Radboud University Nijmegen

Faculty of Science

Detection, Analysis and Measurement of DNS Tunneling Techniques

Master's Thesis Cyber Security

Author:
Damianos Nikou
damianos.nikou@ru.nl
s1083196

Supervisor:
Pol Van Aubel
pol.vanaubel@ru.nl
Second Reader:
Harald Vranken
harald.vranken@ru.nl
External Supervisor:
Moritz Müller
moritz.muller@sidn.nl

October 25, 2024

Contents

1 Introduction **2**

2 Background **3**
2.1 DNS Protocol . . . 3
2.2 DNS Tunneling . . . 4
2.2.1 DNS Tunneling Tools . . . 4
2.3 State-of-the-Art Rules on DNS Tunneling Detection . . . 5

3 DNS Testbed **6**
3.1 Iodine DNS Tunneling Tool . . . 7
3.2 Supporting Tools . . . 8
3.2.1 Tcpdump . . . 8
3.2.2 Wireshark . . . 8
3.2.3 SSH (Secure Shell) . . . 8
3.2.4 SCP (Secure Copy) . . . 8

4 Detection process of DNS Tunneling **8**
4.1 ENTRADA . . . 9
4.2 Prefiltered data on ENTRADA . . . 9
4.3 Transformation of State-of-the-Art Rules to Custom Rules . . . 10
4.4 Detection Rulesets on DNS Testbed and ENTRADA tool . . . 10

5 Validation **12**
5.1 Validation custom rules on DNS testbed . . . 12
5.2 Non-Implemented Rules . . . 13
5.2.1 Rule 3 . . . 13
5.2.2 Rule 7 . . . 13
5.3 Testing custom rules on .nl traffic . . . 14
5.4 Validation of custom rules on Public DNS tunneling server . . . 15

6 Measurements on .nl traffic **16**
6.1 Results on .nl traffic . . . 16
6.2 First Phase: Specific Dates Measurements . . . 17
6.2.1 Top 6 countries . . . 17
6.2.2 Top Query Types . . . 20
6.2.3 Total IPs . . . 22
6.3 Second Phase: Single Day Measurements . . . 24
6.3.1 Top 6 countries . . . 24
6.3.2 Top Query Types . . . 27
6.3.3 IP addresses of a specific date . . . 29
6.3.4 Unique Domain Names of a single day . . . 30
6.4 Third Phase: Specific Domain Name Measurements . . . 31
6.4.1 Top 6 Countries . . . 31
6.5 Summary of Measurements . . . 32

7 Ethics **33**

8 Conclusion **33**

9 Future Work **35**

10 Appendix 1 **38**

1 Introduction

The Domain Name System (DNS) is a fundamental Internet protocol for translating human-readable domain names into IP addresses. Nearly all Internet-connected systems utilize it, which is integral to every network linked to the Internet. DNS is defined in RFC 1035[1].
Furthermore, the DNS operates within a hierarchical infrastructure known as the DNS hierarchy. This structure organizes the domain names into a tree structure, allowing for efficient and decentralized resolution over the Internet. The DNS hierarchy consists of multiple levels, including Root domains, Top-Level domains (TLDs), Second-Level domains and subdomains. Each level of this hierarchy is responsible for managing a specific segment of the domain namespace by contributing to the overall functionality and scalability of the DNS system. We explain further details for DNS protocol in section 2.1.
However, if a DNS server comes under the control of an attacker, it can be exploited to disrupt normal operations. Attackers can redirect computers to fake IP addresses or resources, allowing them to carry out malicious activities without detection. This highlights the critical need to protect the DNS infrastructure to maintain the integrity and security of network communications.
DNS has a history of being attacked or used in attacks. The most common and oldest attacks on DNS are DoS attacks and DNS hijacking, as we see in [2]. The DoS (Denial of Service) attack is a method used to flood a machine with external communication requests. DNS hijacking or DNS redirection is a method that changes the answers to DNS queries. These attacks pose significant threats to the stability and security of the DNS infrastructure[1], requiring continuous efforts to mitigate and prevent their impact.
DNS tunneling[9, 10, 11, 12] also known as DNS exfiltration or covert channel, is a technique that encapsulates data within DNS messages and allows covert communication between two endpoints. It bypasses traditional security measures like firewalls, in which attackers use a way to extract sensitive data. However, we can use it for legitimate use cases where attackers can extract sensitive data, e.g. circumventing censorship. The detection of DNS tunneling is challenging because it looks like regular DNS traffic, and we have to analyze it to find any unusual patterns carefully, as we see in subsection 4.4. Understanding and detecting DNS tunneling for safeguarding networks from exploitation.
Until now, most research has concentrated on detecting DNS tunneling on local networks. We aim to look at DNS traffic at a ccTLD (country code top-level domain) level. This way, we hope to learn more about how DNS tunneling is used worldwide on a larger scale.
The primary objective of this study is to explore DNS tunneling techniques and tools[23] within .nl traffic. This research includes some specific research questions we need to resolve on .nl traffic. The research questions are below:

R1 Which DNS tunneling techniques are used, and what are the distinct features of the DNS channels over the network? How can we use these features to detect them on a TLD (.nl)?

- Which distinct footprint leaves DNS tunneling techniques in network traffic?
- How can we use the previously identified footprints to identify DNS tunneling in DNS traffic captured at the authoritative name servers?

R2 What are the current State-of-the-Art rules on DNS tunneling detection techniques?

R3 How do we transform the detection techniques to apply to higher levels in the DNS hierarchy?

R4 Does this transformation result in rules that detect DNS tunneling while rejecting benign traffic?

R5 What are the characteristics of the identified DNS tunneling attempts? For example, in which countries and networks are DNS tunneling performed?

The .nl traffic is captured at authoritative name servers that are managed by SIDN[1], the DNS operator responsible for the ccTLD (country code Top-Level Domain) of the Netherlands. SIDN stores the registration and manages all the domain names within .nl traffic. Additionally, SIDN provides access to ENTRADA[32], which is a tool that supplies daily DNS data of .nl traffic and enables us to conduct experiments for our research.
The initial phase of our investigation involves a literature review to identify and analyse features, implementation methods, and detection mechanisms of DNS tunneling techniques and tools. In this review, we aim to gain insight into how these techniques manifest within DNS traffic and to understand their behaviours as we see in section 2.3. Subsequently, we implement a Unix-based testbed environment that represents real-world DNS traffic scenarios. This environment includes essential components, including the client, DNS resolver, and authoritative name servers, as described in section 3. By applying DNS tunneling techniques and tools identified through our literature review, we simulate their utilization within this controlled environment in section 3.1. Thus, we can observe and analyze the behaviour of DNS tunneling techniques and tools in a controlled setting, providing valuable insights into their features and characteristics, as we see in section 4.
Furthermore, our research involves detecting DNS tunneling techniques and tools by analysing both the testbed environment and .nl traffic in section 4. Our approach includes implementing detection rules to identify potential DNS tunneling queries within the testbed environment, as seen in section 4.4. Additionally, we employ the ENTRADA tool[32] to analyze .nl traffic behaviour and validate DNS tunneling queries, as we see in section 5.

2 Background

2.1 DNS Protocol

The Domain Name System (DNS) is a fundamental protocol used on the Internet to organize and translate between readable domain names and IP addresses by helping to locate devices, networks and services online.
DNS manages a hierarchical and well-distributed structure called the DNS tree, which includes delegating and mapping domain names on authoritative name servers for registering Internet domain names, as shown in Figure 1. These servers store DNS records[1] specific to domain names and respond to user queries. Figure 1 presents the DNS hierarchy[1] below:

[1]SIDN company website `https://www.sidn.nl/`

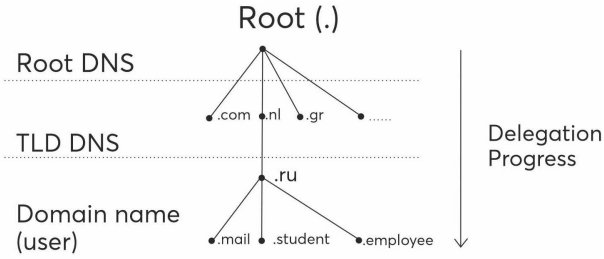


Figure 1: Example DNS hierarchy

DNS records are also known as Resource Records. The Start Of Authority (SOA) records information about a domain or zone, when the domain was last updated, and how long the server should wait between refreshes. Mapping names to IP addresses, there are two types of IP addresses: version 4 and version 6 (A and AAAA). SMTP mail exchanger (MX) records indicate how email messages should be routed. Name server (NS) records indicate which DNS server is authoritative for that domain, and pointer records for reverse DNS lookups (PTR) map IP addresses back to domains. Finally, the CNAME (canonical name) record is used in place of an A record when a domain or subdomain is an alias of another domain.
Historically, DNS was primarily implemented over User Datagram Protocol(UDP). With the evolution of newer technologies such as DNSSec[5] and DoTLS[6], Transmission Control Protocol[4] (TCP) usage has increased. However, DNS tunneling typically operates over UDP. The following subsection describes DNS tunneling in more detail.

2.2 DNS Tunneling

DNS Tunneling operates by embedding data within the DNS queries and responses, as seen in 2.2.1. The covert communication method utilizes the transmission of information between a server and a client while evading traditional security measures.
One specific feature of DNS tunneling is the ability to bypass network protection such as firewalls, making it attractive for attackers seeking to exfiltrate sensitive data from compromised systems. In most networks, DNS traffic is allowed to pass firewalls, making it appealing for malicious actors to encapsulate information[10].
However, it is important to mention that DNS tunneling is not only used for malicious purposes. In some cases, legitimate users may operate to overcome network restrictions or transmit data in environments where other communication channels are restricted.
The detection of DNS tunneling can be challenging. Since DNS is a fundamental protocol for Internet communication, distinguishing between legitimate DNS traffic and covert DNS tunneling traffic requires further analysis. It includes examining patterns, anomalies and unusual behaviour within the DNS traffic to identify potential instances of tunneling.

2.2.1 DNS Tunneling Tools

DNS tunneling tools[23] facilitate the establishment of DNS tunnels, leveraging DNS queries and encapsulating data within DNS messages to utilise controlled communication. These tools typically operate in two parts: client and server.
The client part is installed on the machine the user or attacker works on. It acts as a recursive DNS resolver. This behaviour plays a crucial role in DNS tunneling[9, 10, 11, 12, 13] on each level of the DNS hierarchy. The server part resides on the Command and Control (C2) server, which is external to the network. The server part masquerades as the authoritative name server of the controlled domain.

The client must "search" for the server through the DNS hierarchy. For this, it usually sends queries through the legitimate local recursive DNS resolver.
Despite adhering to the same fundamental principles, the implementation methods of various DNS tunneling tools may vary. Classification of DNS tunneling utilized for data encapsulation distinguishes DNS tunneling tools into main categories: UDP over DNS tunnels and TCP over DNS tunnels[16]. Section 4 provides detailed insights into the functionality and operation of specific DNS tunneling techniques[9, 10, 11, 12, 13, 14]. In summary, DNS tunneling represents a dual communication[8] that poses security risks. Understanding the operation and detection methods for securing networks against potential exploitation.

