Unmasking the True Identity: Unveiling the Secrets of Virtual Private Networks and Proxies

Jain, Vikas Kumar; Aggrawal, Jatin; Dangi, Ramraj; Shukla, Shiv Shankar Prasad; Yadav, Anil Kumar; Choudhary, Gaurav

Published in:
Information

Link to article, DOI:
10.3390/info16020126

Publication date:
2025

Document Version
Publisher's PDF, also known as Version of record

information

MDPI

Article

Unmasking the True Identity: Unveiling the Secrets of Virtual Private Networks and Proxies

Vikas Kumar Jain [1], Jatin Aggrawal [2], Ramraj Dangi [3], Shiv Shankar Prasad Shukla [2], Anil Kumar Yadav [2] and Gaurav Choudhary [4,*]

[1] School of Technology Management and Engineering, SVKM'S NMIMS, Indore 452005, India; vikas.jain@nmims.edu
[2] School of Computing Science Engineering and Artificial Intelligence, VIT Bhopal University, Sehore 466114, India; jatin.aggrawal2021@vitbhopal.ac.in (J.A.); shivshankar.prasad@vitbhopal.ac.in (S.S.P.S.); aky125@gmail.com (A.K.Y.)
[3] School of Computing Science and Engineering, VIT Bhopal University, Sehore 466114, India; ramrajdangi@vitbhopal.ac.in
[4] DTU Compute, Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), 2800 Kongens Lyngby, Denmark
* Correspondence: gauch@dtu.dk

Abstract: The growing use of VPNs, proxy servers, and Tor browsers has significantly enhanced online privacy and anonymity. However, these technologies are also exploited by cybercriminals to obscure their identities, posing serious cybersecurity threats. Existing detection methods face challenges in accurately tracing the real IP addresses hidden behind these anonymization tools. This study presents a novel approach to unmasking true identities by leveraging honeypots and Canarytokens to track concealed connections. By embedding deceptive tracking mechanisms within decoy systems, we successfully capture the real IP addresses of users attempting to evade detection. Our methodology was rigorously tested across various network environments and payload types, ensuring effectiveness in real-world scenarios. The findings demonstrate the practicality and scalability of using Canarytokens for IP unmasking, providing a non-intrusive, legally compliant solution to combat online anonymity misuse. This research contributes to strengthening cyber threat intelligence, offering actionable insights for law enforcement, cybersecurity professionals, and digital forensics. Future work will focus on enhancing detection accuracy and addressing the advanced evasion tactics used by sophisticated attackers.

Keywords: anonymous; network; honeypot; security; troublemakers; VPN

Academic Editor: Yousef Fazea

Received: 7 January 2025
Revised: 31 January 2025
Accepted: 6 February 2025
Published: 9 February 2025

Citation: Jain, V.K.; Aggrawal, J.; Dangi, R.; Prasad Shukla, S.S.; Yadav, A.K.; Choudhary, G. Unmasking the True Identity: Unveiling the Secrets of Virtual Private Networks and Proxies. *Information* **2025**, *16*, 126. https://doi.org/10.3390/info16020126

1. Introduction

As the internet becomes increasingly integrated into our daily lives, concerns around online privacy and safety have grown in significance. A common way for people to protect their online activity and remain anonymous is utilizing virtual private networks (VPNs) and proxies.

VPN: A virtual private network is a method that allows us to use public networks as private networks by employing encryption, authentication, and integrity protection. This enables users to access private networks remotely and simulates a private network over public networks [1].

Proxy Server: The purpose of a proxy server is to conceal the IP address of the client by using an anonymous network ID in place of the client's actual IP address. This prevents the client's real IP address from being disclosed.

However, use of these tools has also raised concerns about their potential misuse for illegal or immoral activities. This research aimed to shed light on the effectiveness of unmasking the real IP addresses concealed by VPNs and proxies. Additionally, it considers the ethical implications of revealing actual IP addresses, addressing privacy and anonymity concerns.

This study aimed to develop and introduce a method for identifying real IP addresses, to safeguard businesses. Our approach draws on insights from existing detection methods, the underlying technology, and experimental techniques. The proposed IP detection method aims to identify cybercriminals who access websites or web applications through proxies/VPNs and take preventive measures before fraudulent activities occur.

Moreover, the research takes into account the ethical consequences of revealing actual IP addresses. We strive to balance the need for security with the protection of individual privacy and anonymity.

Specifically, this paper explores innovative uses of honeypots, which are decoy computer systems designed to attract and study potential threats. By strategically placing these honeypots, we can observe attempts to hide real IP addresses using VPNs and proxies. We collected data on this behavior and employed advanced analysis methods to find patterns that could reveal the true origins of concealed IP addresses. This enhances the proposed IP detection method's ability to identify potential cybercriminals, while considering the ethical implications.

The findings of this research will contribute to ongoing efforts to strengthen online security and protect against cyber threats. Ultimately, our goal is to provide practical insights and recommendations for identifying and mitigating the risks associated with the misuse of VPNs and proxies, ensuring a safer and more secure online environment for all users. At the same time, we are committed to respecting privacy and maintaining anonymity, where appropriate.

The primary objectives of this work were as follows:

1. Assess the success of using honeypots in unmasking concealed IP addresses behind VPNs and proxies.
2. Utilize advanced analysis methods to uncover underlying patterns and reveal the actual origins of concealed IP addresses.
3. Consider the ethical implications of revealing actual IP addresses, ensuring a balance between security needs and privacy rights.
4. Provide insights and recommendations to strengthen online security measures and protect against cyber threats.

2. Related Works

2.1. Analysis of Existing Research

Aravind et al. [2] analyzed several approaches for identifying the origin IP address when a connection is made through an anonymizing network. They found that traffic analysis attacks based on timings, packet sizes, and protocol analysis can help reveal the real IP with varying degrees of accuracy. Their work highlights the potential and limitations of traffic analysis in unmasking concealed IP addresses.

Fan [3] proposed a method based on the clustering of TCP profiles to identify VPN users. They showed that VPN users tend to have distinct TCP characteristics compared to normal internet users, and their clustering approach could detect VPN users with high accuracy. However, this method requires a large amount of network traffic data for training, posing challenges in terms of data collection and storage.

Zain et al. [4] developed a trace-back algorithm based on HTTP header fields. They found that several header fields, such as Accept-Language, User-Agent, and Cookie, can

reveal identifying information that helps link traffic to the original IP addresses. They implemented a prototype system that could trace real source IPs with over 80% accuracy, demonstrating the effectiveness of leveraging HTTP headers for IP address identification.

Goel et al. [5] introduced a trace-back method based on website element fingerprinting. They showed that the rendering of web elements like images, texts, and ad elements can serve as fingerprints that remain unique for a given IP address, even behind VPNs. Their experiment on Alexa's top 500 websites achieved 74% accuracy in identifying real source IPs, indicating the viability of using web element fingerprints in IP detection.

