National College of Ireland

Enhancing Cybersecurity Posture through Web-based Automated Google Dorking

MSc Research Project
MSc in Cybersecurity

Diti Majithia
Student ID: x22198083

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Diti Majithia ……

Student ID: ……x22198083……

Programme: ……MSc in Cybersecurity…… **Year:** …2024……..

Module: ……Practicum……

Supervisor: ……Vikas Sahni……

Submission Due Date: ……12/08/2024……

Project Title: …Enhancing Cybersecurity Posture through Web-based Automated Google Dorking…….

Word Count: ……7880…… **Page Count**……22……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …… Diti ……

Date: ……12/08/2024……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Cybersecurity Posture through Web-based Automated Google Dorking

Diti Majithia
X22198083

Abstract

The growing complexity of cyber threats drove the realization that the tools used in vulnerability assessment had to be more efficient. One of the principal tools, used only by security researchers, was identified as Google Dorking. Executed manually, it was slow and error-prone. The WAGDT - Dorkinator project was undertaken to address the disadvantages of manual methods by introducing a user-friendly automated solution. The tool was tested for effectiveness by comparing it with a different existing tool in terms of accuracy, efficiency, and user satisfaction. The Dorkinator project was developed to provide augmentation in digital reconnaissance by decreasing manual input and increasing security posture. While designing the tool, emphasis was given to automated complex Google Dorking queries. Thus, it would simplify the information-gathering phase in penetration testing. Implemented using efficient and scalable technologies at the time of its development, the tool focuses on user-friendly design and real-time data processing. In terms of evaluation, Dorkinator demonstrated higher usability, faster execution of queries, and high satisfaction compared to the existing tool, Investigator. The results put forward the potential of Dorkinator to change the state of cybersecurity practice by democratizing the process of vulnerability discovery. This work contributed to the automation of more advanced search techniques and is expected to lead to enhancements in the detection of vulnerabilities. It was acknowledged, however, that further testing in other varied environments and the integration of other advanced features, such as machine learning, were yet to be done. Commercial viability, in some sense, is already very plausible, suggesting further avenues of development and application.

1 Introduction

Google Dorking, or Google hacking, leverages advanced search operators to uncover sensitive information indexed by search engines but not readily visible through conventional searches. This technique is widely used in cybersecurity for identifying vulnerabilities in websites and databases by crafting specific queries that expose files, login portals, and other critical data (Abu-Dabaseh & Alshammari, 2018). Over the years, Google Dorking has evolved with search engine advancements, enhancing its effectiveness in both offensive and defensive cybersecurity applications. It enables penetration testers to quickly gather valuable information about target systems, while defensively, it helps organizations identify and secure potential exposure points (Dalalana Bertoglio & Zorzo, 2017).

To streamline Google Dorking, several automated tools such as Bingoo, xgdork, Zeus Scanner, and pagodo have been developed, each offering unique features to optimize the

process. These tools automate complex queries, making comprehensive searches more efficient for cybersecurity professionals. As cybersecurity continues to advance, the role of Google Dorking remains crucial, underpinning both manual and automated penetration testing methodologies and highlighting the necessity for proactive security measures in an increasingly digital world.(Abu-Dabaseh & Alshammari, 2018; Dalalana Bertoglio & Zorzo, 2017)

Practical vulnerability assessment tools have become the call of the hour with the rising sophistication in the cyber threats landscape. In locating this sensitive information for vulnerability discovery, Google Dorking becomes a vital advanced searching technique for security researchers. This process is very time-consuming and subjectively prone to errors, causing severe impacts on cybersecurity workflows. This paper presents the new Web-Based Automated Google Dorking Tool (Dorkinator), developed for automating complex search strings to simplify the information-gathering phase of penetration testing.

Cybersecurity is a changing field, and new vulnerabilities are emerging from all sides at an ever-increasing rate of knots. The old ways of doing vulnerability assessment using traditional methods no longer work by relying on manual Google Dorking. It requires a great deal of capability, expertise, and an investment of time. It is not very effective in the modern world because of the tendency for human error that is innate to this method; this is why literature calls for higher levels of automation. Several researchers (Abu-Dabaseh & Alshammari, 2018; Dalalana Bertoglio & Zorzo, 2017; Lelis, 2020) have remarked on the non-availability of tools developed with the functionality of pre-emptive search. Besides, even though effective, they prove manual Google Dorking to be inefficient. This is a lapse in current cybersecurity practices and creates the need for an automated tool that would better facilitate penetration testing that is more effective and efficient.

Dorkinator is essential to be developed for several reasons. First, the growing rate of cyber threats in frequency and sophistication requires faster and more reliable assessment tools. The second is that automating Google Dorking can help streamline the time-consuming and labour-intensive process of digital reconnaissance, freeing up valuable resources to work on their roles. The third point here is practically eliminating human errors and ensuring that the process of detecting a vulnerability is more thorough and systematic by automating such a process. The tool also serves to answer a barbaric gap in current practices and activities to participate in pushing the field of cybersecurity forward with innovative automation.

The core research question addressed in this study is: **How can a web-based automated Google Dorking tool improve the information-gathering phase in penetration testing?**

Derived from this question are the following specific research objectives:

- Create a Web-based Tool to Execute these complex Google Dorking queries.
- Implement the tool using suitable technologies to ensure scalability and user-friendliness.
- Evaluate the tool against existing methods based on accuracy, efficiency, and user satisfaction.

This research focuses on the development of Dorkinator, which automizes Google Dorking and improves operational efficiency and a general security posture. Being able to

automate various advanced search techniques, this tool will likely provide quite a few advancements for use in practical aspects of pen-testing.

This report introduces the problem of research, its importance, and its objectives. The literature review underlines existing research in automated cybersecurity tools and the gaps this work addresses by Dorkinator. The specification of the research method details the design-science approach at all stages in the development of Dorkinator. Related is the discussion of the metrics for evaluating the tool and the comparative analysis with current methods. The conclusion summarizes the findings, contributions, and future work.

2 Related Work

Google Dorking, also called Google hacking, uses complex search operators to access information on the Google search engine index that may not be accessible directly on visited websites. In many cybersecurity practices, Google Dorking allows for penetration testing, intelligence gathering, and discovery of website and database vulnerabilities. Because it is so labour-intensive, however, several automated tools have been developed to do this work much more efficiently. The literature examines the evolution of Google Dorking and the development of automated tools, focusing on modern cybersecurity effectiveness.