2.3 State-of-the-Art Rules on DNS Tunneling Detection

We examine existing research[9, 11, 12, 13, 14] to determine the State-of-the-Art rules are the detection methods of DNS tunneling. This analysis uses previous studies referencing the same detection methods in existing research described below.

- **Payload analysis:** This method involves examining the payload information of DNS packets. By analyzing the content carried within these packets and, features specific to DNS tunneling techniques can be identified.
- **Traffic Analysis:** Traffic analysis provides an alternative perspective by analyzing the overall DNS traffic over a while. This approach looks at patterns and features of DNS traffic on a global scale to detect effective features indicating DNS tunneling.

Payload Analysis:

1. **Size of request**: DNS tunneling requests may contain unusually long labels of up to 63 characters and overall domain names of up to 255 characters.
2. **Entropy of hostnames**: Legitimate domain names typically consist of recognizable words or meaningful phrases, while encoded names include higher entropy and less using predictable characters. However, there are exceptions, such as domain names used by content delivery networks.
3. **Uncommon Records Types**: Detection methods can involve identifying resource records not commonly used by regular clients, such as "TXT" records.
4. **Policy Violation**: Monitoring for DNS requests sent directly to the internet bypasses internal DNS resolvers and can signal a violation of network policies. However, most DNS tunneling utilities are designed to evade such detection by routing requests through internal resolvers.
5. **Specific Signatures**: Researchers may develop signatures to detect specific DNS tunneling utilities by checking for unique attributes in DNS headers and payload content, such as Sender Policy Framework (SPF) signatures.
6. **Statistical Analysis**: The Detection of DNS tunneling involves examining domain names for certain traits. Legitimate names usually have fewer numbers, while encoded ones have more. Longer sequences of numbers and a larger percentage of the Longest Meaningful Substring (LMS) length may indicate potential tunneling. It's advised to watch for domain names with over 27 unique characters. Analyzing character frequencies helps identify unusual patterns, like repeated consonants or uncommon mixes of numbers and consonants, which could signal tunneling attempts.

Traffic Analysis:

1. **Volume of DNS traffic per IP address**: This method involves monitoring the amount of DNS traffic generated by a specific client IP address. DNS tunneling typically involves limited data per request(up to 512 bytes) and results in multiple requests for communication. In addition, the continuous DNS requests[1] from the client can also indicate tunneling activity.

2. **Volume of DNS traffic per domain**: DNS tunneling tools[23] are configured to tunnel data using specific domain names that lead to increased traffic directed to those domains. However, using multiple domain names can decrease traffic per domain.

3. **Number of hostnames per domain**: DNS tunneling utilities often request unique hostnames with each request, leading to a higher number of hostnames compared to legitimate domain names.

4. **Geographical location of DNS server**: Large volumes of DNS traffic originating from regions where a business does not operate may indicate suspicious activity. This method is particularly useful for enterprises with limited international presence.

5. **Domain History**: Security experts examine the age of DNS records such as "A" or "NS" records to identify the domain names involved in malicious activity that include possible DNS tunneling. Recently acquired domain names with recent record additions may raise suspicion.

6. **Volume of NXDOMAIN responses**: The detection of NXDOMAIN responses indicates non-existent domain names can help to identify DNS tunneling tools like Heyoka[26] that generate numerous such responses. The Heyoka DNS tunneling tool generates a diverse array of responses because crafted to blend seamlessly within DNS traffic, facilitating covert communication and bypassing network restrictions with heightened effectiveness.

7. **Visualization**: Analyzing DNS traffic can help detect DNS tunnels. Tunneled traffic may be distinguished in visualizations and require an expert to perform an interactive analysis.

8. **Orphan DNS requests**: In normal computing, DNS requests typically prelude other requests(e.g webpage requests). Orphan DNS requests occur without subsequent requests and may indicate covert communication.

9. **General covert channel detection**: Security analysts can use specialized detection tools to examine factors such as request timing or compare traffic patterns to statistical norms to detect DNS tunneling.

The State-of-the-Art Rules mentioned in the literature review have been incorporated into the DNS testbed environment constructed for this research, as described in section 3. Detailed descriptions of the implementation of these rules are provided in sections 4 and 5. The effectiveness of these rules needs to be tested in the real world, as mentioned in [12].

3 DNS Testbed

The simulation of DNS tunneling on a DNS testbed provides us with a controlled environment where we observe the behaviour of DNS tunneling. Our DNS testbed

is based on a virtual environment created on VirtualBox[15] to simulate how DNS tunneling looks from the perspective of the .nl authoritative name servers. A DNS testbed is crucial because it allows us to install and implement DNS tunneling tools in a controlled and isolated environment, enabling us to study and understand their behaviour by observing the DNS queries and responses of DNS tunneling. Subsequently, The following Figure 2 below provides an overview of the DNS testbed environment:

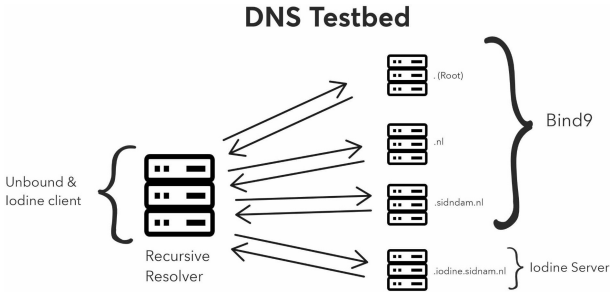


Figure 2: DNS Testbed

The DNS setup consists of five machines connected to a NAT[7] network. These machines utilize Unix-based operating systems, including Ubuntu Server and Ubuntu Desktop. Three of the virtual machines utilize BIND9[17] suite software that is utilized to configure the authoritative name servers for the root domain **"."**, the **".nl"** domain, and the custom domain **".sidndam.nl"**. Each of the authoritative name servers delegates to the next authoritative name server, as in the real-world DNS delegation. The fourth machine runs Unbound software[18] that serves as a Recursive resolver (DNS resolver), and we configured it with custom settings tailored to the research requirements. This machine will also run an Iodine client. The final machine runs the Ubuntu Desktop that we dedicated to installing a DNS tunneling tool[23] that runs the server-side of an Iodine DNS tunnel. Further details regarding the Iodine setup are provided in the subsequent section 3.1.

3.1 Iodine DNS Tunneling Tool

Iodine[23] is a DNS tunneling tool available for various platforms. It supports a wide range of resource records and encoding techniques. This tool establishes a tunnel through DNS by creating interfaces on the server and client to enable communication, as described in more detail in subsection 2.3. On the server side, the Iodine DNS tunneling tool initializes as a daemon service by acting as the tunnel server and remains on standby to receive DNS requests[1] from the client. The client side initiates communication with the server by sending an encoded DNS request in the hierarchy of authoritative name servers until it reaches the controlled authoritative name servers. The tunnel's server can only decode these requests, which offers secure communication.

Iodine Setup

The installation and setup of the Iodine DNS tunneling tool[23] involves several steps. This machine runs an operation system (Ubuntu Desktop) and is used on the Iodine server side, the source of DNS tunneling. The installation and configuration process of Iodine server-side is described in more detail in Appendix 1 Part A.
In our case, the Iodine DNS tunneling tool requires a client-side installation based on a targeted authoritative name server, which is the custom domain name **".sidndam.nl"**.

We configure the authoritative name server to delegate to the Iodine server using an additional and custom label **".iodine.sidndam.nl"**. The running of the Iodine DNS tunneling tool is described in more detail in Appendix 1 Part B.
In conclusion, the above steps establish the DNS tunneling communication between the server-side and client-side of the Iodine tool[23]. This setup allows us to analyze DNS tunneling queries within our DNS testbed. In the next section, we examine methods for detecting the DNS tunneling techniques.

3.2 Supporting Tools

The DNS testbed environment is a small-scale representation of the real-world Internet network. Focusing on .nl traffic that enables the execution of the targeted experiments. The software to achieve this is the following:

3.2.1 Tcpdump

The Tcpdump[19] is a command-line packet sniffer that captures the network data of the DNS network traffic for display. It captures the machine traffic, which acts as an authoritative name server.

3.2.2 Wireshark

The Wireshark[20] is a network protocol analyzer. We use it to capture the DNS network traffic. This software allows the user to capture the DNS network traffic into PCAP[2] files for packet analysis.

3.2.3 SSH (Secure Shell)

The Secure Shell[21] protocol is a network protocol that is used to control a remote server/system securely. SSH[21] transfers the data in an encrypted tunnel between the host and client. Additionally, we utilized this network protocol by sending "TXT" files between the server and client on the DNS testbed to observe the impact during the application of DNS tunneling. We use "TXT" files, which are simple format files that input text.

3.2.4 SCP (Secure Copy)

The Secure Copy Protocol (SCP)[22] is a protocol and command-line tool that allows the user to transfer and copy files and directories between two locations. The research used this tool to securely transfer custom files from the DNS tunneling server to the client side. The difference is that SSH provides secure remote access and command execution.

4 Detection process of DNS Tunneling

The process of detecting DNS tunneling involves two main phases.

- First, we use the DNS testbed environment to simulate DNS tunneling techniques. We utilize network tools to capture DNS network data that can be used to identify the features of DNS tunneling traffic. These data come from various sources, including the **"."(root)** authoritative name server, the **".nl"** authoritative name

[2]PCAP files contain a recording of the packet data of a network.

server, the daemon Iodine[23] server and client and the DNS resolver utilizing DNS tunneling tools.

- We detect DNS tunnelling techniques within .nl traffic in the second phase using the ENTRADA tool[32] explained in 4.1 and 4.2. This tool accesses DNS queries sent by recursive DNS resolvers to the authoritative name servers of the .nl ccTLD.

These two phases include further analysis, which entails running scripts that follow defined rules to detect potential DNS tunneling techniques and tools, as described in section 4.4.

4.1 ENTRADA

The ENTRADA[32](ENhanced Top-level domain Resilience through Advanced Data Analysis) tool captures daily DNS traffic of .nl authoritative name servers, storing data for almost a year. In addition, ENTRADA provides the capabilities to the user to investigate specific datasets of .nl traffic by defining the queries in the tool's database that we need for custom detection rules of DNS tunneling. The data we received from ENTRADA databases is in human-readable format DNS queries, and this feature provides the capability to process them.
ENTRADA is utilized to analyze DNS queries within .nl traffic. Accomplishing this involves transforming the DNS queries captured as PCAP files at the authoritative name servers to a more efficient column data format, facilitating a more straightforward analysis. Additionally, ENTRADA employs an analytical query engine that includes Hadoop[24], a storage framework which stores information in PCAP files that can be queried with Impala[25]. This processed data is directly available for processing the authoritative name servers of .nl traffic.