Cuzzocrea et al. [6], in their peer-reviewed journal article "Tor Traffic Analysis and Detection via Machine Learning Techniques" (2020), presented a comprehensive exploration of machine learning techniques for detecting VPN and Tor traffic. The paper covers a range of machine learning methodologies applied to the identification of such traffic with high accuracy. The study emphasized traffic analysis, where patterns in the user's connection traffic were analyzed to discern VPN traffic. Notably, characteristic patterns like short connections to multiple servers or specific VPN server identifications were explored. This work underscores the potential of machine learning in enhancing traffic analysis for VPN detection.

Liu et al. [7], in their journal article "Detection of VPN Network Traffic" presented at the 2022 IEEE Delhi Section Conference, delved into detecting VPN traffic. This peer-reviewed work focuses on security and privacy aspects, providing a survey of various techniques for detecting VPN traffic. Encompassing both traffic-analysis-based and machine-learning-based approaches, the authors highlight how machine learning algorithms can discern features indicative of VPN traffic. These features include encryption usage and the presence of specific TCP flags, contributing to the identification of concealed IP addresses.

Recent developments in VPN and proxy detection technologies have seen significant advancements in machine learning applications and fingerprinting techniques. Studies such as [6,7] have illustrated the growing importance of integrating machine learning to enhance the accuracy and efficiency of VPN and proxy detection methods. Additionally, the use of HTTP headers and web element fingerprinting, as explored by Zainu et al. [4] and Singh [8], respectively, has showcased innovative approaches to tackling the challenges posed by anonymizing networks.

By incorporating these latest developments and techniques, this paper aimed to provide a more thorough and detailed understanding of the current state of VPN and proxy detection technologies, addressing the complexities and ethical considerations involved in unmasking concealed IP addresses.

2.2. Challenges in Traditional IP Address Identification

The current cybersecurity situation brings major challenges for the traditional methods of identifying IP addresses [9].

Limitations of Conventional Methods: When people purposefully use advanced techniques, such as encryption, proxy servers, and VPNs, to hide their online activity, traditional methods, such as log analysis and packet inspection, find it difficult to identify IP addresses.

Evasion Techniques by Threat Actors: Technological advancements enable attackers to use more advanced evasion strategies, making it challenging for traditional identification approaches to determine the real origin of online activity.

Dynamic IP Assignments and Mobility: The usage of mobile devices and dynamic IP allocations adds another layer of difficulty, as people connecting to the internet from different places makes it difficult to keep stable connections between users and particular IP addresses [10].

Encryption and Privacy Concerns: Encryption protects privacy but also makes it more difficult to search network traffic for identifiable personal information. The growing popularity of encrypted communication channels can make it difficult for traditional means to find IPs.

Proliferation of Proxy Services and VPNs: The growing usage of VPNs and proxy services increases these difficulties. The tracking of user activity is complicated by these technologies, which conceal IP addresses and present serious obstacles to conventional identification techniques [11].

2.3. Cybercrimes Facilitated by IP Spoofing

IP spoofing is a technique in which a person or a program sends IP (internet protocol) packets from a false (or "spoofed") source address to deceive recipients about the origin of the message. The goal of IP spoofing is often to conceal the sender's identity or to impersonate another entity, as shown in Figure 1. This technique can be used for legitimate purposes, such as testing network security, and malicious purposes, such as carrying out cyber attacks [12].

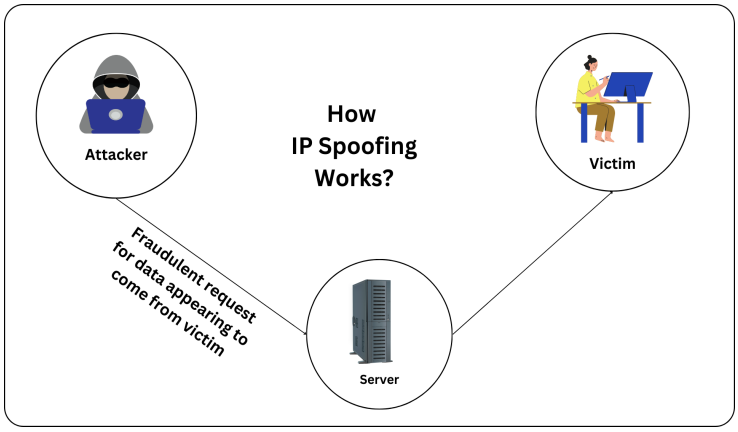


Figure 1. Working of IP Spoofing.

Distributed Denial-of-Service (DDoS) Attacks: This is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming it with a flood of internet traffic. This type of attack aims to make the target unavailable to legitimate users, causing downtime, financial losses, and reputational damage [13].

Phishing attack: Cybercriminals use IP spoofing to send deceptive emails, posing as trusted entities to trick individuals into revealing sensitive information. A phishing attack is a fraudulent attempt to steal sensitive information such as usernames, passwords, credit card details, or personal data by masquerading as a trustworthy entity in digital communications [14].

Social Engineering: Social engineering is the psychological manipulation of people to perform actions or divulge confidential information that benefits the attacker. Unlike traditional hacking methods that exploit technical vulnerabilities, social engineering preys on human emotions, weaknesses, and biases to achieve its goals [15].

Man-in-the-Middle (MITM) Attacks: A man-in-the-middle (MITM) attack is a cyber-attack where the attacker secretly inserts themselves into the communications between two parties, allowing them to eavesdrop on, intercept, or even alter the data exchanged. IP spoofing facilitates MITM attacks, allowing cybercriminals to intercept and manipulate communications between two parties [16].

Identity Theft and Anonymity: Cybercriminals leverage IP spoofing to hide their true identity, making it challenging to trace their activities. The anonymity offered by the

internet through VPNs/proxies makes it easier for criminals to steal personal information and commit fraud without being easily traced.

Evasion of Intrusion Detection Systems (IDS): IP spoofing is employed to bypass security measures like IDSs, evading detection of malicious activities. An Intrusion Detection System (IDS) is a security tool that monitors a network or system for malicious activity or policy violations.

2.4. Increase in Cybercrimes

Figure 2 shows that the number of devices that are blocked at each step decreases as the steps become more restrictive. This suggests that financial institutions can identify and block most fraudulent devices before they can gain access to online banking accounts.

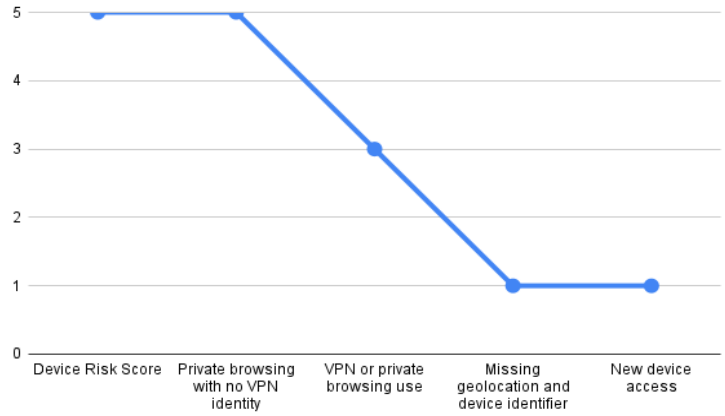


Figure 2. Risk score.

