Comparative Analysis and Development of an Identity Leak Detection Tool

Ritesh Rahatal
Department of Computer Science
University of Exeter
Exeter, United Kingdom
rr519@exeter.ac.uk

Prof. Anne Kayem
Department of Computer Science
University of Exeter
Exeter, United Kingdom
A.V.Kayem@exeter.ac.uk

***Abstract*—In the modern digital landscape, the protection of Personally Identifiable Information (PII) is becoming increasingly urgent as cyberattacks grow in both frequency and sophistication. Identity leak detection tools, such as ”Have I Been Pwned” [1] and ”Identity Leak Checker” [2] are crucial for mitigating the risks associated with data breaches, alerting users to potential exposures of their sensitive information. However, the effectiveness of these tools varies widely, particularly in their ability to detect diverse forms of personal data breaches, including those involving email addresses and passwords. This project undertakes a detailed evaluation of the mechanisms used by leading identity leak detection tools, with a focus on their performance in identifying compromised email addresses and assessing password sensitivity. Through a comparative analysis, the research seeks to uncover the strengths and limitations of these tools, providing a clearer picture of their current capabilities. This analysis will also highlight existing technological and methodological gaps that could leave users vulnerable to undetected breaches. In response to these identified gaps, the project will investigate the development of a new, advanced detection tool. This proposed tool will feature innovations such as a custom-built scraping tool that autonomously gathers publicly available leaked data, stored securely in a MongoDB database. This data will form the foundation for an enhanced threat model, capable of verifying email compromises and conducting thorough password sensitivity analyses. A significant aspect of this research is the comparison between the newly developed model and existing solutions like ”Have I Been Pwned.” This comparison will rigorously assess the new model’s accuracy, security features, and user experience, aiming to demonstrate its potential as a more reliable and user-friendly alternative for identity leak detection. Ultimately, this project aspires to contribute meaningfully to cybersecurity by offering an improved tool that better safeguards online identities and sensitive information. Grounded in established cybersecurity research [3], this work aims to advance the state of identity leak detection technology and enhance the resilience of both individuals and organizations against cyber threats.**

***Index Terms*—Personal Identifiable Information (PII), Cybersecurity, Identity Leak Detection, Data Breaches, Have I Been Pwned, Cyberattacks, Tool Evaluation, User Experience**

I. Introduction

In recent years, the digital landscape has witnessed an exponential increase in data breaches, significantly affecting millions of individuals and organisations worldwide. According to a report by the Identity Theft Resource Center (ITRC), the number of data breaches in the United States alone has seen a consistent upward trend, with breaches in 2021 resulting in millions of compromised records [4]. This surge in data breaches underscores the critical need for effective identity leak detection tools. These tools are essential not only for alerting users about potential exposure of their personal information but also for initiating timely mitigating actions to protect against identity theft, financial fraud, and other consequences of data breaches. Identity leak detection tools serve as a vital component of the cybersecurity infrastructure. They function at the intersection of technology and privacy, contributing significantly to the safeguarding of personal and organisational data [5].

In today’s digital world, the protection of personal information is more critical than ever. The increasing number of data breaches requires robust identity detection tools to quickly and accurately assess the security of user credentials. This report provides an analysis of the identity detection tool developed in this project, comparing it with the existing ”Have I Been Pwned” (HIBP) service. A key feature of the developed tool is its password sensitivity integration mechanism, which checks if an email has been breached and then evaluates the sensitivity of the associated password if a breach is detected.

The increasing frequency and complexity of cyberattacks necessitate a thorough examination of the effectiveness and limitations of current identity leak detection tools. This research project is framed around two primary questions:

1. How effective is the developed threat model in identifying email breaches and assessing password sensitivity compared to existing tools like ”Have I Been Pwned”? This question aims to evaluate the developed model’s ability to accurately detect email breaches and assess password sensitivity. It will involve a comparative analysis of the model’s performance against established tools like ”Have I Been Pwned,” focusing on functionality, features, performance, security and the effectiveness of the password sensitivity check.

2. How can the integration of a scraping tool and a user interface enhance the functionality and user experience of the developed threat model? This question explores the benefits of integrating a scraping tool that collects publicly available leaked data and stores it securely in a MongoDB database. It also examines how setting up a user-friendly web interface for

data upload, processing, and storage can improve the overall functionality and user experience of the threat model.

II. Research Context

A. Related Work

The domain of identity leak detection has been extensively explored in recent years, with a significant emphasis on the development and evaluation of tools designed to alert users about breaches involving their personal information. Notable among these tools is ”Have I Been Pwned,” developed by security expert Troy Hunt. This tool aggregates data from various breaches and makes it searchable to individuals wondering if their data has been compromised [1]. Studies evaluating ”Have I Been Pwned” and similar tools typically focus on their coverage of known data breaches, the timeliness of data updates, and the ease of use of their interfaces [6].

A systematic review by [7] found that while tools like ”Have I Been Pwned” provide substantial utility in checking email and username exposures, they are less effective at capturing data related to more sensitive personal information like financial data or social security numbers. Moreover, these tools often rely on data that is publicly disclosed or available on dark web marketplaces, potentially limiting their ability to detect breaches not publicly known [8].

Research has also shown a divergence in the effectiveness of these tools across different regions and types of data breaches. For instance, tools may perform well in detecting breaches involving North American consumer data but less so with data from other regions, highlighting a geographical bias in database compositions [9].

Password managers are essential tools for maintaining strong, unique passwords across various online services. However, their convenience features can introduce vulnerabilities. The study reviewed by [10] investigates the security flaws in the autofill functionality of mobile password managers. It reveals how these vulnerabilities can be exploited by malicious applications to leak user credentials. The research identifies multiple attack vectors, including invisible fields and context manipulation, where malicious apps trick password managers into autofilling credentials into unauthorized fields. A comprehensive analysis of popular mobile password managers highlighted that some lacked adequate security measures, making them susceptible to these attacks.

Previous studies by [11] focused on encryption and storage mechanisms but did not deeply explore autofill vulnerabilities. ”AutoSpill” extends this understanding by targeting the autofill feature, emphasizing the risks associated with user interaction features in security tools. To mitigate these risks, the study recommends enhanced security checks, user interface improvements, and robust authentication mechanisms. These measures include stricter verification before autofilling, clearer user notifications, and multi-factor authentication. The implications underscore the need for continuous security assessments of password managers, balancing convenience with robust security to protect user data.