4.2 Prefiltered data on ENTRADA

There is a technical limitation on how much data the ENTRADA[32] tool can return. Additionally, a significant portion of the data in ENTRADA is irrelevant to our custom detection rules. Therefore, we limit the data we extract from ENTRADA to run our custom rules on, and we refer to this as "prefiltered data."
Prefiltered data in the context of ENTRADA refers to DNS query data that has been selectively extracted based on specific criteria to focus on potentially suspicious queries indicative of DNS tunneling. We can more effectively identify potential and suspicious DNS tunneling queries by querying ENTRADA with custom rules. These rules are applied through custom SQL queries within the ENTRADA tool, which extract DNS data from databases. The extraction of these DNS data provides specific DNS queries, which we apply to custom detection rules as described in subsection 4.4. This process also aims to filter out unwanted traffic irrelevant to our custom detection rules, ensuring that our analysis targets potentially harmful activities and eliminates noise from benign DNS traffic.
Our SQL queries include parameters that ask for specific information from ENTRADA. These parameters are described in the following list:

- Length of DNS queries.
- Query Types of DNS queries.
- Specific Signatures of DNS queries.
- Specific Dates based on Day, Month and Year.

- Specific label parts of DNS queries are the leftmost labels in our case.
- Regex filters for numbers and characters allowed on DNS queries.
- Regex filters for specific names, as we see below.
- Limitation of the amount of DNS data we request from ENTRADA[32] database.

The parameters above extract specific formats of DNS queries from ENTRADA. We aim to extract these DNS queries, which are used in custom detection rules as described in subsections 4.4 and 5.3, based on the potential features of DNS tunneling queries[9, 11, 12]. These parameters allow us to filter out irrelevant data and focus on DNS queries that exhibit characteristics of DNS tunneling. By applying custom detection rules to these specific DNS queries, we can more effectively detect features indicative of DNS tunneling.
Furthermore, we implement regex filters to exclude certain recurring names from our prefiltered data on ENTRADA. Excluding these names provides a clearer overview because these DNS queries are associated with online services that we do not consider threats. These names, including "_domainkey"[33], "aws"[34], "azure"[35], and "thissubdomainshouldonlyresolveifwildcard," are identified through routine monitoring of .nl traffic. The DNS queries associated with these names contain characteristics typical of DNS tunneling, such as high entropy, specific encoding, and uncommon query types. The validity that these recurring names are not threats and are not included in DNS tunneling is based on our observations and knowledge of legitimate services, as detailed in[33, 34, 35]. Therefore, we determined these names as non-indicative of DNS tunneling activity and omitted them from our analysis.
This approach is informed by our continuous monitoring of .nl authoritative name server traffic and our commitment to refining our research methodology to focus on relevant DNS query patterns indicative of potential DNS tunneling techniques.

4.3 Transformation of State-of-the-Art Rules to Custom Rules

In subsection 2.3, we explored existing literature to gain information on various methods for detecting DNS tunneling techniques and tools[23], referred to as State-of-the-Art rules, found in studies[9, 10, 12, 13, 14]. We implemented these state-of-the-art rules one by one in our DNS testbed and .nl DNS traffic to gain insights into the behaviour of DNS tunneling techniques and tools on actual DNS data.
However, the state-of-the-art rules need modification because the approaches and implementation equipment differ. In particular, the absence of DNS responses in the ENTRADA dataset and using QNAME Minimization[31] present significant differences. The absence of DNS responses means we only have access to DNS queries, and QNAME Minimization limits the amount of data visible in DNS queries.
Due to these differences, we develop custom rules based on adaptations of the State-of-the-Art rules tailored to the DNS queries dataset. This approach allows us to address the limitations of our dataset and effectively apply DNS tunneling detection techniques to the DNS queries we have available. By adapting the rules to fit our specific constraints, we ensure that our detection methods are practical and relevant to the dataset we are analyzing.

4.4 Detection Rulesets on DNS Testbed and ENTRADA tool

In subsection 2.3, we apply state-of-the-art detection techniques outlined in previous literature reviews[9, 11, 12, 13, 14] within the DNS testbed environment, focusing on .nl traffic. This dual application provides an overview of the effectiveness of these techniques

in both controlled and real-world settings. To achieve this, we create predefined rulesets based on insights from the DNS testbed, tailored specifically for detecting DNS tunneling techniques in DNS queries. The development of these rules is informed by observations made within the DNS testbed environment. The rules we applied are as follows:

- **Rule 1**: The implementation of the Shannon Entropy[27] to calculate the entropy of query names per label by setting a threshold of 3.8 based on observations within the testbed environment. If the calculated entropy surpasses this threshold, the rule provides True; if it fails below the threshold, it returns False. The definition of the threshold is informed by insights based on the DNS testbed showing that DNS queries related to DNS tunneling had an entropy higher than 3.8. Additionally, this rule corresponds to the Payload Analysis rule 1, "size of Request" from subsection 2.3.

- **Rule 2**: The detection of "Base32"[28], "Base64"[28], "Hex"[28] and "NetBIOS"[29] encoding by examining the subdomain of each query. The detection of these encoding types is an indicator that exists as one of the features of DNS tunneling in DNS queries, as we see in the Payload Analysis rule 2, "Entropy of hostnames", from subsection 2.3. This rule identifies encoded labels and determines the length of the encoded labels within each subdomain of the DNS query. Our detection approach is informed by observations within the DNS testbed, in which we observe the behaviour of DNS tunneling queries. Subsequently, the documentation of the Iodine DNS tunneling tool[23] described implementing these encoding types into the DNS traffic.

- **Rule 3**: This rule examines the length of the requested query against a defined threshold limit. If the length falls between 50 and 550 bytes, it is True; otherwise, it returns False. We apply this rule to DNS queries within our DNS testbed and the ENTRADA tool[32]. Our observation indicates that both environments typically adhere to this length limit, and our analysis focuses solely on queries, as mentioned in section 4.4.1. Subsequently, the implementation of this rule is based on the Traffic Analysis rule 1, "Volume of DNS traffic per IP address", from subsection 2.3.

- **Rule 4**: The identification of uncommon resource records within each query such as "TXT", "NULL", and "PRIVATE". Their types are reported if these resource records are presented in a DNS query. Otherwise, these are labelled as "Unknown". This rule is established based on State-of-the-Art rules, observations from the DNS testbed and insights from the documentation of the Iodine DNS tunneling tool. Furthermore, Rule 4 follows the Payload Analysis rule 3, "Uncommon Record Types", from subsection 2.3.

- **Rule 5**: In this case, we identify continuous sequences of characters and numbers such as aaa, bbbb, 000, 22222, etc. If continuous sequences are detected, the result is "True"; otherwise, it returns "False". This rule is defined from observations made within the DNS testbed during the implementation of the DNS tunnelling tool.

- **Rule 6**: The identification of the characters "z" or "y" in the first letter and leftmost label of a subdomain. If these characters are detected, the result is "True"; otherwise, it returns "False". This rule is derived from observations conducted during numerous tests within the DNS testbed, along with insights from the documentation of the Iodine DNS tunneling tool[23].

- **Rule 7**: The utilization of the fuzzywuzzy[3] library, utilizing a threshold set at 91 to detect similar subdomains. This Library leverages the Levenshtein[4] distance[30] metric, which quantifies the similarity of two strings. A higher Levenshtein distance indicates greater dissimilarity between the strings. We utilize this rule based on observations within the DNS testbed when we execute the DNS tunneling, which includes instances of similar and encoded DNS queries that present minor similarities, such as differences in one or more characters or numbers.

- **Rule 8**: This rule enables the detection of error types present in each DNS query. The error types we detect are NXDOMAIN, NoNameServer, Timeout and NoAnswer. Identifying the NXDOMAIN error type is crucial for our research because it helps uncover potential DNS tunneling queries based on observations of the Iodine in the DNS testbed. We implement this rule based on established guidelines from the State-of-the-Art rules and observations made on DNS queries within the DNS testbed. Finally, this rule is based on the Traffic Analysis rule 6, "Volume of NXDOMAIN responses", from subsection 2.3.

Based on our observations, the rules mentioned above are implemented individually, yielding massive data. The Rules 5 and 6 are based on observations not present in previous studies. Section 5 uses combinations of these rules that offer meaningful analysis and efficient performance in processing queries. These rule combinations aim to summarize the information obtained, resulting in more focused insights and facilitating the establishment of a reliable scoring system to validate DNS tunneling queries.

5 Validation

Detecting DNS tunneling techniques[9, 11, 12, 13, 14] relies on a scoring system based on the rules from subsection 4.4. This process is divided into two main parts. First, we validate the functionality of the DNS testbed by operating DNS tunneling tools[23]. Second, we utilize the ENTRADA tool[32] to analyze .nl traffic. We employ the rules mentioned above in both phases to detect DNS tunneling occurrences. Finally, we develop a scoring system based on the outcomes of these detection rules, as detailed in subsection 5.3.

5.1 Validation custom rules on DNS testbed

We establish a scoring system where the domain names with the highest similarity with DNS tunneling queries are assigned the highest scores. These scores determine the likelihood of DNS queries representing DNS tunneling [9, 10, 11, 12, 13, 14]. This scoring system is informed by the findings from our literature analysis and observations made within the DNS testbed.
As an example, Table 1 shows a single DNS query observed on the DNS testbed by utilising the Iodine DNS tunneling tool[23]:

[3]Fuzzy string matching like a boss. It uses the Levenshtein Distance to calculate the differences between sequences in a simple-to-use package.

[4]The Levenshtein distance is a number that tells you how different two strings are. The higher the number, the more different the two strings are.

qname	Query Type	ERROR	ERRORTYPE
yrbm22.iodine.*.nl.	NULL	True	NXDOMAIN

Table 1: DNS tunneling query on DNS testbed

The score for this DNS query is 3 points:

- 1 point because leftmost label first letter "y".
- 1 point because NULL Resource Record(RR).
- 1 point because of NXDOMAIN.

This scoring system is explained in more detail in section 5.3.

5.2 Non-Implemented Rules

In subsection 4.4, we removed Rules 3 and 7 of the DNS testbed because .nl traffic does not contribute to our dataset efficiently.

5.2.1 Rule 3

We established a rule, Rule 3, to assess the size of DNS queries. This rule checks if a query's length falls within 50 to 550 bytes. If within this range, the rule returns True; otherwise, it returns False. However, we removed this rule from our rulesets because, despite its implementation, it consistently returns True for all DNS queries in the daily traffic using the ENTRADA tool[32]. Such behaviour occurs because the DNS queries in .nl traffic fall within limits defined by our rule.
Additionally, when we applied custom rules to detect potential DNS tunneling queries, this rule still appeared True. Upon further examination, we found that DNS queries generally contained less information than responses, as evidenced by our DNS testbed. In our study, we analyzed the DNS queries included in .nl traffic because we approach the detection of DNS tunneling only on DNS queries, as detailed in subsubsection 4.4.1. Therefore, we concluded that this rule was ineffective for our purposes and decided to remove it from our rulesets in section 5.2.