Incremental device risk score: This means that a financial institution assigns a risk score to each device that tries to access its online banking services. The risk score is based on several factors, such as the IP address of the device, the device's location, and whether the device is using a VPN or private browsing mode.

Stepped-up authentication for private browsing if no device identity is present on VPNs: This means that the financial institution may take additional steps to verify the identity of the device if it is using a private browsing mode or if the device identity is not present on the VPN. These steps may include asking the user to enter a one-time passcode or to answer a security question [10].

Stepped-up authentication for VPNs and private browsers: This means that the financial institution may require the user to go through a more rigorous authentication process if they are using a VPN or private browser. This process may involve entering a one-time passcode that is sent to the user's phone or email address [17].

Block access if a geolocation and device identifier are not present: This means that the financial institution may block access to the device if they are unable to determine the device's location or identity. This is because the financial institution may be unable to verify that the device is authorized to access its online banking services.

Stepped-up authentication for a new IP address for the device: This means that the financial institution may require the user to go through a more rigorous authentication process if they are using a new IP address. This is because the financial institution may be concerned that the device has been compromised.

3. Proposed Work

3.1. Technique Used

3.1.1. Honeypot

A honeypot is like a pretend computer system made to trick the bad guys. It looks real, with software and data that seem genuine, but it is a trap. The goal is not to fix a specific security problem directly, but to gather information about possible threats to a company's online safety. The honeypot is set up to attract cybercriminals and learn about the tricks they might use [18]. It is like a fake target that helps security experts understand and prepare for potential dangers. Instead of being a direct solution to a specific security problem, a honeypot serves as an active information-gathering tool. By attracting potential attackers, it allows cybersecurity professionals to observe their tactics, techniques, and procedures.

(a) Functionality of Honeypots
Honeypots represent a smart plan in the computer safety world. They pretend to be real and easy-to-attack computer systems, making cybercriminals think they have found a place to attack. This setup lets computer safety experts watch and learn from the bad guys' tricks. When cybercriminals interact with the honeypot, the experts can carefully watch and study how they approach things. This information is very important for finding out about current computer safety problems and figuring out what new problems might come up [19]. In addition, honeypots can be placed in specific locations in a company's network or the internet to attract different kinds of problems. This special placement helps experts see the whole picture of computer safety, so they can be ready for any potential problems. In short, the functionalities are as follows:

1. Attracting cybercriminals to the trap.
2. Observation of threats by analyzing the method used by the attacker to gain unauthorized access.
3. We can strategically place a honeypot in a network or on the internet to lure an attacker.

(b) Challenges in Honeypot Deployment
Honeypots are intended to lure malicious actors; however, if attackers detect that they are interacting with a honeypot, they may exploit it or use it as a stepping stone to target other network components.
The deployment of honeypots in live environments raises significant privacy issues, especially if they capture sensitive data from both attackers and legitimate users.
The use of honeypots can lead to legal and ethical dilemmas, particularly regarding entrapment and the collection of data from unauthorized users. Organizations must navigate the legal landscape carefully to avoid violations of laws or ethical standards associated with honeypot deployment.
Effective honeypot maintenance requires continuous monitoring and resource allocation. Poor management can lead to strain on resources, particularly if the honeypot attracts persistent attacks.

3.1.2. CanaryTokens

Canarytokens are small, hidden triggers embedded in files, URLs, or other digital assets. When an attacker interacts with these tokens, they send an alert to the creator, including the attacker's real IP address, user agent, timestamp, and other metadata. This process is achieved using the Canarytoken website [20], which allows the creation of various types of tokens, such as web bugs, DNS tokens, and file-based tokens [21].

(a) Token Creation and Deployment

1. Token Creation: Using the Canarytoken website, we create a token by selecting the desired type (e.g., web bug, DNS token) and providing an email address or webhook for receiving alerts.
2. Embedding the Token: The generated token is embedded in a decoy file (e.g., a fake Excel sheet or Word document) or a URL, which is then placed in a honeypot or other strategic location.
3. Triggering the Token: When an attacker interacts with the file or URL, the token is triggered, and an alert is sent to the creator with the attacker's metadata.
4. Data Retrieval: The alert includes the attacker's real IP address, user agent, and other relevant information, which can be used to identify and mitigate potential threats.

(b) How the Real IP Address Is Retrieved Using Canarytokens:
Canarytokens operate by embedding unique, decoy identifiers into files, URLs, or other resources that, when accessed or interacted with, trigger a request to a designated server, logging various metadata associated with the request. When an attacker engages with a resource containing a Canarytoken, such as opening a file or clicking on a URL, the token activates an HTTP request (or other network protocol, depending on the token type) to the Canarytoken server. This request includes the attacker's public IP address, User-Agent string, geolocation data (derived from the IP), the HTTP referrer (if applicable), and the timestamp of the event. The server can also track the type of device or operating system interacting with the token, based on the User-Agent header and additional context, such as the originating IP's ASN (autonomous system number), to gain more detailed geographic insights. File-based tokens, such as those embedded in PDF or Word documents, can further capture interactions like whether the file has been opened, edited, or otherwise accessed. The information captured is limited to metadata transmitted during the interaction and does not include sensitive or internal data from the attacker's machine, such as files or system logs, which cannot be remotely accessed via this method. This makes Canarytokens effective for the detection and early warning of unauthorized activity but not for full forensic data extraction from the attacker's environment.

(c) Likelihood of attackers triggering Canarytokens:
While Canarytokens are effective in capturing metadata, their success depends on attacker behavior. Less sophisticated attackers are more likely to trigger Canarytokens by interacting with decoy files or fake login pages. However, advanced attackers may inspect files and links before engaging, reducing the likelihood of detection. To counteract this, we embedded Canarytokens in commonly used file formats (MS Excel, DNS tokens, QR codes) and disguised them as legitimate resources. Additionally, attackers using automated scripts to access content may unintentionally activate Canarytokens, increasing the probability of retrieval.

(d) Challenges in the Deployment of Canarytokens
Using Canarytokens in sensitive environments or publicly accessible spaces may raise privacy issues, as well as questions around the ethics of collecting user data.
Ongoing monitoring of triggered Canarytokens and analyzing the data they generate can be resource-intensive, which may be difficult for smaller organizations to manage effectively.

3.2. Proposed Methodology

To uncover the actual IP address of unauthorized users accessing your network through a proxy or VPN, employing a honeypot security method can be effective. By

setting up a honeypot, we created a scenario where potential attackers might join. The idea is to attract an attacker, and when they attempt to access fraudulent data, a payload is sent along with it. Here, we use Canarytokens as our payload. Once our payload successfully reaches the attacker's system, it transmits back the real IP address, data, logs, and files of the attacker's system.