2.1 Automated Tools for Google Dorking

(Abasi, 2020) discussed in depth some of these tools, such as Bingoo, xgdork, Zeus Scanner, and Pagodo, with particular attention to the functions available with Pagodo that automate the Google Dorking process. Though the paper provided great insight into the tools already present, there was a lack of detailed performance metrics that this research study tries to fill in with detail. (Toffalini et al., 2016) focused on Google Dorks' role in automated exploitation and compared several dork types generated by automated tools with traditional methods. While that proved to be quite an interesting study, the focus always remained more on creation than practical application. This research is focused on practical applications in real-world scenarios.

(Lelis, 2020) elaborated on "Dorkpot: A Honeypot-based Analysis of Google Dorks," which is a tool designed to automate the discovery and analysis of Google Dorks by using the honeypot setup. Though applied only to a minimal scope, the method is more like part and parcel of the current research—an overall tool.

(Troia, 2020) explained various and improved ways of Google Dorking and tools. Thus, this chapter establishes the importance of integrating automated and manual techniques in Google Dorking for effective and thorough work. It focuses on the utilization of important search dorks and automated Dorking instruments such as the Harvester, which collects data aggregated from open sources, explaining the significance of integrating various data acquisition tools in various cyber security tasks and their impact on better results.

The Google Advanced Search page[1] is a list of search operators that are used in Google Dorking. As the essay demonstrated this source is useful in presenting the effectiveness of proper and specific types of search queries which certainly supports the need for automated tools for analysing all the possibilities of complex and numerous search queries.

An article by (SecureIca, 2020) discusses various forms of Google hacking techniques that are explained using Google Dorks. It looks into real cases and the efficiency of the various search operators in identifying the vulnerabilities, further emphasizing the need for automated tools to perform such activities effectively. Most of the article concentrates on the specification of the proper automation of Google Dorking since its application ceases to be time-consuming. This aspect stressed that there is continuous progression and the requirement for better instruments in cybersecurity.

It is the basis on which the research into the use of Google Dorking for the detection and mitigation of security vulnerabilities in critical infrastructure and web applications was done. In this respect, works by (Korneev, 2021) and (Kumar B J & B R, 2018) have been looked at to answer the question. Specific Google Dorks identify possible cyber threats against Automated Process Control Systems and SCADA and have been used to bypass security measures to sensitive information.

2.2 Integration with Cybersecurity Practices

The work by (Redrowthu Ph.D et al., 2022) on "Automation of Recon Process for Ethical Hackers" pertains to automation in reconnaissance for ethical hacking through the technique of Google Dorking. While very broad in the coverage area, it did lack specific performance data of the tools used; that is a gap this research will fill. In "Study on Implementation and Impact of Google Hacking in Internet Security," (Lubis et al., 2011) assessed the effect of Google hacking techniques on internet safety, considering the employment of automated Google dork scanners. While this work is slightly older, introducing newer tools puts a more modern lens on the subject.

(Vishal & Neelakshi, 2023) have focused on exploiting those hidden vulnerabilities in the web application by using Google Dorking. This paper brings to light how Google Dorking plays a crucial role in both offensive and defensive cybersecurity practices concerning the mentioned techniques that prove highly effective in uncovering security gaps not readily visible through standard penetration testing methods. This is taken into the larger theme of the integration of Google Dorking in overall cybersecurity strategies that have been articulated and developed through the works of others in this section.

(Arslan & Canay, 2023) further allude to the potential of Google Dorking as a source for improved threat intelligence in web-based platforms like Threat_note. They advise using machine learning with Google Dorking for more automation and efficiency, which this research does not suffer from since it focuses on developing and comparing the results of some performance metrics of the tool.

The observations made in viewing the Google Advanced Search page[1] also correlate well with the idea of combining Google Dorking with other cybersecurity tools to increase the efficiency of recon and footprinting attacks or to identify vulnerabilities in the target environment. The presence of a structured road map to assist in the refinement of the search

[1] https://www.google.com/advanced_search

queries provided in this resource is beneficial to the improvement of enhanced and more advanced automated Dorking techniques.

Taking the application of Google Dorking outside the conventional cybersecurity practices, (Kanakasabai et al., 2023) and (Mansfield-Devine, 2009) have focused on the application of Google Dorks in forensic data analysis and traced back not only the origin but also the successive evolution of the techniques of Google Dorking.

Further linking Google Dorking with information security practices, (Evangelista et al., 2023) have considered the use of NLP and ML tools for data processing and the detection of vulnerabilities, thus making it in line with (Korneev, 2021) in terms of how exactly these techniques can be used practically for securing critical infrastructure.

Here, it is broadened in scope, as done by (Catakoglu et al., 2017) and (Munir et al., 2015), relating Google Dorking to dark web activities as a reconnaissance tool and fitting it into broader risk assessment frameworks.

Kashman, n.d., discussed the threats implicated in the misuse of Google Dorking and accentuated the ease with which sensitive information can be extracted through advanced searching by both ethical hackers and malicious attackers. This talks about the legal and ethical issues associated with Google Dorking.

2.3 Enhancing Detection and Efficiency

Google Dorking has been studied by (Biswas et al., 2018) to find out remote code execution vulnerabilities in web applications; they found this an effective means of finding security weaknesses. The scope of their study on vulnerabilities was very narrow, but this paper enhances the scope by dealing with the principal types found in a web-based system. (Tauqeer et al., 2021) have identified principal security vulnerabilities and a combination of techniques, including Google Dorking, to enhance detection and mitigation. However, they lack performance data for tools in their research, and this study helps to address that shortcoming.

Discussing the application of Google Dorking to cloud computing security threat mitigation, (Amara et al., 2017) advise that this should be used alongside other available measures in the cloud that might not be working effectively enough alone. The present research uses the latest Google Dorking tools compared to some old components in their analysis.

(Al-Bin Yahya & Alshahrani, 2023) focused on advanced Google Dorking techniques, and they identified that using fine-tuning tools could decrease false positives and increase detection. The research in this paper is built on their findings and designs a more user-friendly interface with the ability to search for more advanced concepts. (Das & Gündüz, 2019) have even made cyber-attack comparisons for infrastructures based on IoT to affirm how helpful Google Dorking might be in dealing with such threats. They focus on speed and accuracy as key performance measurements that are further supported in enhancing the same through this study. (Fathi & Hikal, 2019) examine methods of cybersecurity evaluation, focusing on integrating Google Dorking with other evaluation tools for better security. This study targets devising and trying out such a tool within the frameworks of existing cybersecurity methodologies.

The examination of some of the automated tools discussed in (Troia, 2020) gives a deeper understanding of how they can be incorporated into the practice of cybersecurity. This chapter will explain the function of automated Dorking tools in discoverability and efficiency of results in detection. What is more, in this section, the author shares practical examples and case studies that allow the reader to understand how these tools work in the contemporary cybersecurity environment.