5.2.2 Rule 7

Rule 7, as discussed earlier, is used to identify similar subdomains in the queries. This rule employs the fuzzy-wuzzy library, which utilizes the Levenshtein distance[5][30] to compare subdomains. The Levenshtein distance is crucial for detecting similarity in our DNS test environment.
Detecting similar DNS queries helps us identify potential instances of DNS tunneling[9, 10, 11, 12, 13, 14], especially when implementing tools like Iodine DNS tunneling tool[23] in our DNS testbed. With a 91 similarity threshold, Rule 7 identifies subdomains with minor differences, typically involving 3 or 4 characters or numbers. However, this process is computationally intensive due to the need to compare all subdomains and determine similarity based on our threshold. We assign 2 points for each similar qname, reflecting our scoring system in the DNS testbed. Finally, Table 3 below provides an example illustrating the application of this rule:

[5]Levenshtein distance is a measure of the similarity between two strings, which takes into account the number of insertion, deletion and substitution operations needed to transform one string into the other.

Subdomain 1	Subdomain 2	qname
85c5f18c9d3ec7d1c7.*.*	85c7f13c9d3ec7d1c7.*.*	85c5f18c9d3ec7d1c7.*.*.**.nl.
85c5f18c9d3ec7d1c7.*.*	85c7f13c9d3ec7d1c7.*.*	85c5f18c9d3ec7d1c7.*.*.**.nl.

Table 2: Comparision of potential DNS tunneling query

Table 3 compares subdomains and their corresponding QNAMEs to illustrate potential DNS tunneling patterns. Each row highlights subdomains with specific character sequences, showing variations that could indicate tunneling activity. The QNAMEs combine these subdomains with additional domain segments ending with ".nl."
By not using this rule, we achieve faster results. Conversely, we may miss certain DNS tunneling characteristics present in .nl traffic.

5.3 Testing custom rules on .nl traffic

While evaluating the custom rulesets on .nl traffic, we devised a scoring system to prioritize suspicious DNS queries based on features associated with DNS tunneling techniques. This scoring approach calculates points for each DNS query, integrating insights from literature analysis and observations in our DNS testbed.
As explained in section 5.2, in our rulesets below, we excluded Rule 3, which set a threshold for DNS query lengths due to minimal impact on our .nl traffic dataset. Similarly, Rule 7, aimed at detecting similar subdomains, was omitted to mitigate extensive computational time.
This scoring system is structured by combining subsection 4.4 into the following three distinct rulesets described in Table 3 below:

First Ruleset:	• **Rule 1** assigns 1 point to each label of DNS query if entropy > 3.8, as high entropy suggests a higher likelihood of data encoding typically in DNS tunnelling. • **Rule 2** allocates 2 points for "Base32" and "Base64" detection and 1 point for "Hex" and "NetBIOS", reflecting their common usage in tunneling practices. • **Rule 4** gives 1 point for each detected record type "TXT", "NULL", and "PRIVATE", as their rarity and potential for carrying hidden data make them significant indicators.
Second Ruleset:	• **Rule 5** assigns 2 points for detected continuous patterns, given their strong indicator for techniques used by DNS tunneling. • **Rule 6** provides 1 point if "y" or "z" is the start of the leftmost label of a DNS query, as this feature appears to be an indicator of DNS tunneling tools and techniques.
Third Rule:	• **Rule 8** assigns 1 point for each detected NXDOMAIN error type, an error response often seen in DNS tunneling.

Table 3: Rulesets and Category of Rules

Each ruleset targets specific DNS tunneling characteristics, ensuring focused analysis without conflating different query types. This method facilitates identifying suspicious DNS activities by prioritizing queries showing multiple indicators across rulesets.
Combining all rulesets into one unified framework simplifies the structure but risks diluting the focus on specific tunneling characteristics. The way we apply Rulesets 1 and 2 on the same data implies that we could combine them into 1. However, we kept them grouped with only similar rules because that enhanced our ability to inspect the results visually. Thus gaining more insight into how they behave for purely automated analysis. The grouping into rulesets would not be required. Additionally, we highlight the utilization structure of these rulesets as we see in the following Figure 3:

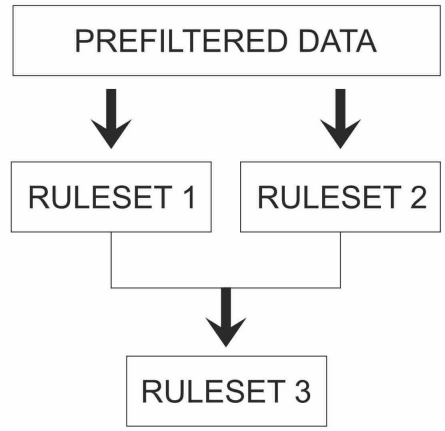


Figure 3: Utilization Structure of Rulesets

The scoring system assigns a score (ranging from 0 to X) to each DNS query based on its alignment with the defined rules. This approach consolidates results for in-depth analysis, providing a detailed overview of DNS queries exhibiting tunneling features outlined in Table 3.

5.4 Validation of custom rules on Public DNS tunneling server

We validate the custom rules of detecting DNS tunneling in .nl traffic.
The validation of these custom rules involves asking a genuine user to utilize the Iodine DNS tunneling tool[23] for a full day and to transfer data over DNS tunneling to observe and analyse DNS traffic from the .nl authoritative name servers. Additionally, the genuine user needs to disable the QNAME minimization[31] to capture all necessary information related to DNS tunneling queries. Disabling QNAME minimization allows us to see the complete DNS queries in .nl traffic, enabling us to observe the behaviour of the potential DNS tunneling queries efficiently.
Subsequently, we asked for the registered domain name and the activation date of DNS tunneling[9, 10, 11, 12, 13, 14] to search for the relative DNS data. In the prefiltered data, we defined the registered domain name and the specific date to the ENTRADA tool[32] to retrieve all related DNS data captured from the .nl authoritative name servers.
Next, we applied the custom rules to detect DNS tunneling queries in the captured .nl DNS data. The custom rules are applied to identify the DNS tunneling queries and are assigned the total score, as we see in section 5.1.
In the case of .nl traffic, we detect a DNS tunneling query that typically includes a "NULL" resource record, an NXDOMAIN error, a continuous pattern, and the leftmost

label's first letter beginning with the character "y". Such a DNS tunneling query is assigned a total score of 5 points. This query is presented in Table 4:

qname	ERROR	ERRORTYPE	Total Score	Query Type
yrbbb0.*.*.nl.	True	NXDOMAIN	5	NULL

Table 4: Validated DNS tunneling query

The Iodine DNS tunneling tool[23] initializes the communication and distributes the score to the validated DNS tunneling query as we describe below:

- NULL resource record is a simple format file that assigns 1 point.
- The leftmost label's first character, "y", assigns 1 point.
- The continuous patterns are assigned 2 points.
- The error type NXDOMAIN that is assigned 1 point.

In Table 4, we observe that both DNS testbed and .nl traffic have common points that present the exact behaviour of DNS tunneling queries as we see in section 5.1.
Finally, we validated the presence of DNS tunneling by combining detection rules applied to .nl traffic with confirmation from the actual DNS tunneling user about the validity of his source IP. This confirmation highlights the efficacy of our custom rules in accurately detecting DNS tunneling queries. Furthermore, similar DNS queries are presented on the DNS testbed in Table 1. We conclude that the validation includes the features of the literature analysis mentioned in the State-of-the-Art rules of the existence of DNS tunneling techniques[9, 10, 11, 12, 13, 14].

6 Measurements on .nl traffic

We examine the prevalence of DNS tunneling in .nl traffic and identify its users. We assessed its frequency and attributes over several months, as seen in section 6.2. Subsequently, we conducted a detailed analysis of suspicious traffic on a specific date to delve deeper into DNS tunneling, as mentioned in section 6.3. Finally, we focused on a single domain name that exhibited a high volume of suspicious DNS queries to explore further in subsection 6.4.
Our measurements primarily focus on determining the frequency of DNS queries originating from different countries, the types of queries and the unique IP addresses involved. Additionally, we evaluate the number of DNS queries associated with each second-level domain (domain name) for a specific date before and after implementing custom rules.

6.1 Results on .nl traffic

The final results on .nl traffic are compiled in a CSV file after applying the combination rulesets in section 5.2. This file summarises potential DNS tunneling queries along with specific details. The data presented in the CSV file include the query name (qname), the query type (Resource Recorcd), the error type and its existence, the autonomous system (ASN), the origin country, the source IP, the timestamp and the total score of each query. The following picture presents a tiny part of the CSV file:

qname	ERROR	Total Score	Query Type
y6tbwbiffzn52xyefrhcgv6ns.xi7cxcqtkfwpd6pm4cxq.*.*.nl.	False	14	TXT
aqyrybnsuih67lxpjqjcmgpxndypwira.ipg5namibkuyv7fbmscq.*.*.nl.	False	14	TXT
aqzrybnsuih67lxpjqjcmgpxndypwira.yhxcvgjuvqtbmnmfazdq.*.*.nl.	False	14	TXT

Table 5: Examples of potential DNS tunneling queries

DNS queries with a total score under 1 point are excluded from the file. These queries are not considered potential DNS tunneling queries because the custom rules did not detect the defined features of DNS tunneling techniques[9, 10, 11, 12, 13, 14]. The results of this CSV file evaluate some DNS queries based on the features of DNS tunneling techniques[9, 11, 12, 13, 14].

6.2 First Phase: Specific Dates Measurements

Initially, we measured the total DNS queries for six specific dates. We defined six specific dates: daily dates with a high workload. These dates are selected based on Tuesdays when we observe the highest number of employees in our company. These queries were categorized into total, prefiltered, and filtered data while implementing custom rules. Figure 4 represents the unfiltered data.

6.2.1 Top 6 countries

Figure 4 displays the top six countries where the DNS queries originated and the corresponding number of DNS queries within the total .nl traffic of specific dates. The United States (country code US) has the highest number of DNS queries in the unfiltered data for each date, showing a large portion of daily .nl traffic comes from the US. The Netherlands (cc NL), Germany (cc DE), Ireland (cc IE), and France (cc FR) also consistently appear with a similar number of queries on all dates. These countries consistently contribute to daily traffic, except on the last date when France (cc FR) was replaced by Sweden (cc SE).