The study by [12] titled ”Usability, security and trust in password managers: A quest for user-centric properties and features” investigates password managers in detail. The authors explore how usability issues, such as complex user interfaces and poor user experience, can deter users from adopting password managers. They also highlight security concerns, including potential vulnerabilities in password storage and autofill mechanisms. Trust plays a crucial role, as users must have confidence in the password manager’s ability to safeguard their data. The study provides a comprehensive analysis of popular password managers, identifying key properties and features that enhance usability and security. Recommendations include improving user interfaces, implementing robust security measures, and fostering user trust through transparency and reliability. By addressing these factors, the research aims to contribute to the development of more user-centric password managers, ultimately enhancing their adoption and effectiveness in improving online security.

Additionally, HIBP and similar services typically require users to check email addresses and passwords separately. While HIBP provides a ”Pwned Passwords” service to check if a password has been compromised, it does not integrate this functionality seamlessly within the email breach check. This separation can lead to a fragmented user experience and may not provide immediate actionable insights about password security. The concept of password sensitivity detection, which involves assessing the strength and commonality of passwords, is crucial in enhancing the security provided by identity leak detection tools. Password sensitivity detection mechanisms aim to identify weak, common, or compromised passwords and prompt users to change them, thereby reducing the risk of unauthorized access [13].

III. Aims and Objectives

A. Aim

The aim of this project is to develop an advanced threat model designed to significantly improve the detection and mitigation of identity leaks, with a particular focus on email breach detection and password sensitivity analysis. The motivation for selecting this topic stems from the escalating frequency and sophistication of cyberattacks, which have increasingly revealed the shortcomings of existing identity leak detection tools. Current tools often lack the comprehensive functionality and user experience needed to effectively protect users from the growing threat landscape.

This project seeks to address these gaps by creating a new threat model that incorporates several key innovations. Firstly, the model will include a custom-built scraping tool capable of gathering publicly available leaked data from the web and securely storing it in a MongoDB database. This database will serve as the backbone for the threat model, enabling it to verify whether a user’s email has been compromised.In addition to the technical advancements, the project will also focus on enhancing the user experience by integrating a user-friendly web interface.

A critical component of this project is a detailed comparative analysis of the proposed threat model against existing tools, such as ”Have I Been Pwned.” This analysis will rigorously evaluate the effectiveness of the new model in several key areas, including the accuracy of email breach detection, the robustness of password sensitivity checks, the overall functionality, feature and security of the system, as well as the user experience provided by the interface. By comparing these factors, the project aims to demonstrate the superior capabilities of the new model and its potential to offer a more reliable and user-friendly solution for protecting against identity leaks.

Ultimately, this project aspires to make a significant contribution to the field of cybersecurity by offering an improved tool for safeguarding individuals' online identities and sensitive information. The enhanced threat model, with its integrated scraping tool and user-centric interface, is designed to provide a more comprehensive and effective defence against the growing threats posed by cybercriminals.

B. Objectives

1) Develop a Threat Model: The primary objective of developing a threat model is to create a robust system that can effectively assess and mitigate security risks associated with user credentials. This threat model operates in two key phases:

a) Email Breach Verification: The model begins by verifying whether an email address has been compromised in any known data breaches. This is accomplished by cross-referencing the email with a comprehensive database of breached accounts. Such a database typically includes information from major security incidents where user credentials were exposed. The goal is to determine if the email has ever been leaked in past breaches, which could indicate a higher risk of being targeted in subsequent attacks.

b) Password Sensitivity Analysis: If the email is found to have been breached, the next step is to assess the sensitivity of the associated password. The model evaluates whether the password is commonly used across different platforms or if it has characteristics that make it easily guessable (e.g., simple patterns, dictionary words). Common passwords are especially vulnerable to attacks like credential stuffing, where attackers use lists of commonly used passwords to gain unauthorized access to accounts. By identifying such weaknesses, the model can inform users about the risks associated with their current password.

Additionally, the model doesn't just stop at identifying vulnerabilities; it provides actionable recommendations. If a password is deemed common or weak, the user is advised to change it immediately. The model offers guidelines on how to create a stronger, more unique password. This proactive approach not only helps users respond to immediate threats but also educates them on best practices for password management, ultimately contributing to improved long-term security.

2) Build a Scraping Tool: The objective here is to develop a specialized tool that can autonomously collect and manage data from publicly available sources, focusing on leaked credentials. The tool operates in several stages:

a) Data Collection: The scraping tool is designed to search the web for publicly available leaked data, often found on forums, paste sites, and other platforms where compromised credentials are shared. The tool needs to be equipped with the ability to navigate and extract data from these sources efficiently, ensuring that it gathers all relevant information that could be used to enhance the threat model.

b) Data Storage: Once collected, the data is stored securely in a MongoDB database. MongoDB is chosen due to its ability to handle large volumes of unstructured data, which is typical in the context of leaked credentials. The database serves as a central repository, where the scraped data is indexed and organized in a way that makes it readily accessible for further analysis. Security is a critical aspect here, as the database may contain sensitive information that needs to be protected from unauthorized access.

c) Integration with the Threat Model: The data stored in MongoDB is then used to inform the threat model. The threat model queries the database to check if an email or password has been part of any recent breaches. This integration ensures that the threat model is always working with the most up-to-date information, allowing for more accurate and timely risk assessments.

d) User Interface Development: Alongside the scraping tool, a user-friendly web interface is developed. This interface allows users to interact with the system directly. For instance, users can upload a text file containing a list of credentials that they want to check against the database. The system processes the uploaded file, extracts the relevant data, and stores it in the MongoDB database for analysis.

The web interface also serves as a portal through which users can receive feedback from the threat model. After processing the data, the system provides users with detailed reports on the security status of their credentials, including whether they have been breached and the sensitivity of their passwords. This user-centric approach not only makes the system accessible to non-technical users but also ensures that they receive actionable insights that can help them improve their security posture.

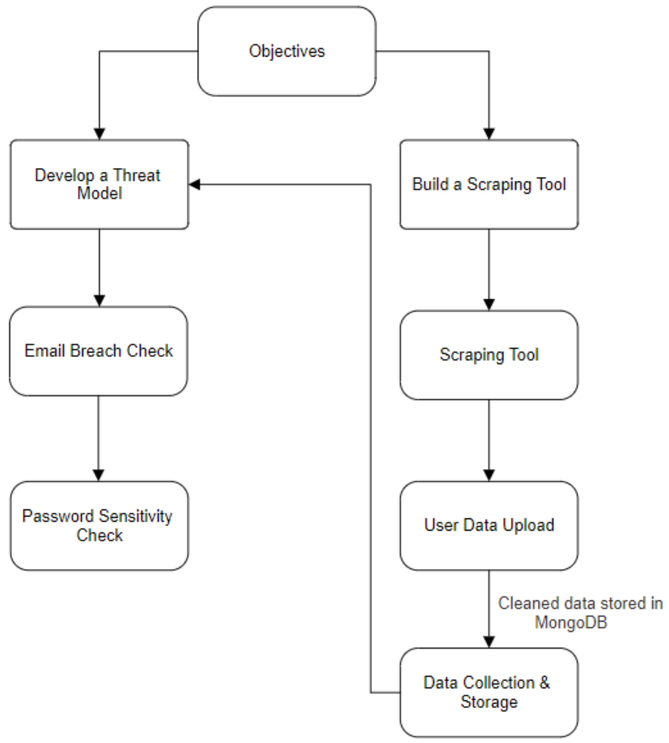


Fig. 1. Flowchart illustrating the workflow and key components of the developed system