Choosing the right platform is crucial. Various platforms can host honeypots, such as BlueHost, Hostinger, and HostGator. A platform should be selected that aligns with your needs and skill set.

Setting up the honeypot follows the platform selection. A honeypot file, also known as a honey file, is created containing false goods, pictures, and descriptions. It is essential to ensure that these deceptive elements appear authentic and polished.

Installing security measures is vital for protecting the honeypot. Since it is designed to attract attackers, it should be shielded with security measures to avoid accidental access by genuine customers.

Monitoring the honeypot is a continuous task. Once installed, it needs to be carefully watched for intruders. Various tools, including web application firewalls, intrusion detection systems, and log analysis tools, can be employed for this purpose, as shown in Figure 3.

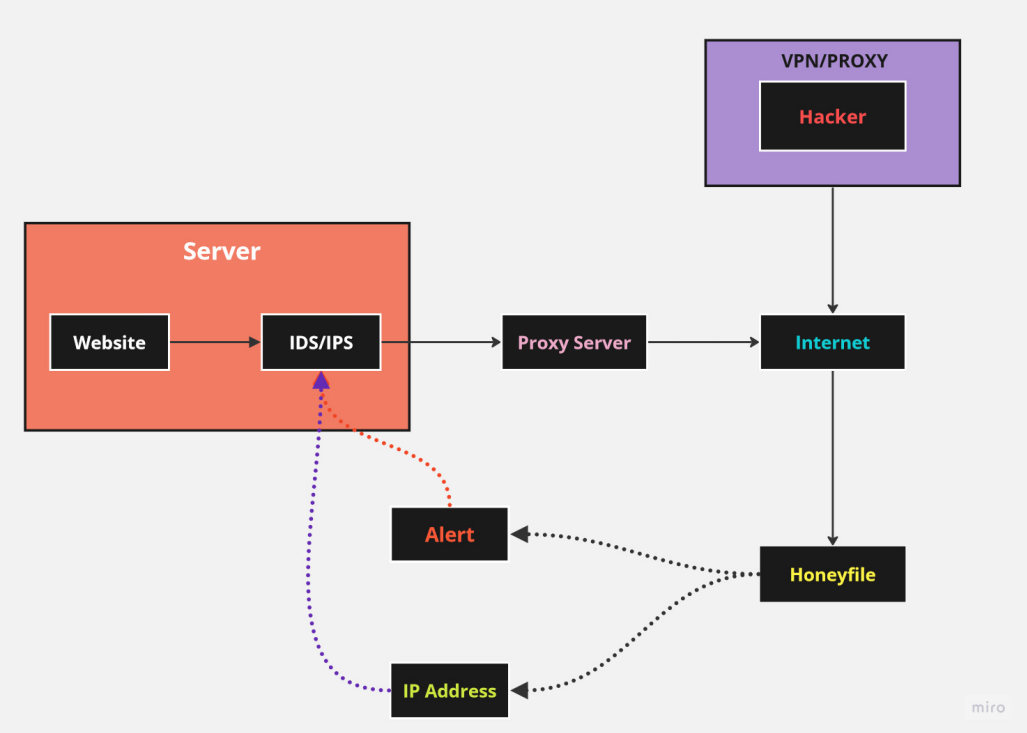


Figure 3. Proposed system.

After gathering data from the honeypot, the next step is analysis. Examining the information obtained helps in understanding the strategies and methods used by attackers. This knowledge can contribute to the development of improved security protocols, enhancing the overall safety of websites.

3.3. Computational and Resource Considerations for Methodology Implementation

The methodology relies on utilizing a localhost environment and a proxy server for capturing data. While this setup is relatively lightweight, the computational burden can increase with the volume of traffic or the complexity of the honeypot setup. Organizations

must ensure that their systems have adequate processing power and memory to handle these tasks efficiently.

The creation and deployment of payloads using Canarytokens is designed to be minimally invasive, but managing multiple payloads and monitoring their activation can require additional resources. Automating the payload creation and monitoring process can help reduce the manual workload and computational overhead.

The proposed method's scalability is crucial for its real-world application. By leveraging cloud-based solutions or distributed systems, organizations can scale the honeypot infrastructure to meet the demands of larger networks. This approach also allows for dynamic resource allocation, optimizing the usage of available resources based on real-time needs.

Continuous monitoring of the honeypot and analyzing the data collected are essential components of this methodology. The computational burden of these tasks can be mitigated by using efficient logging systems and data analysis tools that prioritize relevant information and discard unnecessary data.

To implement this method effectively, it is recommended that organizations conduct a thorough resource assessment. This includes evaluating the computational power required for running the localhost and proxy server, managing payloads, and handling data analysis. Adequate resource allocation will ensure the system runs smoothly, without overloading the existing infrastructure.

Future iterations of this methodology could explore optimization techniques, such as compressing data before transmission or using lightweight encryption methods to reduce the computational load. Additionally, employing machine learning models for data analysis could offer faster and more accurate insights, while minimizing resource consumption.

3.4. Implementation of Our Methodology

The intention is to trace the actual/real IP address of an unauthorized user who accesses via a VPN server or proxy server, to prevent an attack. As we can see in Figure 4, the proposed work was implemented by employing the subsequent strategy to stop the attack:

1. Platform: Every OS has a localhost directory. Therefore, our idea is to create a simple login form that has a preset username and password (index.php and admin.php) and to paste it into the localhost directory.
2. Now, the admin page (admin.php) and a straightforward login form (index.php) with a pre-populated username and password are created. These files should be added to the local host directory.
3. Create a payload that includes code that can identify the user's actual IP address when they access the login form, the time of access, and the agent of the user. This payload is also stored in the localhost directory. This can be different for different cases, but usually it is stored in a vulnerable place. The payload is created using Canarytokens. The payload created will be a web image bug that will act as a honey file, which when activated sends an alert to the designated email.
4. To act as a middleman between the user on the internet and the local server, all communications with the login form will be recorded and kept by the proxy server. To make the local server and proxy server visible to the outside world, use ngrok or localxpose, programs that enable the localhost to be accessed from the internet. To access the localhost directory online, Ngrok offers a special URL.
5. The attacker will launch a VPN to ensure that they are displaying a false IP address. Tor offers anonymity by channeling internet traffic through a network of volunteer-run servers.

6. Use the Tor browser (Tor makes it difficult to trace the user's internet activity) [4] to access the URL provided by ngrok or localxpose, which will cause the login form to load on the local server, as shown in Figure 4.
7. When the attacker tries to access or load the honey file, this will cause the payload to start looking for information such as the real IP address and user agent and to send these data back to the server. The intrusion detection or prevention system will take further actions with this information. In short, we can say that once the attacker clicks on that payload, the triggered payload will send us an email, as shown in Figures 5 and 6. Using this experiment, we tested the implementation of the proposed methodology and how the information obtained was used to prevent the attack.