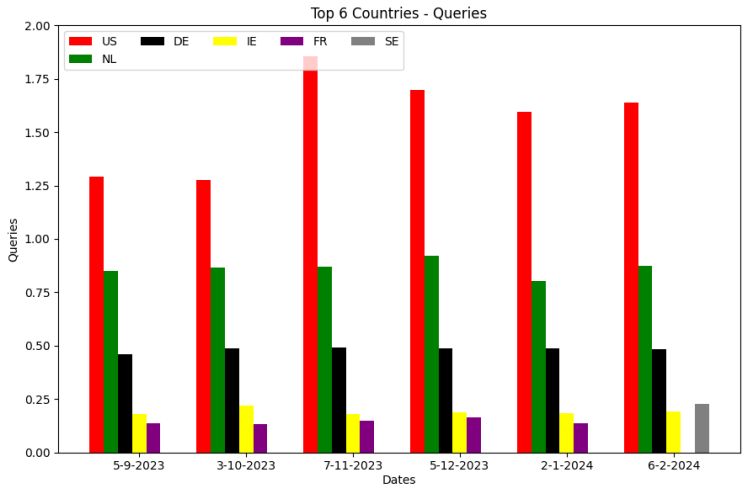
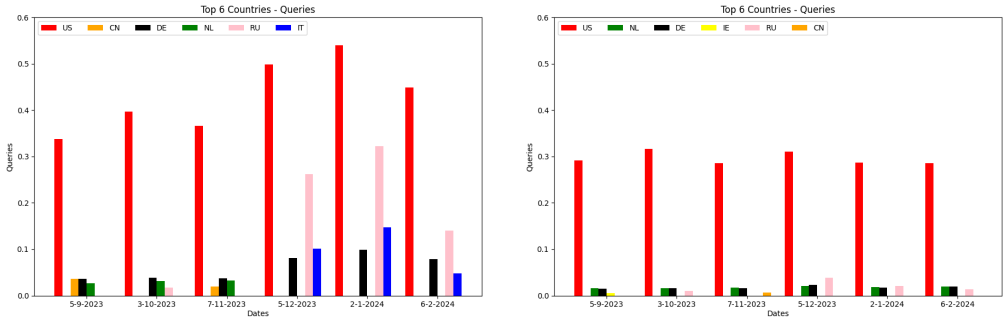


Figure 4: Total DNS queries of Top 6 countries of specific dates

In the following Figures, we compare the DNS queries before and after applying the custom rules designed to detect potential DNS tunneling[9, 10, 11, 12] across all specific dates.

Figures 5(a) and 5(b) show that the United States remains the top source of DNS queries in both prefiltered and filtered data. In Figure 5(a), Germany (cc DE) and the Netherlands (cc NL) are also among the top countries in prefiltered data, though their query counts drop sharply compared to Figure 4. Additionally, China (cc CN), Russia (cc RU), and Italy (cc IT) appear among the top six countries in prefiltered data in Figure 5(a), replacing Ireland (cc IE), France (cc FR) and Sweden (cc SE) from the top 6 in Figure 4. Notably, Russia's appearance on the last three dates suggests potential DNS tunneling.
In Figure 5(b), the Netherlands (cc NL) and Germany (cc DE) are still present, as seen in Figures 4 and 5(a), but with fewer queries in the filtered data. Russia (cc RU) and China (cc CN) also appear in the filtered data, with Russia consistently showing up on the last three dates, similar to Figure 5(a). Ireland (cc IE) is also present in the filtered data in Figure 5(b).



(a) Prefiltered Top 6 countries

(b) Filtered Top 6 countries

Figure 5: Top 6 countries before and after applying custom rules

The figures show that the United States (cc US), the Netherlands (cc NL), and Germany (cc DE) consistently appear in all datasets, forming the core .nl traffic and potentially including DNS tunneling queries.
The DNS query ratios across various countries and dates shed light on how much of their total traffic is suspected to involve DNS tunneling. These ratios represent the percentage of traffic identified as potentially DNS tunneling out of the total DNS traffic for each country. The difference between prefiltered and filtered data provides insight into how effectively filtering rules reduce the dataset.

Country	Date	Prefiltered Ratio (Figure 5(a))	Filtered Ratio (Figure 5(b))	Filtering-to-Prefiltering Ratio
United States	5-9-2023	0.026%	0.023%	88.4%
	3-10-2023	0.031%	0.025%	80.6%
Netherlands	5-9-2023	0.003%	0.002%	66.6%
	3-10-2023	0.004%	0.002%	50%
Germany	5-9-2023	0.008%	0.008%	100%
	3-10-2023	0.008%	0.008%	100%
Russia	5-12-2023	0.23%	0.03%	14.83%
	2-1-2024	0.26%	0.016%	6.26%
	6-2-2024	0.15%	0.014%	9.53%
China	5-12-2023	0.03%	0.003%	10%

Table 6: DNS Query Ratios of Selected Countries

On 5-9-2023, the United States showed that only 0.026% of total DNS traffic was flagged as suspected DNS tunneling in the prefiltered dataset, which decreased slightly to 0.023% after filtering. These findings indicate that a slight amount of United States traffic is suspected to involve DNS tunneling, and the filtering process removed some of that suspected traffic. By 3-10-2023, the prefiltered traffic ratio had increased to 0.031%, with the filtered traffic ratio at 0.025%, indicating that, although the ratio of prefiltered traffic increased, the filtered traffic stayed the same.

For the Netherlands, the prefiltered data on 5-9-2023 was 0.003% of the total traffic, dropping to 0.002% after filtering. The data suggests that only a slight fraction of the Netherlands' traffic is suspected of DNS tunneling. However, this also indicates that our filtering rules reduce the dataset by one-third and can be expressed as a ratio of how much filtered traffic remains from the prefiltered traffic, the "filtering-to-prefiltering ratio," which is 66%. By 3-10-2023, the prefiltered traffic had risen slightly to 0.004%. At the same time, the filtered data remained the same at 0.002%, indicating that, although the ratio of prefiltered traffic increased, the ratio of filtered traffic stayed the same.

In Germany, the prefiltered and filtered data ratios remained steady at 0.008% on 5-9-2023 and 3-10-2023, suggesting that the amount of traffic suspected of being DNS tunneling in Germany did not fluctuate and that filtering did not significantly affect the dataset.

For Russia, however, there were more significant changes. On 5-12-2023, 0.23% of total traffic was suspected of DNS tunneling in the prefiltered dataset, but this dropped sharply to 0.03% after filtering, showing that the filtering rules were very effective in reducing the suspected tunneling traffic, with a filtering-to-prefiltering ratio of 14.83%. By 6-2-2024, the prefiltered traffic ratio had increased slightly to 0.26%, but the filtered traffic dropped further to 0.016%. The resulting filtering-to-prefiltering ratio of 6.26% indicates differences in the traffic, which may suggest less tunneling activity or that tunneling still occurs but lacks the characteristics detectable by the filtering rules. On 6-2-2024, the prefiltered ratio decreased to 0.15%, while the filtered ratio was 0.014%, with a combined ratio of 9.53%, indicating a further reduction in suspected DNS tunneling traffic after applying the filtering rules.

For China, on 5-12-2023, the prefiltered data ratio was 0.03%, with a filtered data ratio of 0.003%. As a result, a filtering-to-prefiltering ratio of 10%, suggesting that a significant portion of the suspected traffic was effectively filtered out. This figure highlights the level of suspected DNS tunneling activity in China and demonstrates the effectiveness of the filtering rules in reducing that traffic.

These observations highlight how much suspected DNS tunneling traffic contributes to the total DNS traffic for each country. Additionally, Table 6 above presents all the percentages mentioned above. Countries like the United States and Germany show a consistent and small proportion of DNS tunneling traffic. In contrast, countries like Russia exhibit more pronounced fluctuations, suggesting potential changes in traffic behaviour. The effectiveness of the filtering rules is particularly noticeable in Russia's case, where a significant portion of suspected traffic is removed after filtering. This analysis underscores the importance of examining prefiltered and filtered data to assess DNS tunneling threats.
Table 6 above shows the case of Russia; the presence of the country in the prefiltered dataset but a significant reduction in the filtered data highlights the efficiency of the detection mechanisms. The fluctuations suggest that potential DNS tunneling activity might be occurring, and further investigation into the nature of this traffic could provide more insights.
In conclusion, by comparing prefiltered and filtered data, we can effectively assess the contribution of suspected DNS tunneling to each country's total traffic and understand the impact of filtering rules in mitigating potential security threats.

6.2.2 Top Query Types

Next, we measured the types of queries within the .nl traffic on the same six specific dates. Figure 6 shows that "A" records are the most common query type across all these dates, making up the largest portion of .nl traffic. The next most frequent query types are "NS", "AAAA", "DS", "MX", and "TXT", which also reflect typical .nl traffic patterns.

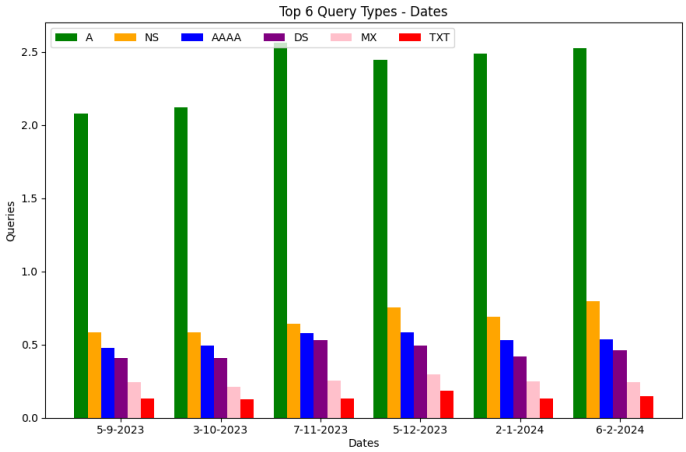
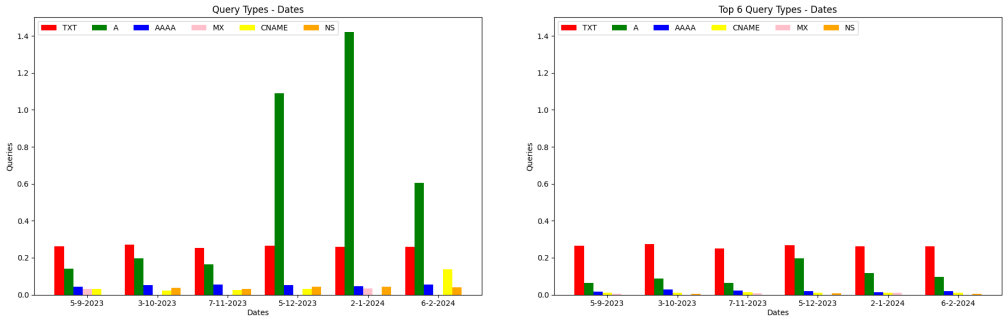


Figure 6: Total Query Types of specific dates

Figure 7 shows the prefiltered DNS queries and the DNS queries with custom rules applied.
In Figure 7(a), "TXT" records remain consistent across all dates. Figure 7(b) shows that during the custom rule implementation, "TXT" records consistently represent the highest traffic on all dates. Additionally, "A" records are the second most common query type in both figures, with a notable increase on the last three dates. Both figures also show significant query counts for "AAAA", "MX", "CNAME", and "NS" records.