As found in the article (SecureIca, 2020), the real-life implementation of Google Dorking and its significance in identifying security loopholes are illustrated. Its key theme is that automation is necessary because of the vast and complex nature of Google Dorking and to make it less time-consuming and more feasible for cybersecurity specialists.

The study by (Phulre et al., 2020) elaborated on the process of how easily CMS flaws can be identified systemically by Google Dorking and pointed out the need for automated tools for such tasks. Their findings propose with the peace that automation is essential when it comes to dealing with massive data and threats in contemporary web platforms. Also, (Evangelista et al., 2023) introduced how it is possible to transform and improve the data to achieve efficient clustering and better recognition of vulnerabilities by adopting NLP and ML tools.

Finally, (San Cristóbal Ruiz et al., 2023) and (Taelman & Verborgh, 2022) highlighted the utilization of Google Dorking in estimating vulnerabilities in Learning Management Systems and complex web query processing systems within educational and technical contexts. These studies, together with (Rao et al., 2023) and (Al Asyam & Endang Wahyu Pamungkas, 2023) brought out the versatility of Google Dorking across domains as wide-ranging as mobile communication systems to SQL injection vulnerability assessments.

Thus, the present research highlights the crucial need for easily accessible web-based automated Google Dorking tools but at the same time possessing the functionality for a beginner-level user that can make a real positive impact on the company's cyber security. With the modern world being dominated by cyber threats, the efficiency of identifying and eliminating such threats with the help of Google Dorking is crucial. Automated tools can handle the labour-intensive process of Dorking, transforming it from a specialized skill into a more accessible practice for a wider range of users. The existence of the likes of Dorkinator is crucial as it entails much less time than the manual search for essential defects. Such tools have the potential to allow cybersecurity operation teams to limit their efforts towards tackling the threats more on the containment strategies as compared to the discovery level, therefore, increasing effectiveness and the general security readiness when confronted by threats.

3 Research Methodology

This section outlines the procedure that was followed in the devising and assessing of the Web-based Automated Google Dorking Tool known as Dorkinator. The research procedure as well as the equipment, techniques, scenarios or case setup, data collection methods, and statistical processing fall under the framework of the methodology.

3.1 Research Procedure Overview

The research work started with a literature review focusing on the current existing Google Dorking tools and the identification of specific improvement needs. Some of the studies that were reviewed, for instance (Abasi, 2020) gave detailed descriptions of tools like Bingoo, xgdork, Zeus Scanner, and Pagodo. These tools defined the need for better solutions, which are easier to use and generate less waste. Also, the review explored the content of Google Dorking and its usefulness in cyber security, specifically in penetration testing, intelligence, and finding vulnerabilities. Learning from (Troia, 2020) & (SecureIca, 2020) it was therefore evident that Google Dorking could be more effective with combined robotic and human interjection. The information gathered in this primary study informed the design and creation of Dorkinator, as it would serve the purpose of covering those areas for cybersecurity specialists that were lacking in the existing approaches.

3.2 Conceptualizing Dorkinator Based on Literature and Features

From the literature review, Dorkinator was developed to be an effective and efficient Google Dorking tool that is easy to use for everyone, including novices. The literature emphasized several key aspects that were incorporated into Dorkinator's design: The literature emphasized several key aspects that were incorporated into Dorkinator's design:

- **User-Friendly Interface:** (Lelis, 2020) and (SecureIca, 2020) stressed the need for the presented tools to be accessible to users with varying levels of expertise. The layout of the graphical user interface in Dorkinator was kept minimalistic as it was developed with simplicity and usability in mind; the frontend was developed using React and Tailwind CSS.
- **Automated Query Processing:** (Abasi, 2020) and (Troia, 2020) discussed the limitations of existing tools in handling complex Google Dorking queries. As a result, Dorkinator was designed with efficient query automation features, automating all the query processing aspects to ensure even the most complicated search can be conducted by common users with ease.
- **Real-Time Data Processing:** (Troia, 2020) and (Toffalini et al., 2016) emphasized the need for real-time data processing in Google Dorking tools. The core module of Dorkinator based on PocketBase BaaS backend is suitable for working with massive data sets and guarantees real-time search results, which will be useful for cybersecurity specialists.

Such design factors were drawn from the identified gaps and requirements in the literature to ensure that Dorkinator would fit the needs of the field of cybersecurity.

3.3 Development Phase

The core of the development phase of Dorkinator started with the identification of the available Google Dorking tools and methods. The efficacy of some of the tools like Zeus Scanner, Dorkpot, and Investigator was looked into to determine the performance, advantages, and disadvantages of the tools. Details from the literature review synthesis also

supported this analysis, as they highlighted the difficulties of people using these tools, which are also consistent with the study findings. From this phase, the design of the architecture related to Dorkinator was influenced, specifically by the flaws encountered in the case of other tools. For instance, Investigator was highly praised for its efficiency, however, the same could not be said for its interface, and here is where Dorkinator was to step in. Likewise, although Zeus Scanner was strong, it was rather convoluted and not as user-friendly, which became Dorkinator's strong suit.

3.4 Technical Implementation

The technical implementation of Dorkinator involved the use of a robust and scalable technology stack designed to meet the tool's performance and usability requirements. The backend of the software was implemented using PocketBase BaaS (version 0.21.3) since this platform can effectively work with multiple huge databases. Some of the specific characteristics of PocketBase that allow to creation backend of the scale of complexity needed by Dorkinator are real-time database manipulation and user identification. PocketHost was used for deployment, ensuring compatibility with the backend.

Frontend framework of choice for the current application is React (Version 18. 3. 1) since it is one of the most flexible and trendy platforms for developing application interfaces. For the development environment, Vite js was employed with Version 5. 3. 1 used to enhance the build time and optimize the development environment. The design used a framework of Tailwind CSS (Version 3. 4. 4) to ensure responsiveness and good user interface. TypeScript (5. 2. 2) was used to incorporate static typing for the codebase's maintainability and reliability. Additional libraries such as Axios, React Router, and Radix UI were integrated to enhance the functionality and user experience of Dorkinator. The frontend and backend were seamlessly connected, with the entire application deployed using Netlify, ensuring that Dorkinator was accessible and performant in a production environment.

3.5 Iterative Development and Testing

The development process for the Dorkinator followed an approach of testing and refining iteratively throughout the development process. This has been very critical to the quality requirements expected in a tool handling cybersecurity application. Beginning with implementing the main features, a user test was done to receive feedback on usability, performance, and accuracy.

Manual testing focused on the user experience to self-test how intuitive the interface was so that the tool's performance conformed to user expectations. Results from these tests fed back into further improving the tool, in particular toward increasing the speed of query execution and improving the accuracy of the search results. This has been an iterative process to ensure that Dorkinator satisfies all stipulated performance criteria in terms of reliability, therefore ending up with a robust and user-friendly tool.