IV. Data and Resources

A. Data Sets

For this project, publicly available datasets have been utilized, which can be accessed from [14]. From these datasets, 10.2 million email IDs with their corresponding passwords have been scraped and stored securely in a MongoDB database. This extensive dataset is used to inform and enhance the developed threat model, ensuring comprehensive coverage and accurate detection of breaches and password sensitivities.

For the password sensitivity integration mechanism, I have used a dataset from [15]. This dataset contains the top 100,000 passwords from the ”Have I Been Pwned” project.

The MongoDB database containing this data is available for review and use in further research. It can be accessed and downloaded in JSON format from this link [16]. This dataset plays a crucial role in the development and enhancement of the threat model. By analyzing the patterns and frequencies of passwords in real breaches, the model can more accurately predict and detect potential vulnerabilities in password choices. Moreover, it allows for the identification of common password practices that may leave users vulnerable to attacks.

Together, these datasets form the foundation of a comprehensive threat detection system designed to enhance cybersecurity by proactively identifying and mitigating risks associated with password breaches.

V. System Implementation and Data Management

A. Threat Model Development

1. Email Breach Check: The system uses a locally stored database of breached emails to verify if a given email has been compromised. This database is regularly updated with data scraped from public sources.

2. Password Sensitivity Check: Upon detecting a breach, the system checks the associated password against a list of common passwords stored in MongoDB. If the password is common, it flags it as sensitive and recommends the user change it immediately. This helps users mitigate potential security risks promptly.

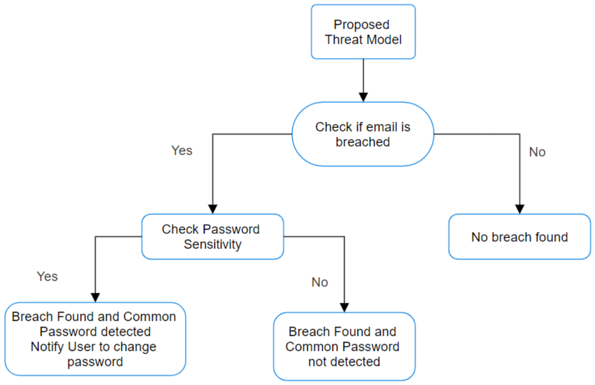

Proposed Threat Model
Check if email is breached
Yes
No
Check Password Sensitivity
No breach found
Yes
No
Breach Found and Common Password detected Notify User to change password
Breach Found and Common Password not detected

Fig. 2. Flowchart illustrating Proposed threat model's workflow

B. Data Collection and Storage

1. Scraping Tool: A specialized tool systematically collects leaked data from publicly available sources. The collected data is securely stored in a MongoDB database for further analysis and use in the threat model.

2. Data Security: Ensuring the security of the collected data is paramount. The database is protected with stringent access controls and encryption measures, safeguarding it against unauthorized access and potential breaches. This ensures the data remains comprehensive, up-to-date, and secure.

C. Technology Used

The technologies used in my project are:

1. Python : The primary programming language used for coding the project. Python's versatility and extensive libraries make it ideal for data processing, analysis, and building machine learning models. The development and testing were conducted in Jupyter Notebook, which provides an interactive environment for writing and executing code.

2. MongoDB : MongoDB was utilised as the database for storing and managing the scraped breach data. Its flexibility and scalability make it suitable for handling large volumes

of unstructured data, such as the 10 million email IDs and passwords collected for this project.

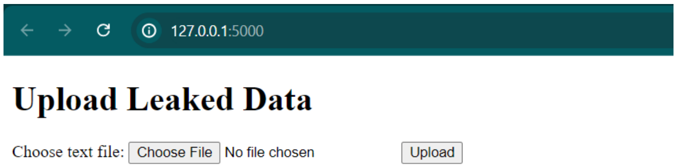
3. Flask : Flask, a lightweight web framework for Python, was used to develop the web pages and the user interface for the project. It allows for the easy creation of web applications, enabling users to upload text files, process data, and interact with the threat model through a user-friendly interface.

4. HTML : HTML (HyperText Markup Language) was used to structure the web pages of the application. It provides the foundation for the web interface, allowing for the creation of forms, tables, and other elements needed for user interaction and data presentation.

VI. Experiment Design and Methods

A. Building of Scraping Tool

Creation of a web page for uploading the leaked data. The process involves reading, cleaning, and securely storing the data in MongoDB.



127.0.0.1:5000

Upload Leaked Data

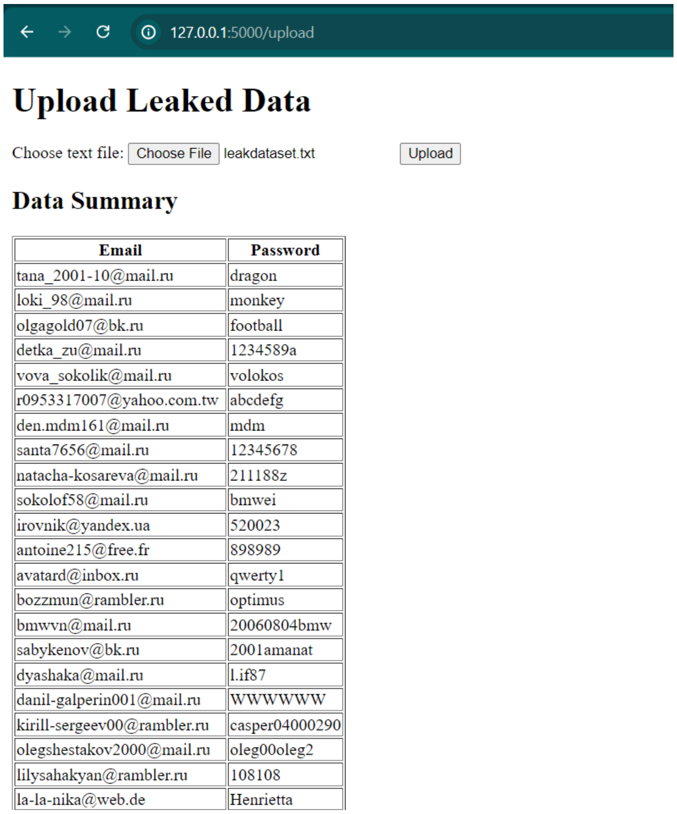
Choose text file: No file chosen

Fig. 3. Web page for uploading leaked data.