(a) Prefiltered DNS query types

(b) DNS query types of custom rules

Figure 7: DNS query types before and after applying custom rules

We conclude that "TXT" records are the most commonly used potential DNS tunneling queries, making up almost one-fifth of DNS queries after applying rules on these dates, as shown in Figure 7(b).

Date	**Query Type**	**Prefiltered Ratio (Figure 7(a))**	**Filtered Ratio (Figure 7(b))**	**Filtering-to-Prefiltering Ratio**
5-9-2023	TXT	0.203%	0.203%	100%
5-9-2023	A	0.008%	0.003%	35.7%
5-9-2023	AAAA	0.060%	0.034%	50%
5-9-2023	NS	0.0001%	0.0001%	100%
3-10-2023	TXT	0.203%	0.215%	99%
3-10-2023	A	0.0084%	0.004%	28.5%
3-10-2023	AAAA	0.006%	0.002%	50%
3-10-2023	NS	0.001%	0.001%	100%
5-12-2023	A	0.04%	0.008%	18.2%
2-1-2024	A	0.0005%	0.00006%	10.7%
6-2-2024	A	0.00024%	0.00004%	16.6%
5-12-2023	TXT	0.145%	0.143%	99%
2-1-2024	TXT	0.200%	0.200%	100%
6-2-2024	TXT	0.179%	0.179%	100%
5-12-2024 to 6-2-2024	Other types	(similar as previous data)	N/A	N/A

Table 7: DNS Query Ratios on Different Dates

On 5-9-2023, the analysis of prefiltered data indicated that 0.203% of the total traffic for TXT queries, 0.008% for "A" queries, 0.06% for "AAAA" queries, and 0.0001% for "NS" queries were suspected to be DNS tunneling. Table 7 shows an overview of the ratios for the four most important query types on different dates. These ratios represent the proportion of total traffic for each query type believed to contribute to DNS tunneling activity, highlighting the extent of suspected tunneling in the overall DNS traffic for that date. To provide perspective, the total amount of prefiltered data on this date accounted for only 0.026% of the total traffic, while the filtered data made up just 0.023%. This shows that only a very small portion of the total DNS traffic was suspected to be tunneling.
After applying filtering rules, the proportions changed. For example, the "A" query

type decreased to 0.003%, indicating that only 35.7% of the prefiltered data remained after filtering. Similarly, the "AAAA" query type fell to 0.034%, retaining 50% of the prefiltered data. In contrast, the "TXT" query remained constant at 0.203%, indicating that nearly all the prefiltered data for this query type persisted after filtering. The NS query ratio remained the same at 0.0001%, suggesting variability in its response to filtering.
By 3-10-2023, the prefiltered data for the "A" query type showed a ratio of 0.0084%, which dropped to 0.004% after filtering, meaning that only 28.5% of the prefiltered suspected tunneling traffic remained. For "AAAA" queries, the prefiltered ratio of 0.006% decreased to 0.002%, again retaining 50% of the prefiltered data. Interestingly, "TXT" queries increased to 0.215% after filtering, indicating that almost 100% of the prefiltered data was retained, demonstrating their resilience against filtering measures. As before, the total amount of prefiltered data remained small compared to overall traffic, further emphasizing the minor proportion of suspected DNS tunneling within the total dataset.
By 5-12-2023, the filtered data for "A" queries had further declined to 0.008%, representing only 18.2% of the prefiltered data. Conversely, "TXT" queries maintained their ratio at 0.145%, with 99% of the prefiltered data persisting. On 2-1-2024 and 6-2-2024, the "A" query traffic decreased further, showing that only 10.7% and 16.6% of the prefiltered data remained after filtering, respectively. In contrast, "TXT" queries consistently retained 100% of their prefiltered and filtered DNS data.
These results suggest that, while only a small percentage of the total DNS traffic is suspected to involve DNS tunneling, the filtering measures have varying levels of success depending on the query type. "A" and "AAAA" queries show significant reductions after filtering, while "TXT" queries remain resilient, consistently retaining nearly all their prefiltered data. The ratios for "NS" queries indicate that filtering methods may not strongly impact this query type.
From this analysis, it is evident that "TXT" queries consistently retain almost all of their prefiltered data across various dates, highlighting the limited impact of filtering on them. This strong association with DNS tunneling suggests that filtering mechanisms may need to be enhanced to address the challenges posed by "TXT" queries.
However, "A" and "AAAA" query types exhibit more substantial reductions after filtering, with only a fraction of their prefiltered data remaining. The variations in the NS query type, which sometimes increase after filtering, could indicate anomalies in traffic or specific characteristics of how these queries are processed.
In conclusion, the comparison between prefiltered and filtered data underscores the effectiveness of filtering for "A" and "AAAA" queries while revealing the need for targeted strategies for "TXT" queries, which remain a significant vector for DNS tunneling. This variation highlights the importance of tailoring detection and filtering strategies to account for the distinct behaviours of different query types.

6.2.3 Total IPs

In our final measurement, we focused on identifying the unique IP addresses of both IPv4 and IPv6 over the same six specific dates. Figure 8 shows the total number of unique IP addresses, revealing that IPv4 is more common in .nl traffic than IPv6.

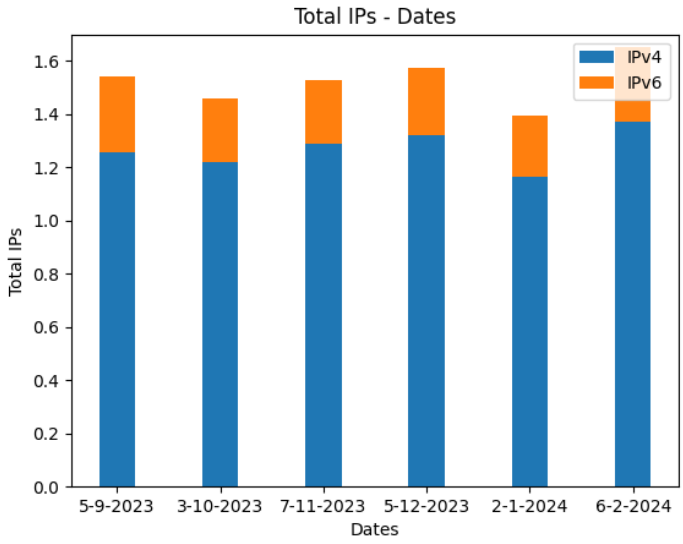
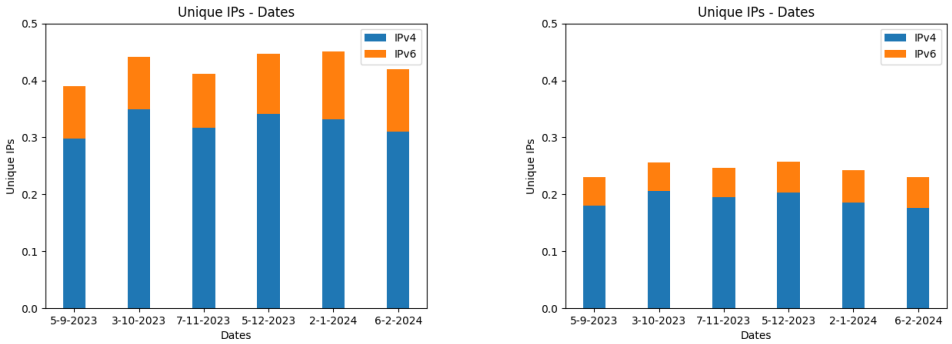


Figure 8: Total IP addresses of specific dates

Next, we measured the unique IP addresses in prefiltered DNS queries and potential DNS tunneling queries while applying custom rules. Figures 9(a) and 9(b) show that IPv4 addresses are more common than IPv6 addresses. Additionally, there is an increase in the ratio of IPv6 addresses in the prefiltered DNS queries, as shown in Figure 9(a). However, when custom rules are applied, Figure 9(b) shows a decrease in the number of unique IP addresses associated with potential DNS tunneling compared to the prefiltered data.



(a) Prefiltered unique IP addresses

(b) Filtered unique IP addresses

Figure 9: Unique IP addresses before and after applying custom rules

Figures 8 and 9 show the total, prefiltered, and filtered unique IPv4 and IPv6 addresses. We observe that IPv4 addresses generated the most traffic in .nl compared to IPv6 addresses. Additionally, Figure 9 shows a sharp reduction in the number of unique IP addresses compared to Figure 8 for all specific dates. Finally, we observe that the potential DNS tunneling retains a low amount of IPv4 and IPv6, as shown in Figure 8(b), relevant to prefiltered data and the total amount of IP addresses as presented in Figures 9(a) and 8, respectively.

6.3 Second Phase: Single Day Measurements

In the second phase of our measurements, we focused on a particular day that showed many suspicious and potential DNS tunneling queries. We also noticed that on this day, we noticed a specific hostname that appeared in many second-level domains. Therefore, we decided to evaluate this specific date.

6.3.1 Top 6 countries

Figure 10 illustrates the top six countries contributing to the total DNS queries observed on a specific date. Initially, the United States (cc US) had the highest traffic volume, followed by the Netherlands (cc NL), which accounts for approximately half as much DNS traffic as the US. Additionally, Germany (cc DE), Ireland (cc IE), and France (cc FR) were among the top countries with lower DNS query traffic on this day. Figure 10 also shows the United Kingdom (cc GB) appearing after France (cc FR). Overall, the top countries on this date are similar in terms of both countries and the amount of DNS queries compared to Figure 4 in the previous section.

