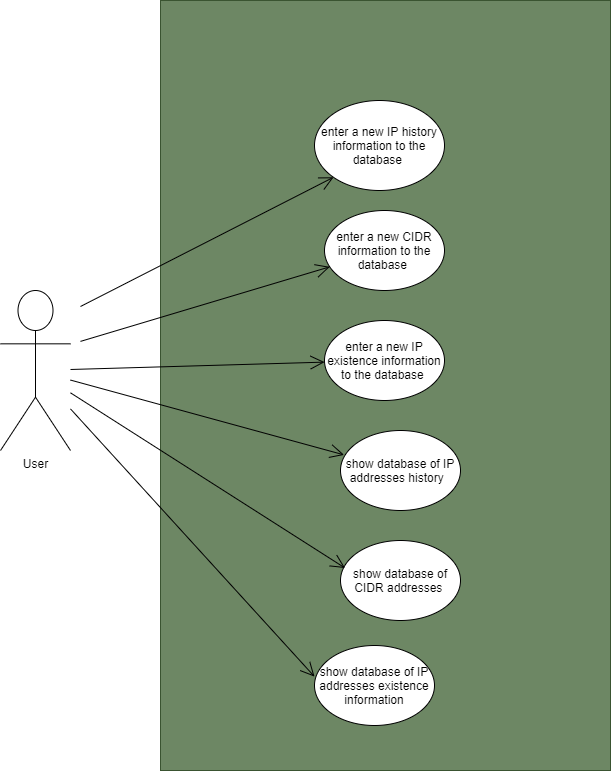
• IP history: Display information and downloaded torrents from IP entered by the user.

• IP list: Display a list of IP addresses from a CIDR entered by the user, that currently tracked by Peer API.‏

• IP exists: Display if the entered IP from the user is exists in the Peer API.

The server-side includes: connection to the Peer API, handling the database and to provide routing to get the required data.

**Base tool use case diagram**



**Base tool showcase**