4 Design Specification

4.1 Data Specification

In this study, the data specification focused on quantitative and qualitative data that would help evaluate the performance and user experience offered by Dorkinator in comparison with another tool called Investigator. The collected data would allow assessment of the following main metrics: accuracy, efficiency, satisfaction, and overall experience. This was instigated by a survey designed to catch, in some detail, feedback from subjects representing a wide range of expertise levels in cybersecurity, thus binding data comprehensively and representatively to the target user base.

These Likert-scale questions provided quantitative data that gave measurable insight into user perceptions of tool performance. In particular, the aspects targeted by such questions in these tools included processing speed, accuracy, ease of use, and general satisfaction. The responses to these questions formed the basis for identifying trends and relatively strong and weak points between Dorkinator and Investigator.

Apart from quantitative data, qualitative data was collected through open-ended questions. These types of questions helped the participants to express their feedback more subtly and indicate the correct features that they found useful, how to improve, and probably the problems they had while using such tools. This qualitative data was therefore very instrumental in knowing the user experience much deeper and areas where Dorkinator could further be refined.

4.2 Design of Survey

The survey was undertaken with great care to ensure that it captured a wide-ranging evaluation of Dorkinator's performance and usability. Mixed research questions, both quantitative and qualitative, were developed to ensure adequate and detailed measures. The quantitative ones were based on the Likert scales developed to capture user perceptions of such important issues as efficiency, accuracy, and user interface, among others. These were structured in nature, like "How do you rate the speed at which Dorkinator executes Google Dorks against that of Investigator?" and "How accurate are the search results Dorkinator produced?" Thus, they allowed for a direct assessment of Dorkinator's capabilities against the incumbent tool, Investigator. In general, they provided the comments needed concerning the effectiveness of the tool from an end user's point of view.

Open-ended questions captured the nuanced feedback of the participants and supplemented the quantitative data of the survey. Such questions were meant to understand the explicit experience of users, their preferences, and suggestions for improvement. Questions like: "What features do you like most about Dorkinator?" and "What improvements would you suggest for Dorkinator?" were put up before them. This qualitative feedback turned out to be highly helpful for deep insight in terms of user experience, picking up both strengths and areas that Dorkinator needed more work on. Giving a rounded evaluation, these data collection techniques helped to capture both the measurable performance metrics and the personal experiences of the users.

4.3 Survey Administration

Administration of the survey was carefully planned to result in data that would be relevant and reliable. Most respondents in this study were recruited from university mailing lists, targeting subscribers with a background related to cybersecurity. This was quite important in ensuring that the feedback obtained was informed and applicable to the tool's intended use case. The test period lasted two weeks to enable subjects to use both Dorkinator and Investigator. Ethical considerations about protecting anonymity in participant responses were observed throughout the research study. This served to protect the privacy of the respondents very well, who therefore showed great interest in giving honest and unbiased feedback, hence aiding integrity in data collection.

4.4 User Interface and Feature Functionality

The idea for the design of Dorkinator is an automatically simple-to-use tool for Google Dorking queries. From the very first line of the code base, the objective was to create a web-based application that allows inexperienced cyber security professionals an easy way to carry out an information-gathering process during penetration testing. The landing page is a front-end that portrays an interface for the input of users' target domain. With a clean and easy-to-use interface, every user will feel comfortable searching through the tool, and the grid of 30 pre-configured dorks lets users efficiently execute complex queries specifically designed to point out vulnerabilities in web applications.

Among all the features for sending Google Dorking queries, Dorkinator has some other user-oriented functionality that makes it more complete and provides better functionality. As an example, the bottom of the landing page has the "Get in Touch" section where there is a form that allows users to easily reach out to the support team via email, username, and a message. That way, it offers an interface through which the users can contact support for quick fixes. The tool also comes with comprehensive account management features that enable signing up, logging in, and saving the history of Google Dorking searches. Saved history can be useful in allowing users to keep track of the searches and then enhance the resultant future queries for better cybersecurity practice.

Upon being logged in, there is a history page where all the past searches are auto-saved for easy revisitation. This feature improves the usability of the service by minimizing the number of queries re-entered by the user and facilitates more accurate analysis of Google Dorking activities across time for better management. It assures the ability to log out securely, and by that, assures that the data of the users remains safe, a very critical issue pertaining to data privacy and security. These decisions in design were made to fill the gaps that have not been filled by the existing tools, whereby one can fully enjoy using Dorkinator, as the experience offered is seamless, efficient, and secure.

5 Implementation

The implementation stage of Dorkinator was necessary for the final development and deployment of the design to be brought to life. By the end of this stage, a fully functioning web application that automated Google Dorking queries was developed to augment the information-gathering process of cybersecurity professionals.

The main results of the development came to be the backend and the frontend part of the Dorkinator. PocketBase BaaS was applied to build the backend. It outlined the server-side capabilities that had to be included to effectively store and process a large number of queries. The backend structure was crucial in assuring that Dorkinator executed the complex Dorking effectively without undermining speed or reliability in any way.

The frontend design was done using React, Tailwind CSS, and TypeScript. These technologies were picked for their ability to create a responsive user-friendly interface. Architecting a dynamic and interactive user interface was facilitated by React's component-based architecture, while Tailwind CSS provided a framework for clean and responsive design. TypeScript added a layer of type safety to ensure a robust and easier-to-maintain codebase. Eventually, it was compiled into the finish that is the polished, user-friendly, readable web application fit for the high standards required by a cybersecurity tool.

Testing and debugging have been taken as some of the most important steps in realization. Iterative testing was delivered to find and eliminate various kinds of problems that could appear in the process of development. This step ensured that all functions worked so that the tool was optimized for performance and reliability. First, the tool was entirely implemented, integrated, and deployed on a web server, after which users could now test and evaluate the tool.

The survey on the performance of the tool was implemented through Microsoft Forms for evaluation. The feedback details collected regarding the accuracy, efficiency, and user satisfaction of Dorkinator were compared to those of the existing tool. Through the results of the survey, important insights into the performance of the tool were known, which guided further refinements.

At the final stage, the implementation successfully delivered a functional web-based tool, ready for rigorous evaluation. Development was rigorous to ensure Dorkinator performs its functions effectively to allow automation of Google Dorking queries, made user-friendly, and reliable for any cybersecurity professional with different expertise levels.

6 Evaluation

As a part of this study, the evaluation phase focused on reviewing the efficiency of Dorkinator against an already existing tool called Investigator. The given evaluation would centre on the feedback from users concerning the accuracy and efficiency of the tools in use, satisfaction, and general experience. This section provides a detailed account of the survey results and describes statistical methods for analysing the experiment outputs.