The backend process for data upload in a Flask web application involves several steps:

1) The application runs in debug mode at http://127.0.0.1:5000/.
2) When a user accesses the root URL, a GET request is received, and the server responds with a 200 status code, indicating success.
3) A subsequent GET request for the favicon results in a 404 status code as the file is not found.
4) When a user uploads data via a POST request to the /upload endpoint, the server processes the request successfully, returning a 200 status code.
5) The uploaded file, `leakdataset.txt`, is processed by loading its contents into a pandas DataFrame, displaying data in email and password columns.
6) Finally, the data from the DataFrame is inserted into MongoDB for storage.

This process is depicted through the following user interfaces and backend processes:



127.0.0.1:5000/upload

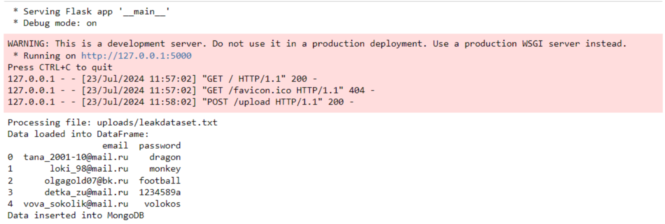
Upload Leaked Data

Choose text file: leakdataset.txt

Data Summary

Email	Password
tana_2001-10@mail.ru	dragon
loki_98@mail.ru	monkey
olgagold07@bk.ru	football
detka_zu@mail.ru	1234589a
vova_sokolik@mail.ru	volokos
r0953317007@yahoo.com.tw	abcdefg
den.mdm161@mail.ru	mdm
santa7656@mail.ru	12345678
natacha-kosareva@mail.ru	211188z
sokolof58@mail.ru	bmwei
irovnik@yandex.ua	520023
antoine215@free.fr	898989
avatard@inbox.ru	qwerty1
bozzmun@rambler.ru	optimus
bmwvn@mail.ru	20060804bmw
sabykenov@bk.ru	2001amanat
dyashaka@mail.ru	l.if87
danil-galperin001@mail.ru	WWWWWW
kirill-sergeev00@rambler.ru	casper04000290
olegshestakov2000@mail.ru	oleg00oleg2
lilysahakyan@rambler.ru	108108
la-la-nika@web.de	Henrietta

Fig. 4. User Interface for uploading data

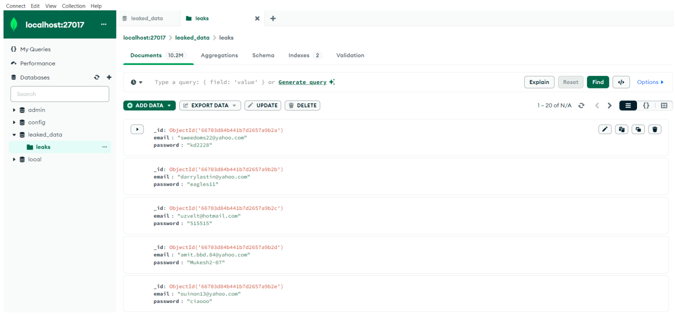


```
 * Serving Flask app '__main__'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [23/Jul/2024 11:57:02] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Jul/2024 11:57:02] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [23/Jul/2024 11:58:02] "POST /upload HTTP/1.1" 200 -
Processing file: uploads/leakdataset.txt
Data loaded into DataFrame:
                  email  password
0  tana_2001-10@mail.ru    dragon
1       loki_98@mail.ru    monkey
2      olgagold07@bk.ru  football
3      detka_zu@mail.ru  1234589a
4  vova_sokolik@mail.ru   volokos
Data inserted into MongoDB
```

Fig. 5. Backend Process for data upload

B. MongoDB Database Storage

Around 10.2 million leaked data records are scraped and stored in MongoDB for secure and efficient access. The following figure illustrates the storage setup:

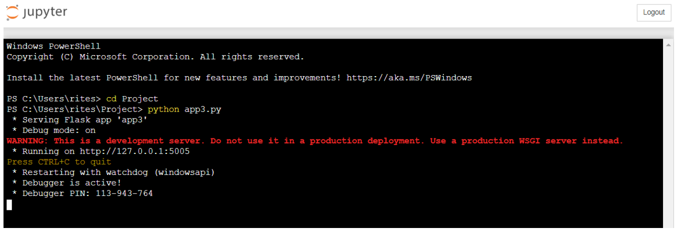


```
_id: ObjectId('66703d84b441b7d2657a9b2a')
email : "sweedoms22@yahoo.com"
password : "kd2228"

_id: ObjectId('66703d84b441b7d2657a9b2b')
email : "darrylastin@yahoo.com"
password : "eagles11"

_id: ObjectId('66703d84b441b7d2657a9b2c')
email : "uzvelt@hotmail.com"
password : "515515"

_id: ObjectId('66703d84b441b7d2657a9b2d')
email : "amit.bbd.84@yahoo.com"
password : "Mukesh2-07"

_id: ObjectId('66703d84b441b7d2657a9b2e')
email : "ouinon13@yahoo.com"
password : "ciaooo"
```

Fig. 6. MongoDB storage setup.