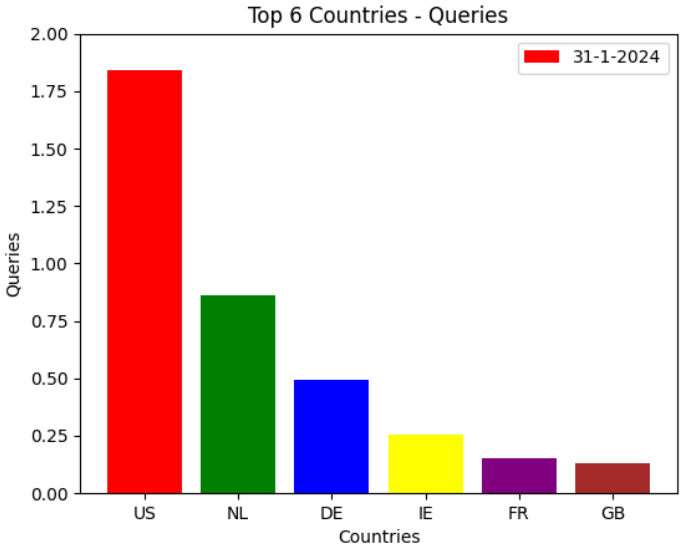
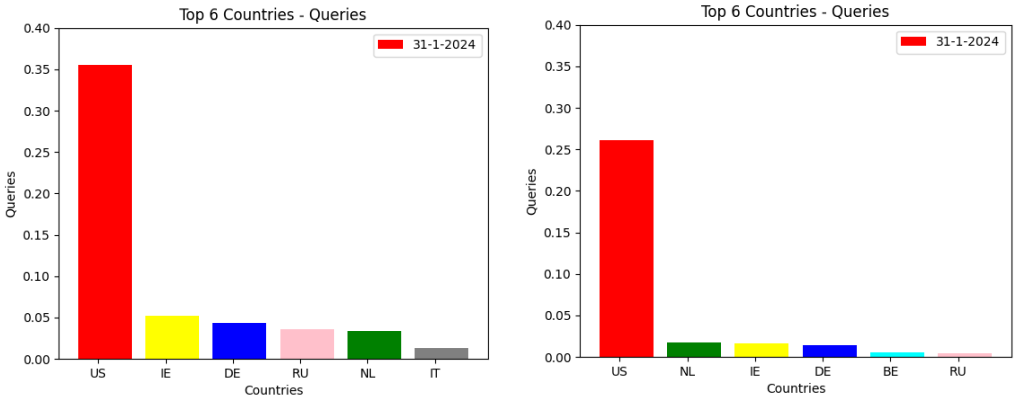


Figure 10: Total DNS queries of Top 6 countries of a single day

In Figures 11(a) and 11(b), we examined the top six countries represented in both prefiltered data and after applying custom rules. The United States (cc US) consistently has the highest traffic in both cases.
In Figure 11(a), we see that Ireland (cc IE), Germany (cc DE), Russia (cc RU), the Netherlands (cc NL), and Italy (cc IT) are the top countries in prefiltered DNS queries. However, Figure 11(b) shows a slightly different trend after applying custom rules, where the Netherlands (cc NL), Ireland (cc IE), and Germany (cc DE) become the prominent countries. Notably, Russia (cc RU) is among the top three countries after the United States (cc US) in prefiltered DNS queries but is replaced by the Netherlands (cc NL) after applying custom rules in Figure 11(b).



(a) Prefiltered Top 6 countries

(b) Filtered Top 6 countries

Figure 11: Top 6 countries before and after applying custom rules

In Figure 11, we observe that France (cc FR) and United Kingdom (cc GB) were replaced by Russia (cc RU), Italy (cc IT) and Belgium (cc BE).
Moreover, the data consistently shows Ireland (cc IE) and Germany (cc DE) among the top six countries across all metrics in the .nl traffic dataset for the specified date. However, when comparing Figures 5 and 11, we notice that the countries consistently present are the Netherlands (cc NL), Germany (cc DE), and Russia (cc RU). Subsequently, we observe that the countries Belgium (cc BE), Russia (cc RU) and Italy (cc IT) are not presented in the total amount of DNS queries in Figure 10 and are represented in prefiltered and filtered data as we noticed in Figure 11, which shows that the potential DNS tunneling for these countries' traffic is higher than the countries shown in Figure 10.
The analysis of DNS query ratios for the United States (cc US), Netherlands (cc NL), Germany (cc DE), Ireland (cc IE), and Russia (cc RU) reveals significant changes across different filtering stages, providing insights into the contribution of suspected DNS tunneling traffic relative to total DNS queries from these countries.

Country	**Prefiltered Ratio (Figure 11(a))**	**Filtered Ratio (Figure 11(b))**	**Filtering-to-Prefiltering Ratio**
United States	0.019%	0.013%	73.6%
Netherlands	0.006%	0.002%	15.5%
Germany	0.009%	0.003%	37.8%
Ireland	0.014%	0.006%	33.7%
Russia	0.014%	0.003%	35.1%

Table 8: DNS Query Ratios for Selected Countries

On 5-9-2023, the prefiltered data for the United States made up 0.019% of the total DNS queries, while the filtered data accounted for 0.013%. As a result, a ratio of 73.6% indicates that a significant portion of the suspected tunneling traffic remains after filtering. These findings suggest the presence of highly effective tunneling mechanisms, as a considerable amount of suspicious traffic remained after filtering.
For the Netherlands, the prefiltered data was only 0.006%, which decreased to 0.002% in the filtered dataset. The ratio of 15.5% from filtering to prefiltering indicates that most of the suspected DNS tunneling activity was successfully removed, showing that it is not a significant issue in the Netherlands. The filtering methods have effectively

decreased the amount of suspicious traffic. For Germany, prefiltered data starts at 0.009% and decreases to 0.003% after filtering, yielding a ratio of 37.8%. While some suspected tunneling activity is ongoing, a significant amount of the overall traffic is not related to tunneling activities, suggesting a modest level of effectiveness in filtering. The filtering successfully removed a substantial amount of questionable traffic, indicating the necessity for ongoing analysis to enhance detection capabilities. In Ireland, the percentage of prefiltered data decreased from 0.014% to 0.006% following filtering, resulting in a filtering-to-prefiltering 33.7% ratio. These results suggest that although some possible tunnelling activity exists, filtering has successfully decreased it overall. The moderate ratio implies continued tunnelling activity needs more monitoring and improved filtering techniques.

For Russia, the prefiltered data was 0.014%, which decreased to 0.003% in the filtered dataset, resulting in a ratio of 35.1%. After filtering, the relevance of Russian queries significantly decreased, suggesting that a smaller portion of its overall traffic is involved in tunnelling. This shows that the filtering systems can recognize and eliminate suspect traffic from Russian DNS requests.

Table 8 above summarizes the DNS query ratios for the selected countries, providing the prefiltered and filtered data along with their respective filtering-to-prefiltered ratios. The analysis indicates that DNS tunneling efficiency varies significantly across countries. The United States shows a substantial ratio of suspected tunneling queries after filtering, suggesting tunneling mechanisms are prevalent and require continuous monitoring. In contrast, the Netherlands demonstrates effective filtering with a low ratio, indicating minimal DNS tunneling activity.

Germany, Ireland, and Russia show moderate levels of suspected tunneling activity. The ratios suggest that significant portions have been successfully filtered out while some traffic remains. These different levels of effectiveness highlight the importance of customizing detection and mitigation strategies to each country's unique DNS traffic patterns. The research highlights the importance of consistently monitoring and improving filtering methods to better detect DNS tunneling in various geographical locations. Strengthening cybersecurity defences against evolving threats in DNS traffic will require enhanced filtering capabilities.

When examining the DNS query ratios relevant to subsection 6.2.1, we noticed similarities and differences across the countries we studied. While many countries showed relatively stable DNS query ratios over time, the data from 31-1-2024 highlights additional fluctuations, particularly in Russia. For example, Germany maintained consistent ratios of 0.008% across both dates and datasets (Figures 5(a) and 5(b)), indicating a steady DNS traffic pattern with minimal fluctuation. Similarly, the Netherlands shows only minor changes between the prefiltered (Figure 5(a)) and filtered data (Figure 5(b)) over time. The United States displays a slight increase in its ratios from 0.026% to 0.031% in prefiltered data (Figure 5(a)) and from 0.023% to 0.025% in filtered data (Figure 5(b)) between 5-9-2023 and 3-10-2023, suggesting a relatively consistent level of DNS query traffic even after filtering.

However, Russia's DNS query ratios show significant fluctuations. While 6.2.1 already highlighted changes in Russia's traffic from 14.83% on 5-12-2023 (Figures 5(a) and 5(b)) to 6.26% on 2-1-2024, the data from 31-1-2024 already introduced volatility. After this decrease, Russia's ratios rose again to 9.53% by 6-2-2024, indicating ongoing instability in DNS traffic. These fluctuations may be linked to changes in DNS tunneling activity or external factors influencing traffic patterns.

Finally, the data from 31-1-2024 emphasizes the diversity of DNS traffic trends across countries. While some, like Germany and the Netherlands, demonstrate stability, others, such as Russia, exhibit significant changes over time. This constant highlights the need for further investigation to better understand these trends, particularly in

countries like Russia, where DNS tunneling activity may contribute to the observed fluctuations.

6.3.2 Top Query Types

In Figure 12, we analyze various query types in the total DNS queries within the traffic on a specific date.
Initially, we observed that query type "A" makes up the most significant portion of DNS queries, commonly seen in daily .nl traffic. Following this, we noticed that query types "NS" and "AAAA" also appear frequently, with a slight difference in their frequency. Additionally, the query types "DS", "MX", and "TXT" contribute significantly to the volume of queries on this single day and also present similarities with Figure 6 related to the ratio of all query types in subsection 6.2.2.

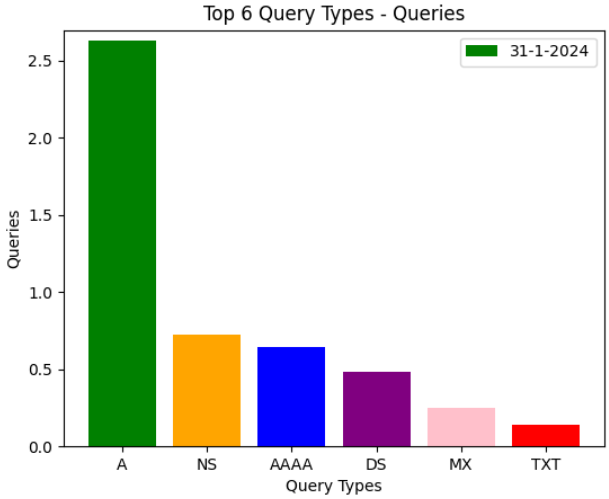
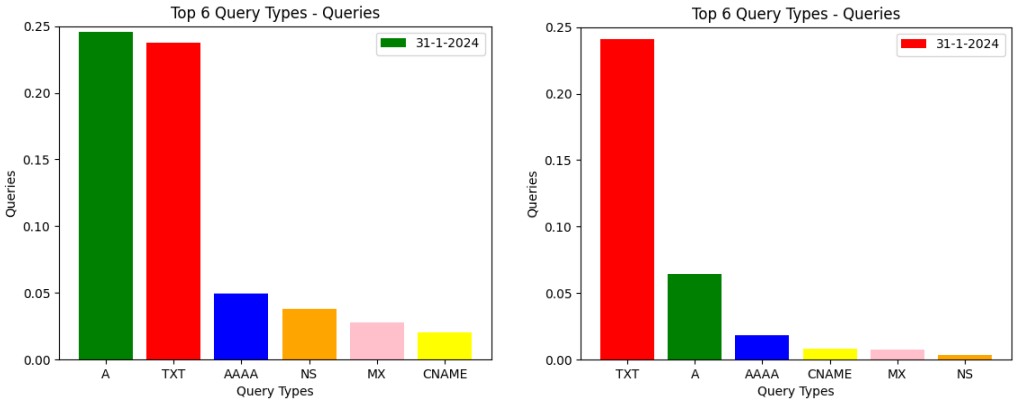


Figure 12: Top Query Types Prefiltered and Filtered

In Figure 13, we examined the distribution of query types in prefiltered queries and those processed with custom rules. Figure 13(a) illustrates that "A" and "TXT" records constitute a significant portion of the queries. In contrast, Figure 13(b) depicts a stable presence of "TXT" records with a notable decrease in "A" records compared to Figure 13(a). Also, we observe differences in subsection 6.2.2 that presented the "TXT" as the top query record in Figure 7(a).
Subsequently, in Figure 13(a), "AAAA" records emerge as the third most common query type, followed by "NS", "MX", and "CNAME" records with a slight decrease.
Conversely, in Figure 13(b), "AAAA" records represent the third-highest traffic, albeit nearly half the volume observed in Figure 13(a). Subsequently, we see in Figure 13(b) the "CNAME", "MX", and "NS" records with lower queries than in Figure 13(a). Finally, we see that Figure 13(a) exhibits differences from Figure 7(a) and similarities from Figure 13(b) with Figure 7(b) in subsection 6.2.2.