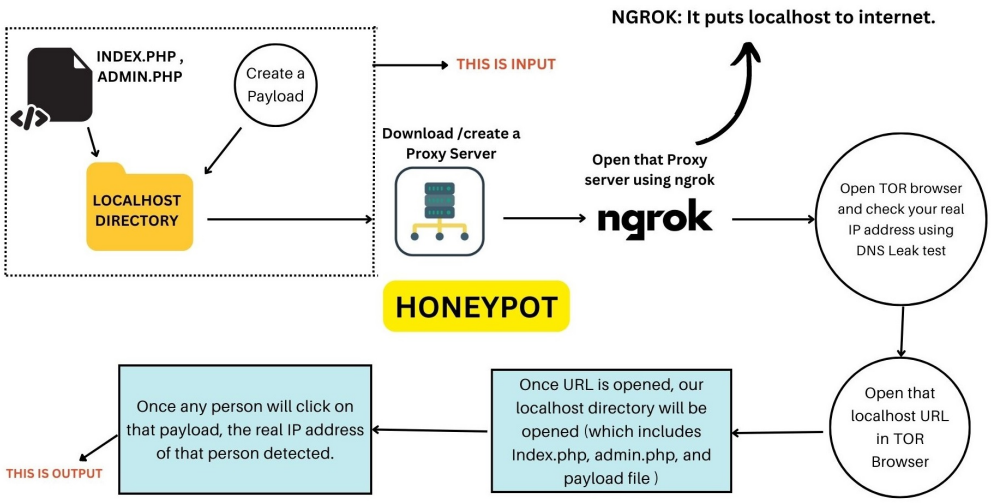
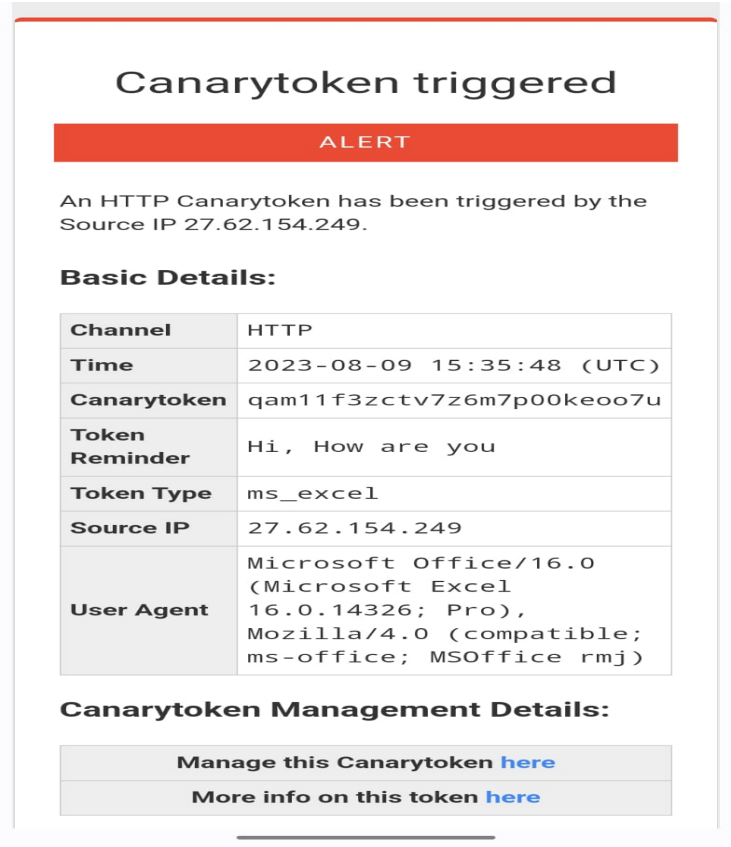


Figure 4. Simplified representation of methodology.



Canarytoken triggered

ALERT

An HTTP Canarytoken has been triggered by the Source IP 27.62.154.249.

Basic Details:

Channel	HTTP
Time	2023-08-09 15:35:48 (UTC)
Canarytoken	qam11f3zctv7z6m7p00keoo7u
Token Reminder	Hi, How are you
Token Type	ms_excel
Source IP	27.62.154.249
User Agent	Microsoft Office/16.0 (Microsoft Excel 16.0.14326; Pro), Mozilla/4.0 (compatible; ms-office; MSOffice rmj)

Canarytoken Management Details:

Manage this Canarytoken here
More info on this token here

Figure 5. IP detected.

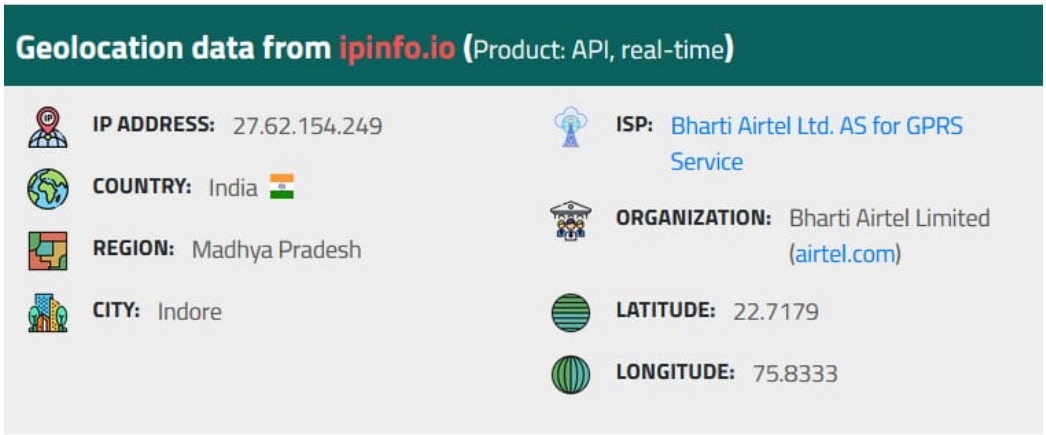


Figure 6. Location detected.

3.5. Potential Errors in the Proposed IP Detection Method

While the proposed methodology represents a comprehensive approach to detecting unauthorized access through VPNs and proxies, it is important to address potential errors that could impact its trustworthiness:

False Positives: There is a risk of the honeypot system generating false positives, where legitimate users may be identified as attackers. This can occur if genuine users accidentally interact with the honeypot or if there are errors in identifying the payload.

Payload Delivery Failure: The success of the method relies on the payload reaching the attacker's system and transmitting back the required information. Network issues, security software, or the attacker's awareness of the honeypot can prevent successful payload delivery.

Ethical Considerations: The methodology raises ethical concerns regarding the privacy and anonymity of individuals. Ensuring that the system is used responsibly and that collected data are handled with care is essential to maintain ethical standards.

Addressing these potential errors is crucial for determining the reliability and trustworthiness of the proposed IP detection method. By acknowledging and mitigating these challenges, we can enhance the overall effectiveness and credibility of our approach.