C. *Proposed Threat Model (Developed Tool)*

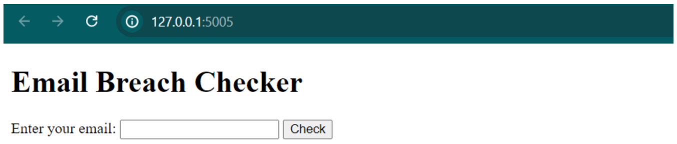
To navigate to the project directory, the command `cd Project` is used, which changes the current working directory to the ”Project” folder. The application is started with the command `python app3.py`, which launches the Flask application. The server runs in debug mode and is accessible at http://127.0.0.1:5005/, demonstrating the Flask development server running within a Jupyter notebook environment.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\rites> cd Project
PS C:\Users\rites\Project> python app3.py
 * Serving Flask app 'app3'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5005
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 113-943-764
```

Fig. 7. Launching the Flask application.

This setup integrates a Flask backend with a simple frontend to build an interactive web application. It uses the uploaded dataset of compromised emails and passwords to verify user inputs. When a user enters their email address, the application checks if it has been compromised and evaluates the security of the associated password. The application provides users with valuable information and recommendations, helping them enhance their account security.



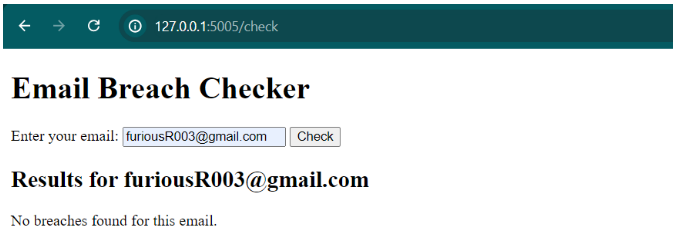
Email Breach Checker

Enter your email: Check

Fig. 8. Verifying user inputs in the web application.

Examples of the application’s functionality include:

- Checking an email that is safe and not breached:



Email Breach Checker

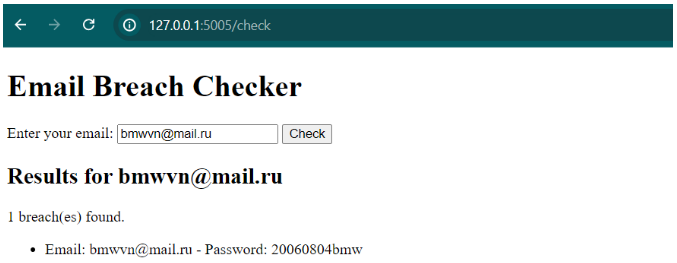
Enter your email: furiousR003@gmail.com Check

Results for furiousR003@gmail.com

No breaches found for this email.

Fig. 9. Email not breached scenario.

- Identifying a breached email without a common password warning:



Email Breach Checker

Enter your email: bmwvn@mail.ru Check

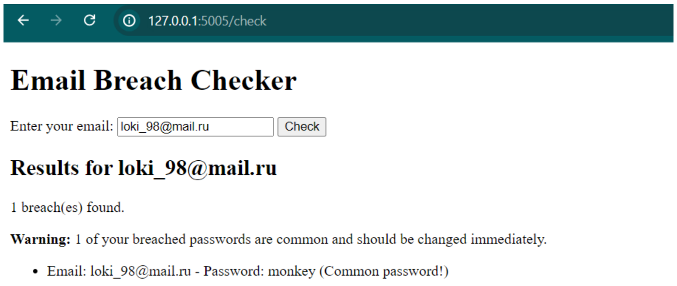
Results for bmwvn@mail.ru

1 breach(es) found.

- Email: bmwvn@mail.ru - Password: 20060804bmw

Fig. 10. Email breached, no common password warning.

- Alerting a user to change a compromised and common password:



Email Breach Checker

Enter your email: loki_98@mail.ru Check

Results for loki_98@mail.ru

1 breach(es) found.

Warning: 1 of your breached passwords are common and should be changed immediately.

- Email: loki_98@mail.ru - Password: monkey (Common password!)

Fig. 11. Email breached with common password warning.

D. Comparison of Functionality and Features

The Proposed developed tool and "Have I Been Pwned" (HIBP) both serve as critical resources for detecting email breaches, but they implement this functionality through markedly different approaches. These differences manifest in their use of data sources, performance characteristics, and security measures, each impacting the overall effectiveness and suitability for various users and organizations.

1) Email Breach Check Mechanism: Both the Proposed developed tool and HIBP provide email breach detection capabilities, yet their methodologies diverge significantly. HIBP operates on a centralized, external database that aggregates data from a vast array of known breaches, including high-profile incidents that have exposed millions of user credentials. This database is meticulously maintained and regularly updated by HIBP's administrators, ensuring that it reflects the latest information available. Users interact with this database via API calls, allowing them to check if their email addresses have been compromised. This centralized approach offers extensive data coverage and simplicity of use, making HIBP a widely recognized and trusted service. In contrast, the Proposed developed tool employs a locally stored database within MongoDB. This database is periodically updated with data scraped from publicly available sources, which could include lesser-known breaches that are particularly relevant to the user or organization. The local database setup provides greater flexibility in data management, allowing users to tailor the database content to their specific needs, such as focusing on industry-specific breaches or incorporating custom lists of compromised credentials. Additionally, the Proposed developed tool integrates a password sensitivity analysis as part of the breach detection process. This feature assesses whether the password associated with a breached email is commonly used or easily guessable, thereby adding an extra layer of security within the breach detection workflow.

2) Data Sources and Updates: HIBP's reliance on an external, centralized database accessed via API is both an asset and a limitation. On the one hand, it grants users access to an extensive and well-curated set of breach data, covering a broad spectrum of security incidents globally. The centralized nature of the database ensures systematic updates, maintaining a high level of accuracy and reliability. However, this model also introduces dependencies on the API's response times and the rate at which HIBP administrators can update the database, potentially leading to delays in reflecting the most recent breaches. The Proposed developed tool, on the other hand, leverages a local database approach. This allows for more frequent and flexible updates, as users can incorporate new breach data as soon as it becomes available, without waiting for third-party updates. This is particularly advantageous for organizations that need to respond quickly to emerging threats or require tailored data to meet specific security needs. The ability to customize the database, such as by adding industry-specific breach information or updating the list of common passwords, ensures that the Proposed developed tool can be fine-tuned to the unique requirements of its users.

3) Performance Considerations: The performance of these tools is another critical area of differentiation. HIBP's API-based system, while robust, is subject to the inherent limitations of network-based operations. Each request to check an email or password against the breach database involves network latency, as the request must travel over the internet and depend on the response time of HIBP's servers. Although HIBP is optimized to handle large volumes of queries efficiently, factors such as server load, network conditions, and API rate limits can impact performance, particularly during peak usage times. The Proposed developed tool mitigates these performance issues by performing all operations locally. By housing the breach data within a MongoDB database on the user's system, the tool eliminates the latency associated with network communication, leading to faster and more consistent response times. MongoDB's indexing and querying capabilities can be optimized to further enhance performance, enabling the Proposed developed tool to process large datasets quickly and efficiently. This local processing advantage is particularly beneficial in scenarios where immediate feedback is essential, such as in automated security systems or environments with stringent response time requirements.