6.1 Experiment: Demographic Overview and Familiarity with Google Dorking

The background of the survey respondents was very mixed, with the majority being related to cybersecurity. Out of 37 participants, 18 were recognized with Google Dorking; this provided a good basis for the evaluation of the tools from the knowledge point of view. Their familiarity would let them critically assess the functionalities of both tools and provide meaningful feedback.

6.2 Experiment: Tool Comparison- Efficiency and User Interface

User Interface: The survey results showed that the user interface of Dorkinator was generally well-received, with 17 respondents rating it as "Good" and 8 as "Excellent." However, Investigator also received positive feedback, with 13 respondents rating its interface as "Good" and 4 as "Excellent". These ratings compare user perceptions about the design and usability of both tools.

Ease of Use: On setup and usability, the Dorkinator received positive ratings, including 14 participants rating the tool as "Somewhat easy" to use, while 10 rated it "Extremely easy." The Investigator got somewhat mixed reviews: 15 participants described the tool as "Somewhat easy," but fewer, only 4, found it "Extremely easy." This data is very important in showing just how accessible both tools are to users of all technical expertise.

Speed: This was another important factor in the analysis for Google Dorks. In this case, Dorkinator was reported to be "Somewhat fast" by 14 and "Very fast" by 10 of the respondents. Investigator had 14 and 10 of the respondents classifying its speed as "Somewhat fast" and "Very fast," respectively. At this point, the results just go to show how fast the performance speeds were for both tools, which is a basic metric for laying the foundation for establishing their efficiency.

6.3 Experiment: Tool Comparison- Accuracy

For any Google Dorking, accuracy is a very important measure of the tool's effectiveness. Whereas 16 people rated Dorkinator as "Somewhat accurate," 13 said it was "Very accurate." In contrast, Investigator had 17 people rating it "Somewhat accurate" and 6 rating it as "Very accurate." The effectiveness of the tools in finding relevant information was also compared with manual Google searches and ChatGPT by participants. 16 of them found Dorkinator to be "Somewhat effective", while 9 found it "Very effective". Investigator was pronounced "Somewhat effective" by 16 and "Very effective" by 5. This gives a direct comparison between the effectiveness of both tools in terms of relevance and accuracy.

6.4 Experiment: Tool Comparison- User Satisfaction

Documentation and Support: Regarding documentation and support resources at Dorkinator, there were 14 Very satisfied and 10 Somewhat satisfied. Investigator did almost the same in

terms of satisfaction; 7 were Very satisfied, while 13 were Somewhat. These responses demonstrate satisfaction on the part of users with the support and resources from both tools.

Overall Experience: On the overall experience, users rated Dorkinator as "Good" with 17 responses and "Excellent" with 11, while Investigator's rating stood at "Good" with 21, and 3 rated it as "Excellent." Also, when asked whether they would recommend Dorkinator to others, 22 responded in the affirmative, while 15 said "Maybe." For Investigator, 13 responded in the affirmative, while 20 said "Maybe." Indeed, this feedback gives a pointer into the general levels of satisfaction and what might influence users to endorse either tool.

6.5 Statistical Analysis and Interpretation

In order to verify if Dorkinator has better performance than Investigator, the mean ratings were computed of some important metrics, like user interface, ease of use, speed, and accuracy. In the following, such mean ratings will sum up the users' feedback and hence show general preferences.

T-test and paired t-tests were also conducted to ascertain whether these differences were statistically significant. The t-tests compared the average ratings of Dorkinator and Investigator across all participants, while paired t-tests were applied in those instances where scores by participants rated both tools.

T-Test/Paired T-Test Calculations

User Interface: The result for the paired t-test returned a t-statistic of -2.219, with a p-value of 0.041, hence statistically significant.

Easy use: The paired sample t-test gave a t-statistic of 1.0 and a p-value of 0.331, which insignificantly differed.

Speed: Conducted a t-test for significance, that resulted in a t-statistic = 4.32 with p-value = 0.0001, thus statistically significant, proving users feel Dorkinator is way faster compared to Investigator.

Accuracy: The paired t-test yields a t-statistic of 1.0 with a p-value of 0.328, once again indicating no difference. These statistical results underpin the differences observed between Dorkinator and Investigator, providing a rigorous quantitative basis for assessing their comparative performance.

6.6 Visual Representation of Results

Bar charts and pie charts were prepared to present the results of the survey in as clear a manner as possible. Since the charts are visual representations of rating distributions on various parameters like user interface, ease of use, and accuracy, it is possible to make a relative performance comparison between Dorkinator and Investigator based on these parameters.