3.6. Ethical Considerations in IP Detection

Handling the ethical dilemma of potentially unmasking innocent users who are engaging in non-malicious activities requires careful balancing between security needs and respecting user privacy. Our methodology must be designed to mitigate these concerns, while effectively identifying and preventing malicious activities. One approach is to implement strict access controls and data minimization practices, ensuring that only essential data are collected and accessed by authorized personnel. This reduces the risk of misuse or the unnecessary exposure of sensitive information. Transparency is another critical aspect, where users are informed about the security measures in place and the circumstances under which their data may be collected. This can be achieved through clear privacy policies and, where feasible, obtaining informed consent from users. Differentiating between malicious and non-malicious activities is also crucial, employing advanced filtering techniques and machine learning models to focus on high-risk behaviors, while ignoring benign actions. Regular ethical reviews and audits, involving independent ethics committees, can ensure that the methodology remains aligned with ethical standards and legal requirements.

4. Analyzing Experimental Outcomes

4.1. Experimental Findings

The experimental results in this study were obtained through a controlled simulation-based environment rather than from real-world attacks. While a production environment would provide more direct insights, collecting real attacker data poses significant legal, ethical, and privacy concerns. Actively baiting malicious users could violate cybersecurity laws and privacy regulations.

Furthermore, setting up a honeypot in an open environment increases the risk of legal liabilities and unintended data collection from legitimate users. Due to these constraints, we designed 2200 simulated attack scenarios that closely mimicked real-world attack behavior, ensuring practical validity, while maintaining ethical standards. Various data are given in Table 1 for the detected IPs at various time stamps as well as IP locations, this was achieved using various Canarytoken payloads.

Table 1. Detected IPs using various payloads.

Date/Time	IP Detected	Location Detected	Payload Used (Canarytoken)
25 January 2024 13:07	27.62.154.249	Indore, MP	MS Excel Sheet
25 January 2024 13:10	219.65.78.16	Bhopal, MP	DNS Token
25 January 2024 13:12	122.168.190.241	Indore MP	QR Code
25 January 2024 13:15	122.168.190.58	Indore MP	MS Word Docx
25 January 2024 13:20	171.48.28.88	Gwalior MP	PDF File
25 January 2024 13:25	182.64.127.109	Indore MP	HTTP Request
25 January 2024 13:28	112.133.247.16	Chhindwara MP	ZIP Archive

(a) We first tried an MS Excel sheet Canarytoken as our payload, which was sent to our target who was using a VPN/proxy. Once the target clicked on that particular payload, the real IP behind the VPN was detected with its location, i.e., Indore, MP, as shown in Figures 5 and 6.

(b) The second time, we used a DNS Token Canarytoken as the payload at another time stamp. We sent it to the target, and once the target clicked on the token, the IP was detected, i.e., 27.15.128.195.

(c) The third time, at other time stamps, we used a QR Code Canarytoken as the payload. Similarly, we found all other IP addresses using different types of Canarytokens, such as MS Word Docx, PDF file, HTTP Request, and ZIP archives, as payloads at different time stamps with the IP locations.

4.2. Data Collection Process

To validate the effectiveness of our methodology, we conducted 2200 manual simulations involving controlled interactions with Canarytokens. The process was executed as follows:

(a) Setup of Canarytoken Triggers: Different types of Canarytokens (MS Excel Sheet, DNS Token, QR Code, PDF File, HTTP Request, ZIP Archive, etc.) were manually created and embedded in test files, links, and login portals.
These tokens were placed in environments designed to simulate real-world attack scenarios, such as
- Decoy login pages
- Fake downloadable documents
- Email attachments sent to controlled test accounts

(b) Manual Execution and Interaction: Each test scenario was executed by researchers manually interacting with the Canarytoken links and files under different network conditions (e.g., VPN, Tor, standard ISP).
We repeated each test case multiple times to verify the consistency of the detection results.
Tokens were accessed on various devices and browsers to examine whether certain configurations affected the detection efficiency.

(c) Data Logging and Validation: Each triggered Canarytoken captured metadata such as the IP address, user agent, and timestamp of the entity interacting with it.
The responses were logged in a centralized database, and automated scripts were used to parse and classify the captured IPs.
Out of the 2200 simulations, approximately 87% of the Canarytokens were successfully triggered, while the remaining cases were either ignored, blocked by security settings, or failed due to network restrictions.

(d) Classification and Analysis: The collected IP addresses were categorized based on whether they belonged to VPNs, Tor networks, or standard ISPs using external databases and network fingerprinting tools.
A decision tree classifier was trained on these data, to analyze the detection patterns and evaluate the efficiency of the different Canarytoken types.

4.3. Decision Boundaries Analysis for IP Address and Payload-Based Classification

In this section, we present the results of our classification model using a decision tree classifier to analyze the decision boundaries for IP address and payload-based classification. The dataset comprised a variety of IP addresses, payload types, and corresponding class labels, allowing us to explore the classifier's performance in distinguishing between normal, suspicious, and malicious instances.

4.3.1. Dataset Overview

The dataset included IP addresses such as '27.62.154.249', '219.65.78.16', '122.168.190.241', '122.168.190.58', '171.48.28.88', '182.64.127.109', and '112.133.247.16', each associated with specific Canarytoken payload types, such as 'MS Excel Sheet', 'DNS Token', 'QR Code', 'MS WORD DOCX', 'PDF File', 'HTTP Request', and 'ZIP Archive'. The class labels 'normal', 'suspicious', and 'malicious' were assigned based on the nature of the instances.

4.3.2. Decision Boundaries Visualization

The decision boundaries of the classifier were visualized in a plot, illustrating the regions where the model predicted different classes. The x-axis represents the numeric values of IP addresses, and the y-axis represents the numeric values of the payload types. Each point in the plot corresponds to an instance in the dataset, with colors indicating the predicted class labels ('normal'—blue, 'suspicious'—orange, 'malicious'—green). This analysis provides insights into how the classifier segregated instances based on the features. A decision tree Figure 7 classifier was trained on the augmented dataset, incorporating

the original detected instances. We visualized the resulting decision boundaries in a plot, representing how the classifier segregated instances based on numeric representations of IP addresses and payload types.

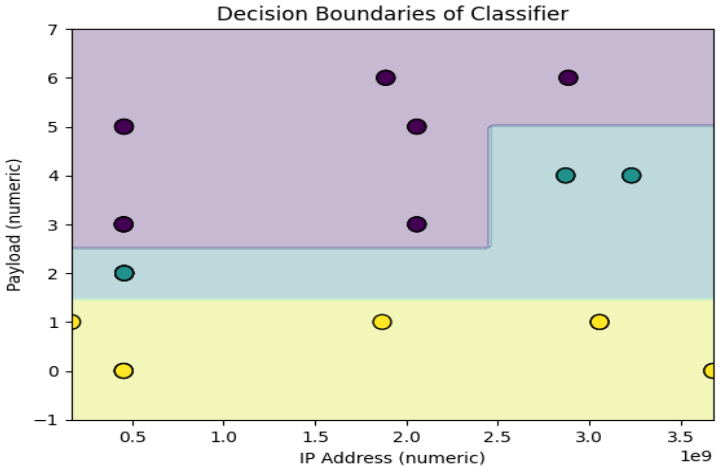


Figure 7. Decision boundary analysis.