4) Security Measures: Security is a paramount concern for both tools, though their approaches are tailored to their respective architectures. HIBP secures its API communication channels using encrypted endpoints, which protects the data as it is transmitted between the user's system and HIBP's servers. Additionally, HIBP often anonymizes responses, reducing the amount of sensitive data that is exposed or retained. However, because HIBP operates as an external service, there is an inherent reliance on the security practices of the service provider. Users must trust that HIBP's infrastructure is secure and that their data is handled appropriately. The Proposed developed tool offers a different security paradigm by keeping all data management and processing within a controlled local environment. This local approach significantly reduces the attack surface by minimizing the number of external dependencies and limiting data exposure to external networks. The Proposed developed tool employs strict access controls, ensuring that only authorized personnel can interact with the sensitive data stored in the MongoDB database. Furthermore, the tool can implement robust encryption protocols for data at rest, ensuring that even if unauthorized access occurs, the data remains secure. This approach provides users with greater control over their security environment, reducing the risks associated with relying on third-party services.

E. Enhanced Password Sensitivity Integration

1) Integrated Workflow: HIBP requires users to perform separate checks for email breaches and password breaches. Users must first query the system to check if their email has been involved in any known data breaches. To check if their password has been compromised, they must use the "Pwned Passwords" service, which involves a separate query. My proposed model consolidates these checks into a single, seamless workflow. This integrated approach reduces the complexity for users by allowing them to perform both email breach checks and password sensitivity analysis simultaneously. This means users are immediately informed about the breach status of their email and the sensitivity of their password in one streamlined process. This not only saves time but also reduces the chances of users neglecting one aspect of their security (e.g., checking the password but not the email, or vice versa).

2) Contextual Sensitivity Analysis: HIBP checks whether a password has been found in its database of breached passwords but does not provide any additional context regarding the sensitivity of the password, such as whether it is commonly used or carries specific risks. My proposed model goes beyond simple breach detection by providing a contextual sensitivity analysis of the password. It checks whether the password is commonly used and informs the user of the associated risks. By understanding the context in which a password is commonly targeted, users receive actionable advice to improve their security posture. This includes not just being notified of a breach but also understanding why certain passwords are more vulnerable and how to choose a more secure alternative.

3) Customizable Sensitivity Criteria: HIBP relies on a static list of breached passwords, which, while comprehensive, does not adapt to new data trends or allow for customization based on specific user needs or emerging security threats. My proposed model introduces customizable sensitivity criteria, allowing the password list to be tailored and updated based on the latest trends, data, and specific organizational or user requirements. This ensures that the sensitivity analysis remains relevant and up-to-date, aligning with current security best practices. The ability to adapt to new trends and emerging threats provides a dynamic and responsive security measure, offering a higher level of protection.

4) Immediate Feedback and Recommendations: HIBP provides users with feedback on whether their password has been found in its breach database but does not directly offer recommendations for mitigating the associated risks. My proposed model offers immediate feedback on both the breach status and the sensitivity of the password. If a password is deemed too common or weak, users are advised to change it and are provided with guidelines on selecting a more secure alternative. This proactive approach ensures that users are not only informed of potential vulnerabilities but are also guided on how to address them promptly, enhancing their overall security.

5) Enhanced Security Awareness: HIBP focuses primarily on breach notification, alerting users when their credentials have been compromised, but it does not significantly contribute to broader security awareness or education. My proposed model takes a more holistic approach by enhancing overall security awareness. It educates users on the importance of strong, unique passwords and promotes better password hygiene through its sensitivity analysis. By flagging common or weak passwords, the model encourages users to adopt proactive security measures, such as regularly updating passwords and avoiding easily guessable ones. This educational component is crucial in fostering a security-conscious mindset, ultimately leading to better protection against potential attacks.

VII. Results

Testing Results : A sample of 100 email IDs, including both PII (Personally Identifiable Information) and non-PII breached emails, was tested. The dataset was sourced from Kaggle, an open-source platform.

Tested data: `testing_results.csv`

In terms of accuracy, the HIBP Model achieves an accuracy of 91.92%. whereas my Proposed model demonstrates a higher accuracy, reaching 94.95%. When considering the average time taken for detection, the HIBP Model averages at 0.249 seconds. My proposed model, however, is not only more accurate but also faster, with an average detection time of 0.202 seconds.

Model	Accuracy (%)	Avg. Detection Time (s)
HIBP Model	91.92	0.249
My Proposed Model	94.95	0.202

TABLE I
Comparison of Accuracies and Detection Times between HIBP Model and My Proposed Model

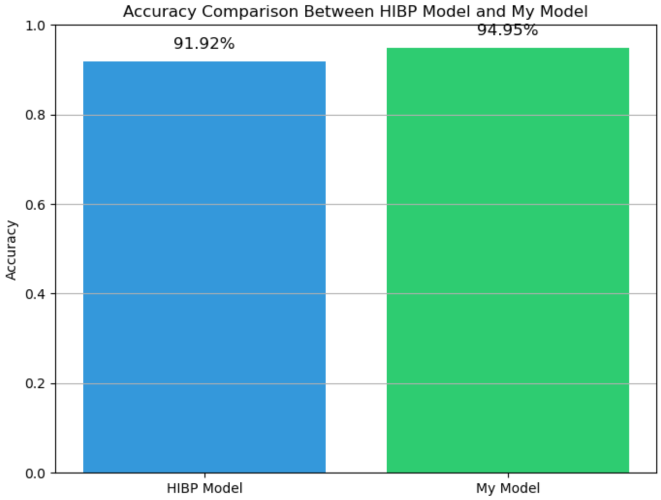


Fig. 12. Comparison of model accuracies for breach detection

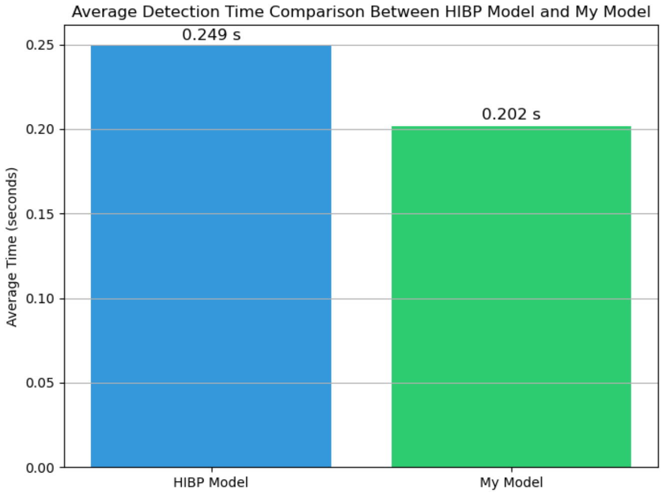


Fig. 13. Average time taken by each model for breach detection

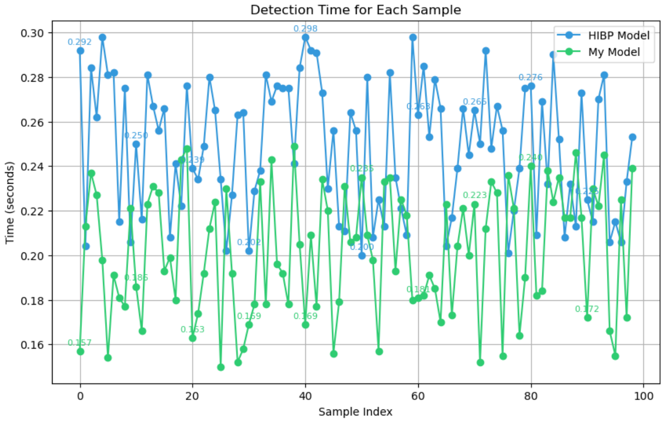


Fig. 14. Line graph showing detection time of the models

The line graph shows that my proposed model consistently detects breaches faster than the HIBP Model across all samples, indicating higher efficiency. Both models exhibit consistent performance, with my model displaying less variability in detection times. Occasional spikes in times are seen in both models, but my proposed model generally maintains quicker responses. Overall, my Proposed model provides a more efficient and stable detection time compared to the HIBP Model.

VIII. Future Work

As the threat model continues to evolve, several areas have been identified for future work to ensure the system remains robust, accurate, and user-friendly. These improvements will help maintain the model's relevance in the face of an ever-changing cybersecurity landscape.