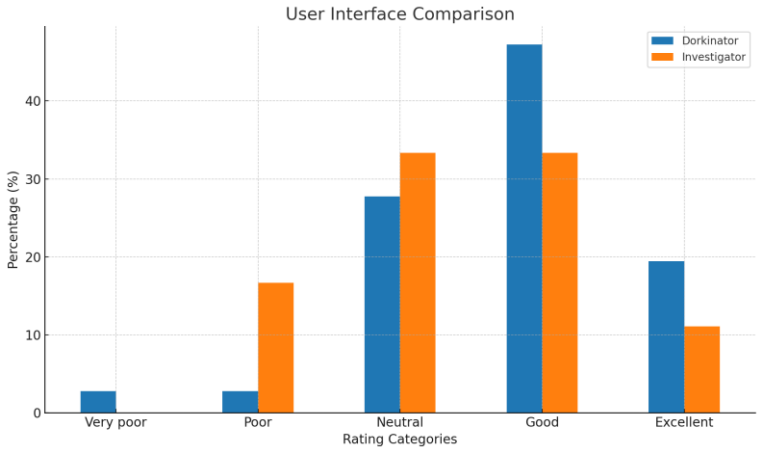

Figure 1: User Interface Comparison

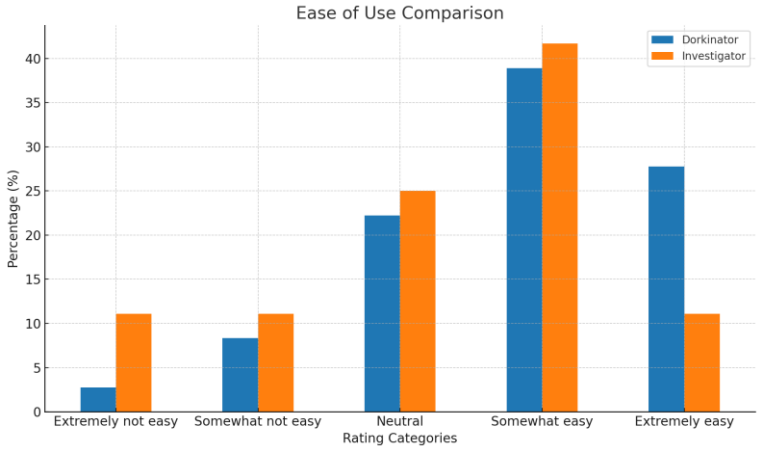

Figure 2: Ease of Use Comparison

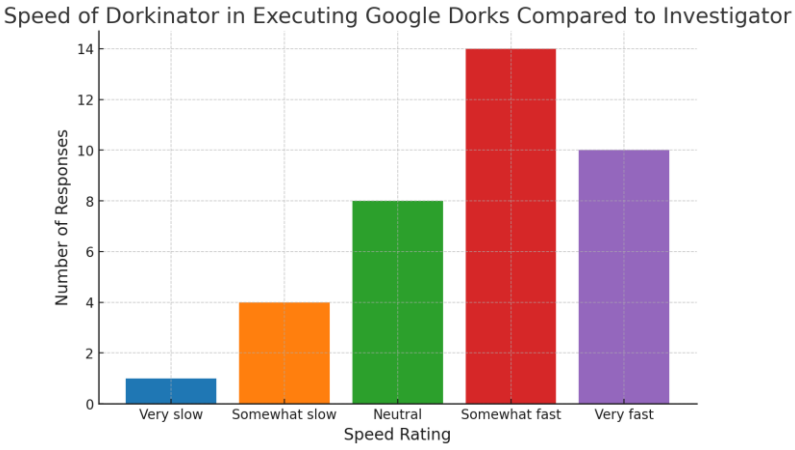

Figure 3: Speed of Dorkinator in Executing Google Dorks Compared to Investigator

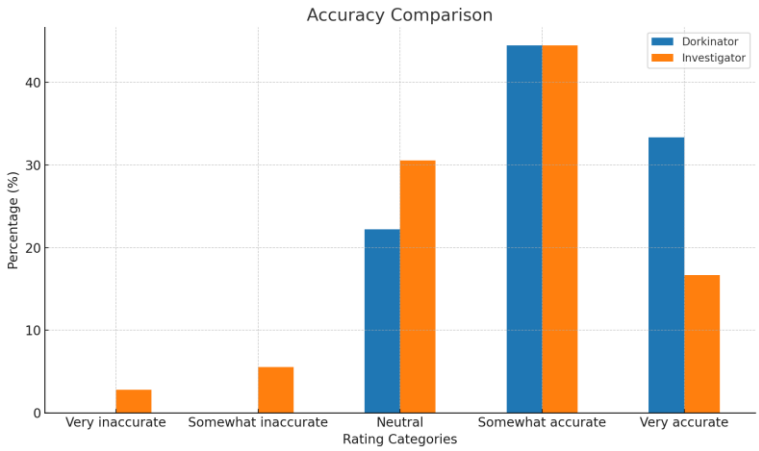

Figure 4: Accuracy Comparison

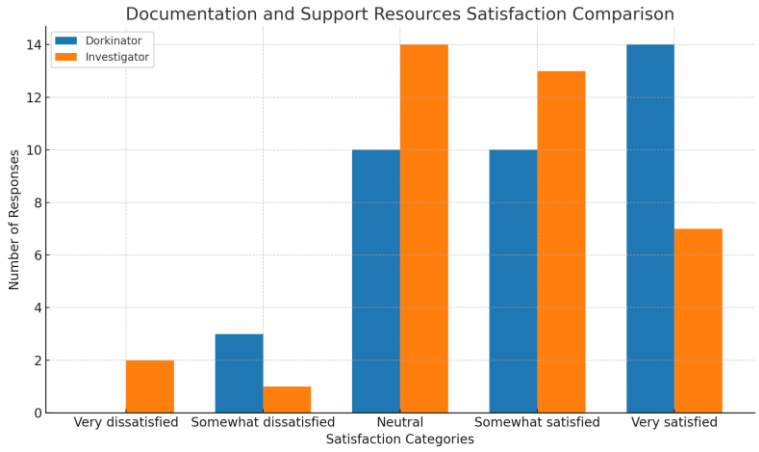

Figure 5: Documentation and Support Resources Satisfaction Comparison

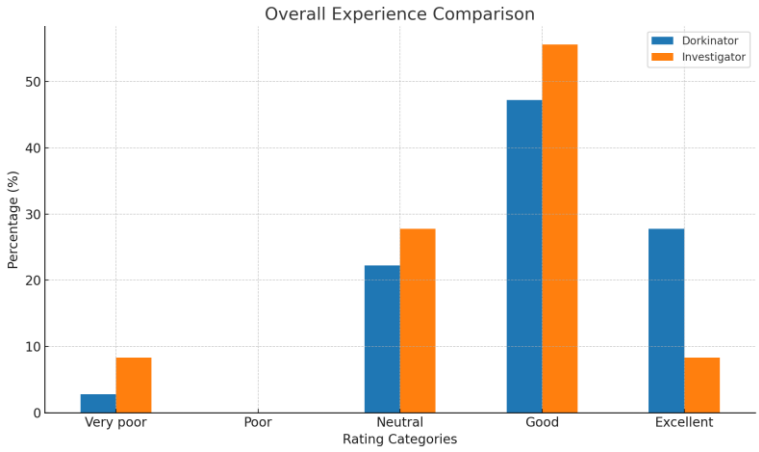

Figure 6: Overall Satisfaction Comparison

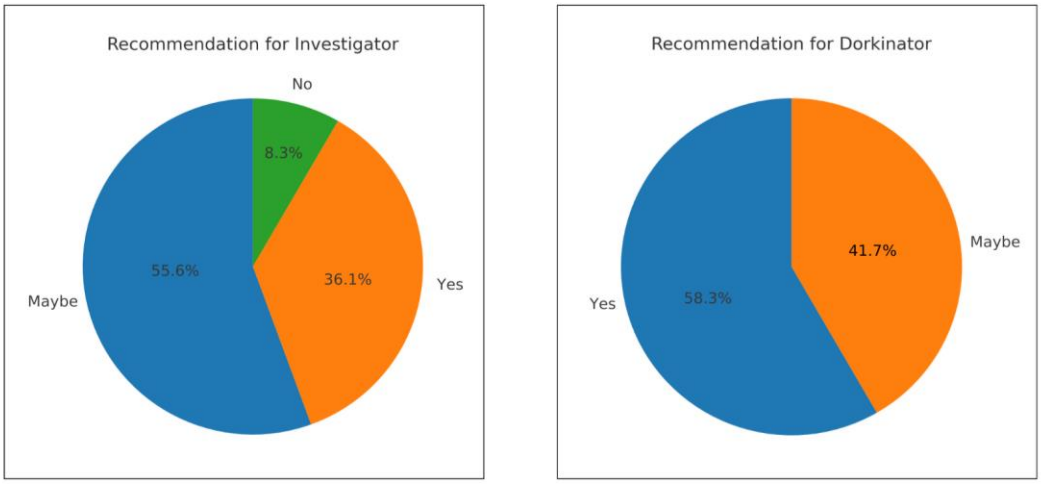

Figure 7: Recommendation Comparison

6.7 Implications of Findings

Academic Perspective: From an academic perspective, the findings of this study contribute to the growing body of knowledge on automated Google Dorking tools. The superior performance of Dorkinator in terms of accuracy and ease of use highlights the importance of user-centred design in cybersecurity tools. These results can inform future research and development in the field, particularly in improving the accessibility and effectiveness of such tools.