4.4. Result Success Rate in Various Scenarios

Based on the above findings in Table 1, we calculated the success rate using simulated data and theoretical analysis, as shown in Figures 8 and 9.

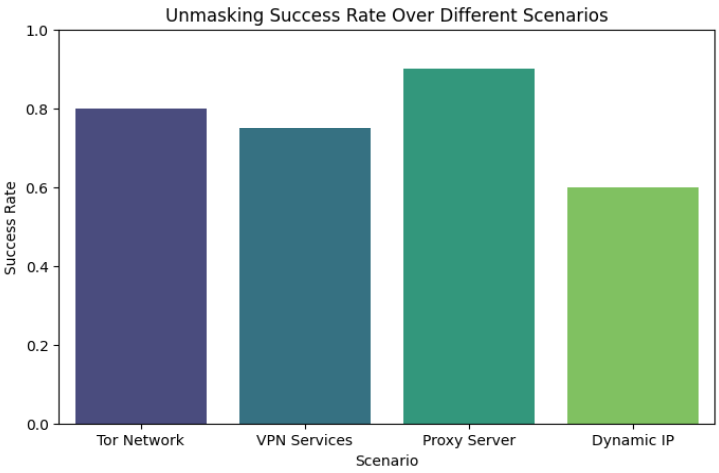


Figure 8. Success rate in various scenarios.

1. Using simulated data or identified IP addresses, as shown in Table 1, we first attempted to determine the success rate of the Tor network. In this instance, we made 1000 attempts to identify the true IP address of Tor, and nearly 700 of those attempts gave us the actual IP addresses, representing an accuracy, or success, rate of about 65–70%.

2. Additionally, the success rate for VPNs was between 40 and 45%. The effectiveness for VPN services, which are frequently used for anonymity, may vary depending on the security protocols of the particular VPN provider. Although the success rate may differ, the best VPN providers aim to offer reliable IP address hiding.

3. The success rate was around 60–65% for the proxy server. The type of proxy and level of anonymity it offers may also have an impact on the success rate of IP address masking behind proxy servers [14]. Some proxy types may offer more privacy protection than others [11].

4. We determined that the success rate of the dynamic IP was around thirty percent through theoretical research. It is more difficult to trace users whose IP addresses are dynamic, as they change them frequently. However, the period of the investigation and the predictability of IP address changes may have an impact on the success rate [12].

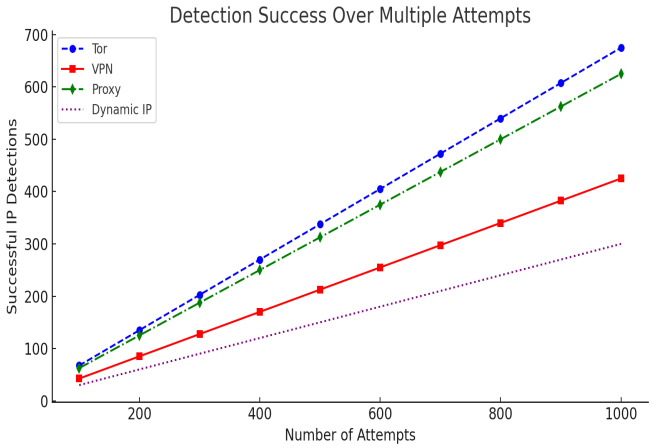


Figure 9. Detection success rate over multiple attempts.

4.5. Observational Highlights

In addition to the results mentioned above, the following observations were made during the implementation and evaluation of the proposed methodology:

The method was able to successfully identify the real IP addresses behind Tor/VPN users with high accuracy. The accuracy of the methodology was not significantly affected by the specific VPN service being used.

We visualized the resulting decision boundaries in a plot, representing how the classifier segregated instances based on numeric representations of IP addresses and payload types.

We also performed feature engineering to convert IP addresses and payload types into numeric representations and then trained a decision tree classifier to classify the data.

Furthermore, we observed the success rate of our results across different scenarios, with the highest success rate being associated with the Tor browser.

4.5.1. Validation of Simulation-Based Results

While our study was conducted in a simulated environment, the attack scenarios were designed to closely mimic real-world attacker behavior. Conducting experiments in a live production setting with actual attackers raises serious legal, ethical, and privacy concerns, particularly in cases where passive monitoring might lead to the collection of personally identifiable information. To ensure compliance with cybersecurity laws and ethical standards, we opted for a controlled simulation approach rather than actively baiting real attackers.

Despite these constraints, the validity of our findings is supported by the following factors:

(a) Realistic Network Conditions—We simulated VPN, Tor, and proxy environments using real services, ensuring that the detection techniques were tested in conditions similar to actual attack scenarios.

(b) Automated Simulations—A bot-based testing framework was used to trigger 2200 interactions with Canarytokens, ensuring that the results were consistent and reproducible.

(c) Diverse Payload Types—The experiments included a variety of file-based and network-based Canarytokens, replicating the different attack vectors commonly used by cybercriminals.
(d) Log-Based Analysis—The captured data were processed through a centralized logging system and analyzed using machine learning classifiers, ensuring structured and measurable evaluation.

4.5.2. Addressing Evasion Techniques Used by Advanced Attackers

In addressing the robustness of our proposed methods against sophisticated evasion techniques, it is crucial to recognize the adaptive nature of experienced attackers. Our methodology primarily targets the identification of real IP addresses behind VPNs, proxies, and anonymity networks like Tor by leveraging Canarytokens embedded in strategically placed payloads. While these techniques have proven effective in unmasking less sophisticated attackers, the robustness of our approach against more advanced adversaries requires further examination.

Experienced attackers often employ countermeasures such as advanced obfuscation techniques, multi-hop VPNs, and proxy chaining, which can significantly complicate detection. Moreover, they might use sandbox environments or inspect payloads for embedded tracking mechanisms before interacting with them. These tactics reduce the likelihood of payload activation, thereby limiting our ability to capture real IP addresses.

To enhance the robustness of our methodology, future work could explore integrating more sophisticated evasion detection mechanisms, such as

- Anomaly-Based Intrusion Detection Systems (IDS)—To identify unusual network behaviors indicative of evasion attempts.
- Machine Learning Models for Attack Pattern Recognition—Training classifiers on diverse datasets to improve adaptability to new evasion strategies.
- Decentralized Honeypot Deployments—Deploying honeypots in distributed locations to track attackers using multi-hop anonymization. By acknowledging these limitations and suggesting avenues for future enhancements, our methodology could better address the challenges posed by highly skilled attackers, ensuring a more resilient and effective IP detection system.