1. Regular Updates To maintain the accuracy and reliability of the threat model, it is crucial to ensure that the breach data and the list of common passwords are regularly updated. Cyber threats are constantly evolving, with new breaches and password vulnerabilities emerging frequently. Implementing automated scripts that periodically fetch new data from reliable sources and update the MongoDB database is essential. This process can be automated using scheduling tools like cron jobs, reducing the need for manual intervention and ensuring that the system always works with the most current information. Regular updates will enhance the model's ability to identify emerging threats and provide users with timely warnings, contributing to more effective cyber threat detection and prevention [17]

2. Optimize MongoDB Queries As the database grows in size and complexity, it is essential to continuously optimize MongoDB queries to maintain performance. Slow query times can hinder the effectiveness of the threat model, especially when processing large datasets. Future work will focus on fine-tuning the database by maintaining proper indexing, optimizing query structures, and possibly implementing sharding for horizontal scaling as needed. Monitoring tools will be deployed to track query performance, identify bottlenecks, and make adjustments in real-time. Continuous optimization will ensure that users experience fast and reliable responses when interacting with the system, even as the data volume increases, thereby maintaining the overall efficiency and usability of the threat model [18]

3. Enhance User Interface Improving the user interface (UI) is another critical area of future work. A well-designed UI is essential for making the threat model accessible to a broader audience, including users who may not have technical expertise. Enhancements will focus on refining the interface based on user feedback, which will be collected through surveys, usability testing, and in-app feedback mechanisms. These improvements could include more intuitive navigation, clearer data visualization, and personalized user settings. By prioritizing the user experience, the threat model can become a more valuable tool for a diverse range of users, ultimately increasing its adoption and impact. This approach aligns with established principles of usability engineering and web usability, ensuring the interface meets high standards of accessibility and user satisfaction [19]

4. Expand Data Coverage To improve the comprehensiveness of the password sensitivity checks, it is important to broaden the scope of the common passwords list. This can be achieved by incorporating data from recent breaches and other reputable sources, ensuring that the model stays up-to-date with the latest password trends. Collaborating with cybersecurity communities, such as forums, research groups, and professional networks, can provide access to the most current data and insights. Expanding data coverage will not only improve the accuracy of the password sensitivity analysis but also enhance the model's ability to detect and mitigate emerging threats. Incorporating a diverse set of password data will strengthen the model's ability to provide relevant and

effective security recommendations [20]

5. Strengthen Security Measures As the threat model handles sensitive user data, strengthening local security measures is a priority. Future work will involve continuously enhancing the security protocols to protect the integrity of password sensitivity checks and stored data. This includes implementing advanced encryption methods, secure access controls, and regular security audits to identify and address vulnerabilities. Adopting industry best practices for data protection, such as using encryption both at rest and in transit, will ensure that user data remains secure. Additionally, regular penetration testing and threat modeling exercises will be conducted to simulate potential attacks and reinforce the system's defenses. These measures will align with the latest standards in cryptography and secure coding practices, ensuring that the threat model remains resilient against evolving cyber threats [21]

IX. Data Governance and Ethics

The ethical considerations and legal compliance associated with data scraping and the usage of personal information in identity leak detection tools are critical, especially in light of stringent data protection laws such as the General Data Protection Regulation (GDPR) in the EU and the California Consumer Privacy Act (CCPA) in the US. This project prioritizes consent and transparency by ensuring that, where applicable, explicit consent is obtained from users before processing their data. Users are informed clearly about what data is collected, how it is used, and their rights concerning their data, thereby enhancing operational transparency [22]. In terms of legal compliance, the tools developed adhere to GDPR principles, including lawfulness, fairness, and transparency, while also respecting rights such as data portability and the right to be forgotten. Furthermore, the project conducts regular ethical impact assessments to evaluate the potential consequences of data scraping activities, ensuring that these practices do not adversely affect individuals whose data is involved [23]. These measures collectively demonstrate a strong commitment to ethical governance and compliance with relevant data protection laws.

X. Risk Assessment

During the development of the identity detection tool, several potential risks were considered. A key concern was data quality and reliability, given the reliance on publicly available datasets. Publicly sourced data can vary in quality, completeness, and accuracy, potentially impacting the effectiveness of the threat model. To mitigate this risk, a comprehensive data validation process was implemented, including the removal of duplicates, verification against multiple sources, and regular updates to the data sets. User privacy was a critical concern, especially since the tool deals with sensitive information such as email addresses and passwords. Measures were taken to anonymise data wherever possible and ensure that user data was protected throughout the system. Legal and ethical considerations were also taken into account. The project ensured compliance with all relevant legal frameworks and ethical standards, particularly concerning the use of publicly available data. Secure storage and transmission protocols were implemented to safeguard data privacy and prevent unauthorized access.

Finally, the risk of technological obsolescence was acknowledged, given the rapidly evolving nature of cybersecurity threats and technologies. To mitigate this, the project was designed with flexibility in mind, allowing for easy updates and integrations of new technologies and methodologies as they emerge. Regular review and updates to the system were planned to keep the tool relevant and effective against new threats. By addressing these risks through careful planning and proactive measures, the project aimed to deliver a reliable, secure, and user-friendly identity detection tool.