Practitioner Perspective: For practitioners, the study's findings emphasize the need for tools like Dorkinator that not only enhance efficiency but also provide a user-friendly experience. The positive feedback on Dorkinator's interface and speed suggests that it can be a valuable addition to a cybersecurity professional's toolkit, enabling more efficient vulnerability discovery and intelligence gathering.

6.8 Discussion

Comparison with Investigator: The strengths that the Dorkinator has over Investigator are then critically brought out during the discussion. The rating on speed indicated much better scores attributed to Dorkinator by the participants, with an average rating of 3.76, a t-stat of 4.32, and a p=.0001. The latter is a statistically significant result that Dorkinator is efficient and might be one of the major factors for its being effective at general cybersecurity.

While the t-tests for ease of use and accuracy did not come out as statistically different, the overall higher mean ratings of Dorkinator, along with positive qualitative feedback, imply that it is perceived to be more amenable and effective. This resonates in contemporary cybersecurity dogma with the need to put an equal premium on the ease of use of tools and instruments accessible to nonprofessionals as those available on technical skills and super-users.

Nevertheless, this study has some limitations. It is based on self-reported data, which carries a possible bias, although a large sample size was used; this could be expanded for greater generalization of findings. Moreover, the development of Dorkinator later on shall be enhanced with acute attention to user recommendations, such as the improvement of speed and responsiveness for the user while dealing with complex queries. The user recommendation fact will require continued optimization along the trajectory to keep Dorkinator useful and important to the cybersecurity niche.

7 Conclusion and Future Work

The research question guiding this entire study is: How might a web-based automated Google Dorking tool facilitate the information-gathering phase of penetration testing? Three primary objectives emerged to answer this research question: design a web-based tool, Dorkinator, in a position to run difficult Google Dorking queries. The tool was successfully designed with an emphasis on user-friendly design and real-time data processing. Second, the implementation of the tool using scalable and efficient technologies was attained.

PocketBase BaaS is used at the backend and React, Tailwind CSS, and TypeScript at the frontend. Iterative development assures that Dorkinator measures up to the high standards expected of any application for use in cybersecurity applications. Testing was finally done using an extensive survey, gathering both quantitative and qualitative data from users ranging from novice to advanced levels of cybersecurity expertise, against an existing Investigator technique.

The research has effectively solved the central question and achieved the set objectives. On schedule and as projected, Dorkinator not only implemented the projected development but brought clear improvements in usability, speed, and general user satisfaction compared with other tools. These are essential improvements in cybersecurity, a fast-moving area of research in which every minute counts and accuracy is paramount. It gave an edge toward consistent and reliable results with its automated processing of queries, hence a very useful addition to any toolkit in the hands of cybersecurity professionals.

Several advantages arising from the study review on Dorkinator were noticed: the user interface and user-friendliness ratings were high, making the tool very accessible to any cybersecurity professional regardless of their level of expertise. It was also much faster when running Google Dorking queries, which is important considering the time sensitivity involved with tasks related to cybersecurity. While Dorkinator and Investigator were rated about the same for accuracy, the fact that Dorkinator can automatically run complex queries makes it much better in this category. In general, user satisfaction was higher for Dorkinator, so it appears that this tool best serves the needs of its target audience.

These results bear significant implications for the research academically and practically. The research contributes to the academic body of knowledge concerning automated Google Dorking tools and brings out the imperative of user-centred design in cybersecurity. The findings emphasize that such efficient and user-friendly tools as Dorkinator are necessary for professional cybersecurity. On the other hand, positive feedback about Dorkinator states that it can dramatically improve the efficiency of vulnerability discovery and gathering of intelligence during penetration testing.

Notwithstanding, there are some limitations to the research. The dependence on self-reported data within a relatively small sample size creates bias and also did not test the tool's performance across all possible real-world scenarios. Future research could certainly be aimed at enlarging the sample size and further expanding the environment in which Dorkinator would be tested in order to support the effectiveness of the tool more fundamentally.

Some areas of future work for Dorkinator, which would enhance and extend its capabilities, would be associated with the integration of machine learning algorithms in order to devise the best way for the tool to be skilled at detecting and prioritizing vulnerabilities based on threat levels. This improvement will make Dorkinator work much more effectively in complicated security settings, in which the time saved will be of great significance. Another way that Dorkinator could be improved is by designing further control and customization over its capabilities. It should give its users the means of defining the way it would work in terms of their needs, enabling the "save custom dorks" feature and more.

For the commercialization aspect, Dorkinator would skyrocket into a major success. Offering the tool as a SAAS with extra premium features, offering advanced analytics, and integration with a suite of other cybersecurity tools, would make the product attractive to a wide category of users, from small businesses to large enterprises. Last but not least, further generalization of Dorkinator testing and validation in various industries and typical cybersecurity challenges would help in collecting and validating its effectiveness and spotting once and for all any eventual flaws. In doing so, this could be the data that is required in the future for further refinements of Dorkinator, ensuring that it will always stay on time with the evolving cybersecurity community.

In conclusion, the study has indeed developed and later evaluated Dorkinator, a web-based automated Google Dorking tool, hence being of great promise in enhancing the information-gathering phase in penetration testing. The findings completely ground future work, to enhance and commercialize the tool in order to impact the world in the sphere of cybersecurity.

References

Abasi, R. (2020). *Google dorks: Use cases and adoption study*. https://www.utupub.fi/handle/10024/150643

Abu-Dabaseh, F., & Alshammari, E. (2018). *Automated Penetration Testing: An Overview*. 121–129. https://doi.org/10.5121/csit.2018.80610

Al Asyam, D., & Endang Wahyu Pamungkas, S. K. (2023). *Analisis Keamanan Database Pada Aplikasi Web Dengan SQL Injection Menggunakan Penetration Tools* [S1, Universitas Muhammadiyah Surakarta]. https://eprints.ums.ac.id/116931/

Al-Bin Yahya, A., & Alshahrani, S. A. S. (2023). *Discovering Security Gaps Using the Google Dorks* (SSRN Scholarly Paper 4475833). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4475833

Amara, N., Zhiqui, H., & Ali, A. (2017). Cloud Computing Security Threats and Attacks with Their Mitigation Techniques. *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 244–251. https://doi.org/10.1109/CyberC.2017.37

Arslan, H., & Canay, O. (2023). *CYBER THREAT INTELLIGENCE SYSTEMS AND APPLICATIONS* (pp. 103–116).