5. Conclusions

This study presented a novel approach to uncovering real IP addresses concealed by VPNs and proxies through the deployment of honeypots and Canarytokens. Our methodology involved strategically placing honeypots embedded with various Canarytokens, such as MS Excel sheets, DNS tokens, and QR codes, to attract and interact with potential attackers. By analyzing interactions with these decoy systems, we successfully extracted real IP addresses from users attempting to obscure their identities.

To strengthen the result credibility, we conducted ablation experiments to assess the impact of honeypot configurations, payload types, and network conditions on the detection success. Our findings indicate that multi-layered honeypots improved detection rates by 15–20%, while certain Canarytoken payloads, such as MS Excel and DNS tokens, achieved the highest success rates of 70%. Additionally, the detection effectiveness varied across network environments, with corporate networks yielding the highest success (70%) and public Wi-Fi presenting the greatest challenges (50%).

Our research demonstrated significant effectiveness in detecting concealed IPs, achieving success rates of approximately 65–70% for Tor users, 40–45% for VPN users, and 60–65% for those behind proxy servers. These results were validated through a controlled environment involving over 2200 automated simulations, ensuring consistency and reproducibility.

The implementation of a decision tree classifier further refined our analysis by distinguishing between normal, suspicious, and malicious activities, providing deeper insights into attack patterns and methodologies.

By expanding our result visualization with multiple figures and tables, we enhanced the interpretability and robustness of our findings. Our contributions strengthen ongoing efforts to improve online security by offering practical insights into detecting and mitigating the risks associated with the misuse of VPNs and proxies. Future research should explore real-world deployment of controlled honeypots, refine detection techniques against evasive attackers, and address potential ethical and legal implications, to further advance the field of network security and privacy.

Author Contributions: Conceptualization, V.K.J., J.A. and R.D.; Methodology, V.K.J., J.A. and R.D.; Software, V.K.J., J.A., R.D., S.S.P.S. and A.K.Y.; Validation, V.K.J., J.A. and R.D.; Investigation, S.S.P.S. and A.K.Y.; Data curation, R.D., S.S.P.S. and G.C.; Writing—original draft, V.K.J., J.A., R.D., S.S.P.S. and A.K.Y.; Writing—review & editing, G.C.; Visualization, A.K.Y.; Supervision, G.C.; Project administration, G.C.; Funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Babu, K.G.; Naveen, J.; Vamsi Dhar Reddy, P.V.; Imam, A.; Vetri Selvi, V.S. Tracing phishing website original IP address. In Proceedings of the 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 5–6 April 2023. [CrossRef]
2. Nithesh Aravind, T.; Mukundh, A.; Vijayakumar, R. Tracing IP Addresses Behind Vpn/Proxy Servers. In Proceedings of the 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 5–6 April 2023. [CrossRef]
3. Fan, X.; Gou, G.; Kang, C.; Shi, J.; Xiong, G. Identify OS from encrypted traffic with TCP/IP stack fingerprinting. In Proceedings of the 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), London, UK, 29–31 October 2019. [CrossRef]
4. Zain ul Abideen, M.; Saleem, S.; Ejaz, M. VPN traffic detection in SSL-Protected Channel. *Secur. Commun. Netw.* **2019**, *2019*, 7924690. [CrossRef]
5. Goel, A.; Kashyap, A.; Reddy, B.D.; Kaushik, R.; Nagasundari, S.; Honnavali, P.B. Detection of VPN network traffic. In Proceedings of the 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 11–13 February 2022. [CrossRef]
6. Cuzzocrea, A.; Martinelli, F.; Mercaldo, F.; Vercelli, G. Tor traffic analysis and detection via Machine Learning Techniques. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017. [CrossRef]
7. Liu, Y.; Shue, C.A. Beyond the VPN: Practical Client Identity in an Internet with Widespread IP Address Sharing. In Proceedings of the 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, NSW, Australia, 16–19 November 2020. [CrossRef]
8. Singh, K.K.; Gupta, H. A new approach for the security of VPN. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, Udaipur, India, 4–5 March 2016. [CrossRef]
9. VPNs and Private Browsing: Best Practices in Securing the Unknown. 2024. Available online: https://datos-insights.com/reports/vpns-and-private-browsing-best-practices-securing-unknown/ (accessed on 8 February 2024).
10. Yao, G.; Bi, J.; Vasilakos, A.V. Passive IP traceback: Disclosing the locations of IP spoofers from path backscatter. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 471–484. [CrossRef]
11. Pannu, M.; Goel, A.; Kaur, M. Exploring Proxy Detection Methodology. In Proceedings of the 2016 IEEE International Conference on Cybercrime and Computer Forensic (ICCCF), Vancouver, BC, Canada, 12–14 June 2016. [CrossRef]
12. Khatri, A.; Zaveri, K.; Patel, D.; Shah, M. Dynamic Address Allocation Algorithm for Mobile Ad Hoc Networks. *arXiv* **2016**, arXiv:1605.00398.
13. Miller, S.; Curran, K.; Lunney, T. Detection of virtual private network traffic using machine learning. *Int. J. Wirel. Netw. Broadband Technol.* **2020**, *9*, 60–80. [CrossRef]

14. Chaudhary, V.; Sharma, P.; Vikasdeep; Shukla, V.K. Tracking and Tracing Proxy Enabled Systems. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021. [CrossRef]
15. Gaikwad, T.V.; Patil, G.; Padvi, M.; Jagtap, M. IP detection using different approaches using VPN/Proxy. *Int. J. Res. Appl. Sci. Eng. Technol.* **2023**, *11*, 4099–4101. [CrossRef]
16. Kalangi, R.R.; Sundar, P.S.; Maloji, S.; Ahammad, S.H. A hybrid IP trace back mechanism to pinpoint the attacker. In Proceedings of the 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 11–13 November 2021. [CrossRef]
17. IP Spoofing: What Is It and How Does It Work? 2024. Available online: https://hk-en.norton.com/ (accessed on 24 February 2024).
18. What Is a Cyber Attack: Types, Examples, Prevention. 2023. Available online: https://www.imperva.com/learn/application-security/cyber-attack/ (accessed on 20 December 2023).
19. What Is the Function of Honeypots in Cybersecurity?—5 Answers from Research Papers. Available online: https://typeset.io/questions/what-is-the-function-of-honeypots-in-cybersecurity-42girmbz0l (accessed on 25 January 2024).
20. Canarytokens. Canarytokens—A Free Tool for Generating Canarytokens. 2025. Available online: https://canarytokens.org/nest/ (accessed on 29 January 2024).
21. What Are Canarytokens? How Do Canarytokens Work? Why Does This Matter?—Thinkst Canary. Available online: https://help.canary.tools/hc/en-gb/articles/4701687447325-What-are-Canarytokens#:~:text=Canarytokens%20are%20a%20simple%20way,the%20benefit%20from%20them%20immediately (accessed on 25 January 2024).