XI. Conclusion

The comparative analysis highlights the strengths and unique features of the developed identity detection tool, particularly its integrated password sensitivity check. This feature distinguishes it from existing tools like ”Have I Been Pwned” (HIBP), which, although comprehensive and widely recognized, lacks an integrated mechanism to evaluate password sensitivity directly within the breach detection process.

The developed tool not only verifies if an email has been breached but also assesses the sensitivity of the associated password. If the password is found to be common or weak, the user is immediately informed and advised to change it. This proactive approach enhances the overall security by addressing one of the most critical aspects of personal data protection—password strength. Furthermore, the tool offers faster, localised processing due to its use of a MongoDB database for storing and managing the scraped breach data. This ensures that the tool operates efficiently and effectively, even with large datasets. The integration of a web interface using Flask allows users to upload text files, process data, and interact with the tool seamlessly, enhancing the user experience.

By combining these features with robust data handling and advanced processing capabilities, the developed tool provides a comprehensive and user-friendly solution for identity detection and password security. The incorporation of the password sensitivity check, in particular, represents a significant advancement over existing tools, offering users a more detailed and actionable security assessment. This positions the developed tool as a superior choice for individuals and organisations seeking to safeguard their digital identities against the increasing threat of cyberattacks.

Declaration of Originality and Citation

I hereby declare that the work presented in this project report is solely my own. All sources of information and literature that have been utilised in this project have been acknowledged through proper citations. Any assistance received in the course of this project has also been acknowledged.

I have ensured that the research, analysis, and conclusions drawn in this project reflect my individual efforts and intellectual contributions. By adhering to these practices, I have endeavoured to produce an original and authentic piece of work that adheres to the ethical standards of academic research.

References

[1] T. Hunt, "Have I Been Pwned," Website, 2013. [Online]. Available: https://haveibeenpwned.com/. [Accessed: Aug. 3, 2024].

[2] "Identity Leak Checker," 2014. [Online]. Available: https://sec.hpi.de/ilc/. [Accessed: Aug. 3, 2024].

[3] R. Ávila, R. Khoury, R. Khoury, and F. Petrillo, "Use of Security Logs for Data Leak Detection: A Systematic Literature Review," *Secur. Commun. Networks*, vol. 2021, pp. 6615899:1–6615899:29, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232225538.

[4] Identity Theft Resource Center, "Annual Data Breach Report," 2022. [Online]. Available: https://www.idtheftcenter.org/publication/2022-data-breach-report/. [Accessed: Aug. 3, 2024].

[5] X. Shu, D. D. Yao, and E. Bertino, "Privacy-Preserving Detection of Sensitive Data Exposure," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, pp. 1–1, May 2015. doi: 10.1109/TIFS.2015.2398363.

[6] P. Mayer, Y. Zou, F. Schaub, and A. J. Aviv, "Now I'm a bit angry: Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them," 2023. [Online]. Available: https://www.ftc.gov/system/files/documents/public_events/1582978/now_im_a_bit_angry_-_individuals_awareness_perception_and_responses_to_data.pdf.

[7] P. Mayer, Y. Zou, B. M. Lowens, H. A. Dyer, K. Le, F. Schaub, and A. J. Aviv, "Awareness, Intention, (In)Action: Individuals' Reactions to Data Breaches," *ACM Trans. Comput.-Hum. Interact.*, vol. 30, no. 5, pp. 1–53, Sep. 2023. doi: 10.1145/3589958.

[8] L. Cheng, F. Liu, and D. D. Yao, "Enterprise data breach: causes, challenges, prevention, and future directions," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 7, no. 6, pp. e1211, Jun. 2017. doi: 10.1002/widm.1211.

[9] M. Barati and B. Yankson, "Predicting the Occurrence of a Data Breach," *Int. J. Inf. Manag. Data Insights*, vol. 2, pp. 100128, Nov. 2022. doi: 10.1016/j.jjimei.2022.100128.

[10] A. Gangwal, S. Singh, and A. Srivastava, "AutoSpill: Credential Leakage from Mobile Password Managers," in *Proc. 13th ACM Conf. Data Appl. Secur. Privacy*, Charlotte, NC, USA, 2023, pp. 39–47. doi: 10.1145/3577923.3583658.

[11] A. Alrushaid and R. Algarawi, "Security Analysis on Password Managers Applications," 2020. [Online]. Available: https://www.researchpublish.com/upload/book/paperpdf-1600434399.pdf.

[12] S. Chaudhary, T. Schafeitel-Täntinen, M. Helenius, and E. Berki, "Usability, security and trust in password managers: A quest for user-centric properties and features," *Comput. Sci. Rev.*, vol. 33, pp. 69–90, Aug. 2019. doi: 10.1016/j.cosrev.2019.03.002.

[13] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," in *Proc. 2012 IEEE Symp. Secur. Priv.*, 2012, pp. 553–567. doi: 10.1109/SP.2012.44.

[14] "Collection1," Internet Archive. [Online]. Available: https://archive.org/download/Collection1_201901. [Accessed: May. 29, 2024].

[15] "Pwned Passwords Top 100k," National Cyber Security Centre. [Online]. Available: https://www.ncsc.gov.uk/static-assets/documents/PwnedPasswordsTop100k.txt?ref=troyhunt.com. [Accessed: June. 26, 2024].

[16] "Scraped Leak Data," Google Drive. [Online]. Available: https://drive.google.com/file/d/1YiUjud9P6t44gultkjzJkBukugWQ3RdA/view?usp=sharing.

[17] S. Saeed, S. A. Suayyid, M. S. Al-Ghamdi, H. Al-Muhaisen, and A. M. Almuhaideb, "A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience," *Sensors*, vol. 23, no. 16, article 7273, 2023. doi: 10.3390/s23167273.

[18] A. Karras, C. Karras, D. Samoladas, K. Giotopoulos, and S. Sioutas, "Query Optimization in NoSQL Databases Using an Enhanced Localized R-tree Index," in *Proc. 2022 Int. Conf. Data Science and Knowledge Engineering for Sensing Decision Support*, 2022, pp. 372-385. doi: 10.1007/978-3-031-21047-1_33.

[19] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. ISBN: 9780080520292.

[20] P. Gasti and K. Rasmussen, "On the Security of Password Manager Database Formats," in *Proc. 19th ACM Conf. Computer and Communications Security*, 2012. doi: 10.1007/978-3-642-33167-1_44.

[21] M. Howard, D. Leblanc, and B. Valentine, *Writing Secure Code*. USA: Microsoft Press, 2001. ISBN: 0735615888.

[22] T. Zarsky, "Transparency in Data Mining: From Theory to Practice," in *Studies in Applied Philosophy, Epistemology and Rational Ethics*, pp. 301–324, 2013. doi: 10.1007/978-3-642-30487-3_17.

[23] V. Krotov, L. Johnson, and L. Silva, "Legality and Ethics of Web Scraping," *Commun. Assoc. Inf. Syst.*, vol. 47, pp. 539–563, Jan. 2020. doi: 10.17705/1CAIS.04724.