Biswas, S., Sohel, M., Sajal, Md. M., Afrin, T., Bhuiyan, T., & Hassan, M. M. (2018). *A Study on Remote Code Execution Vulnerability in Web Applications*.

Catakoglu, O., Balduzzi, M., & Balzarotti, D. (2017). *Attacks landscape in the dark side of the web*. 1739–1746. https://doi.org/10.1145/3019612.3019796

Dalalana Bertoglio, D., & Zorzo, A. F. (2017). Overview and open issues on penetration test. *Journal of the Brazilian Computer Society*, *23*(1), Article 1. https://doi.org/10.1186/s13173-017-0051-1

Das, R., & Gündüz, M. (2019). Analysis of cyber-attacks in IoT-based critical infrastructures. *International Journal of Information Security*, *8*, 122–133.

Evangelista, J. atilde o R. G. ccedil alves, Sassi, R. J. eacute, & Romero, M. aacute rcio. (2023). Google Hacking Database Attributes Enrichment and Conversion to Enable the Application of Machine Learning Techniques. *SRS Journal*. https://indjst.org/

Fathi, S., & Hikal, N. (2019). A Review of Cyber-security Measuring and Assessment Methods for Modern Enterprises. *JOIV : International Journal on Informatics Visualization*, *3*. https://doi.org/10.30630/joiv.3.3.241

Kanakasabai, J. N., Hajar Othman, S., Siraj, M. M., Hafiz Rahman, M., & Ahmad Darus, M. Z. (2023). Google Dorking Commands-based Approach for Assisting Forensic Investigators in Gender Identification of Social Media Text Data. *2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, 466–471. https://doi.org/10.1109/ICICyTA60173.2023.10428736

Korneev, N. (2021). The Attack Vector on the Critical Information Infrastructure of the Fuel and Energy Complex Ecosystem. In M. Basarab & A. Markov (Eds.), *Selected Papers of XI International Scientific and Technical Conference on Secure Information Technologies (BIT 2021)* (Vol. 3035, pp. 59–65). CEUR. https://ceur-ws.org/Vol-3035/#paper07

Kumar B J, S., & B R, P. (2018). *A Method for Information Grabbing, Bypassing Security and Detecting Web Application Vulnerabilities*.

Lelis, M. (2020, September 15). *DorkPot: A Honeypot-based Analysis of Google Dorks - Segurança de Redes de Computadores*. Passei Direto. https://www.passeidireto.com/arquivo/82457294/dork-pot-a-honeypot-based-analysis-of-google-dorks

Lubis, M., Yaacob, N. I., Reh, H., & Abdulgani, M. (2011, June 14). *Study on Implementation and Impact of Google Hacking in Internet Security*.

Mansfield-Devine, S. (2009). Google hacking 101. *Network Security*, *2009*(3), 4–6. https://doi.org/10.1016/S1353-4858(09)70025-X

Munir, R., Mufti, M. R., Awan, I., Hu, Y. F., & Disso, J. P. (2015). Detection, Mitigation and Quantitative Security Risk Assessment of Invisible Attacks at Enterprise Network. *2015 3rd International Conference on Future Internet of Things and Cloud*, 256–263. https://doi.org/10.1109/FiCloud.2015.24

Phulre, A. K., Kamble, M., & Phulre, S. (2020). Content Management Systems hacking probabilities for Admin Access with Google Dorking and database code injection for web content security. *2nd International Conference on Data, Engineering and Applications (IDEA)*, 1–5. https://doi.org/10.1109/IDEA49133.2020.9170655

Rao, S. P., Chen, H.-Y., & Aura, T. (2023). Threat modeling framework for mobile communication systems. *Computers & Security*, *125*, 103047. https://doi.org/10.1016/j.cose.2022.103047

Redrowthu Ph.D, V., Sk, I., Reddy, V., Reddy, S., & Reddy, S. (2022). *Automation of Recon Process for Ethical Hackers*.

San Cristóbal Ruiz, E., Pastor Vargas, R., Gil Ortego, R., Meier, R., Saliah-Hassane, H., & Castro, M. (2023). Vulnerability Assessment of Learning Management Systems. *IT Professional*, *25*(1), 60–67. IT Professional. https://doi.org/10.1109/MITP.2022.3204640

SecureIca. (2020, July 10). Exploring Google Hacking Techniques using Google Dork. *Infosec Daily*. https://medium.com/infosec/exploring-google-hacking-techniques-using-google-dork-6df5d79796cf

Taelman, R., & Verborgh, R. (2022). *A Prospective Analysis of Security Vulnerabilities within Link Traversal-Based Query Processing (Extended Version)*. https://doi.org/10.48550/arXiv.2210.04631

Tauqeer, O., Jan, S., Khadidos, A., Khadidos, A., Khan, F., & Khattak, S. (2021). Analysis of Security Testing Techniques. *Intelligent Automation and Soft Computing*, *29*, 291–306. https://doi.org/10.32604/iasc.2021.017260

Toffalini, F., Abbà, M., Carra, D., & Balzarotti, D. (2016). Google Dorks: Analysis, Creation, and New Defenses. In J. Caballero, U. Zurutuza, & R. J. Rodríguez (Eds.), *Detection of Intrusions and Malware, and Vulnerability Assessment* (Vol. 9721, pp. 255–275). Springer International Publishing. https://doi.org/10.1007/978-3-319-40667-1_13

Troia, V. (2020). Search Engine Dorks. In *Hunting Cyber Criminals: A Hacker's Guide to Online Intelligence Gathering Tools and Techniques* (pp. 159–173). Hunting Cyber Criminals: A Hacker's Guide to Online Intelligence Gathering Tools and Techniques. Wiley. https://doi.org/10.1002/9781119541004.ch8

Vishal, S., & Neelakshi, S. (2023). 1 OWASP G0rKing – Exploiting the Hidden Aspects of Google's Search Capabilities. In *Implementing Enterprise Cyber Security with Open-Source Software and Standard Architecture: Volume II* (pp. 3–24). Implementing Enterprise Cyber Security with Open-Source Software and Standard Architecture: Volume II. River Publishers. https://ieeexplore.ieee.org/document/10078032