(a) Prefiltered Top 6 countries

(b) Filtered Top 6 countries

Figure 13: Top 6 countries before and after custom rules

Our analysis suggests that in Figure 13(b), the most common query types after filtering are "TXT" records. This suggests that "TXT" records may be linked to potential DNS tunneling queries. This observation contrasts with Figure 12, where "TXT" records are not commonly found in .nl traffic, just like in subsection 6.2.2.
The analysis of DNS query ratios at different filtering stages reveals how much suspected DNS tunneling traffic contributes to the total DNS queries for various query types. Understanding these ratios is crucial for evaluating the effectiveness of filtering mechanisms and the prevalence of tunneling activities. The presented ratios indicate the proportion of suspected DNS tunneling traffic concerning total traffic for each query type, demonstrating how effectively filtering mechanisms have reduced this traffic. A higher ratio suggests a more significant contribution of suspected DNS tunneling traffic, while a lower ratio implies effective filtering. Additionally, the ratios of filtered to prefiltered data highlight the extent of data reduction achieved through filtering.

Query Type	**Prefiltered Ratio (Figure 13(a))**	**Filtered Ratio (Figure 13(b))**	**Filtering-to-Prefiltering Ratio**
A	0.009%	0.002%	26.12%
TXT	0.17%	0.17%	100%
AAAA	0.007%	0.002%	37.55%
NS	0.005%	0.0004%	9.06%

Table 9: DNS Query Ratios for Different Query Types.

For the "A" query type, on the date of analysis, the prefiltered ratio is 0.009% of the total data, and the filtered data ratio is 0.002%. These ratios indicate that only a tiny fraction of the total "A" queries is suspected of tunneling. The filtering-to-prefiltering ratio of 26.12% suggests that only a tiny portion of "A" queries remains after filtering, implying that this type is less effective for DNS tunneling.
Conversely, the "TXT" query type exhibits a prefiltered data ratio of 0.17% of the total data, with a filtered ratio also at 0.17%. This results in a filtering-to-prefiltering ratio of 100%, signifying that nearly all suspected tunneling traffic for this query type persists after filtering. This high retention rate suggests that "TXT" records are particularly effective for DNS tunneling, facilitating efficient data exfiltration.
Regarding the "AAAA" query type, the prefiltered data ratio is 0.007% of the total

data, while the filtered data ratio is 0.002%. This results in a filtering-to-prefiltering ratio of 37.55%, indicating suspected DNS tunneling traffic retention. Although some data loss occurs, it remains a viable option for tunneling, albeit less effective than "TXT" records.
The "NS" query type has a prefiltered data ratio of 0.005% of the total data and a filtered data ratio of 0.0004%. A filtering-to-prefiltering ratio of 9.06% shows a significant reduction in data. These findings indicate that the NS query type is ineffective for DNS tunneling due to substantial loss of suspected tunneling traffic during filtering.
Table 9 above provides the query types based on prefiltered, filtered, and the corresponding filtering-to-prefiltering ratios, making interpreting the results easier to understand.
The analysis reveals that the "TXT" query type is the most effective for DNS tunneling, retaining nearly all of its prefiltered data after filtering, which allows for efficient data exfiltration. The "AAAA" query type shows reasonable data retention, making it a viable option for tunneling despite some data loss. Moreover, the "A" and "NS" query types encounter substantial reductions in suspected tunneling traffic, diminishing their effectiveness for DNS tunneling. Eventually, DNS tunneling is most efficient with query types that minimise data loss, with "TXT" records offering the highest data retention and effectiveness.
Based on the analysis from sections 6.2.2 and 6.3.2, similarities and differences in the behaviour of DNS query types across different filtering stages emerge. Both sections consistently show that the "TXT" query type retains nearly all its prefiltered data after filtering, underscoring its strong resistance to data reduction processes and confirming its effectiveness for DNS tunneling.
However, differences manifest in the retention ratios for other query types. For "A" queries, section 6.3.2 shows a stable reduction from 0.009% prefiltered to 0.002% filtered data. In contrast, section 6.2.2 reveals more variability across different dates, ranging from 0.008% prefiltered to 0.003% filtered on 5-9-2023 and as low as 0.0005% prefiltered to 0.00006% filtered on 2-1-2024. The "AAAA" query type in section 6.3.2 shows a reduction from 0.007% prefiltered to 0.002% filtered, while section 6.2.2 presents variations, such as 0.060% prefiltered to 0.034% filtered on 5-9-2023. The "NS" query type demonstrates the most notable differences; section 6.3.2 shows a substantial reduction, decreasing from 0.005% in the prefiltered data to 0.0004% in the filtered data. Section 6.2.2 reveals varying trends, with specific instances indicating increased data after filtering.
These differences suggest that while "TXT" queries remain primarily unaffected by filtering, the impact on other query types varies significantly depending on the dataset and date, reflecting dynamic filtering criteria and traffic characteristics.

6.3.3 IP addresses of a specific date

Table 6 shows the distribution of IPv4 and IPv6 addresses on a specific date. On this date, there are 1.4 million IPv4 addresses and 250,000 IPv6 addresses in total. After prefiltering, the numbers drop sharply to 31,000 IPv4 addresses and 16,000 IPv6 addresses. In the final filtering stage, there are 18,000 IPv4 addresses and 5,000 IPv6 addresses.

	Total IP addresses	Prefiltered IP addresses	Filtered IP addresses
IPv4	1.400.000	31.000	18.000
IPv6	250.000	11.000	5.000
Total	1.650.000	42.000	23.000

Table 10: IP addresses version 4 and version 6

Table 6 above illustrates significant differences between the total IP addresses related to prefiltered and filtered IP addresses. By providing the potential DNS tunneling included in a low amount of IP addresses based on the results of the custom rules that detect the potential DNS tunneling queries. The row of Total illustrates the total number of IP addresses between IPv4 and IPv6 that are related to total, prefiltered, and filtered data. The ratio of these IP addresses is expressed either as IPv4/IPv6 or IPv4/Total and IPv6/Total. Subsequently, we observe no sharp differences between prefiltered and filtered IP addresses. Finally, we see the similarity in all cases in subsection 6.2.3.

6.3.4 Unique Domain Names of a single day

In the final measurement, we assessed the correlation between DNS queries and second-level domains (domain names). Figure 14 depicts a histogram showing the total number of domain names within .nl traffic on a specific date. The histogram reveals a significant presence of second-level domains (domain names) associated with many queries on this particular date. The large number of queries results in nearly 30,000 domain names in the first half of this histogram. Additionally, many second-level domain names have only a few queries within the .nl traffic. These less frequent queries are shown in the second half of the histogram, as seen in Figure 14.

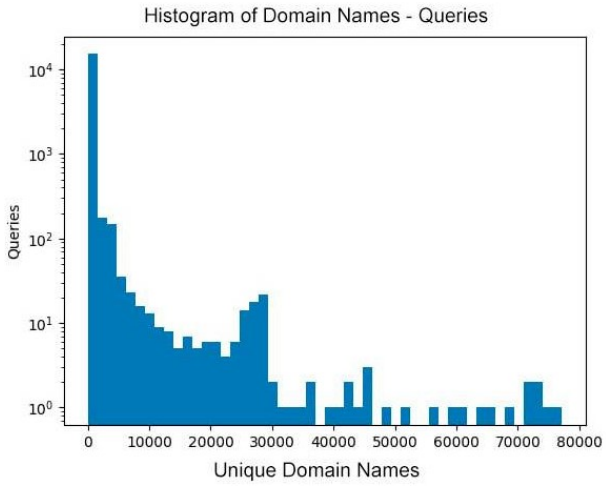
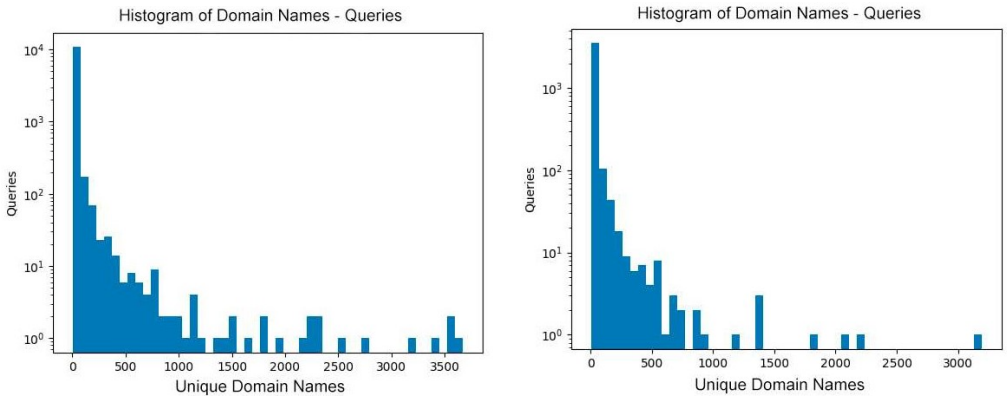


Figure 14: Histogram of Unique Domain Names Prefiltered and Filtered

The histograms in Figure 15 illustrate DNS queries associated with second-level domains (domain names), comparing data from prefiltered sources to those during custom rule implementation on the specified date. Although somewhat hard to see at first glance, the first histogram in Figure 15(a), representing prefiltered data, displays more second-level domain names than the second histogram in Figure 15(b). In the latter, fewer second-level domains (domain names) are depicted, as indicated by the x-axis label. Additionally, we observe a reduction related to the number of unique domain names

between figures 14 and 15. Subsequently, we see that in Figure 14, an improved amount between 10.000 and 15.000 unique queries and a sharp difference of almost 70.000 and 75.000 unique domain names disappeared in Figures 15(a) and 15(b). Figure 14 shows that the unique domain names are almost 80.000 for total DNS queries. Related to the unique domain names in Figure 15, they exist in 3.500 and 3.000 between the prefiltered and filtered data. The small number of domain names used for DNS tunneling indicates an attempt to blend in with regular .nl traffic, making the activity harder to detect. The finding is interesting because it highlights a significant difference in the number of unique domain names between Figures 14 and 15 after applying the prefiltered and filtered rules. Applying these rules reduces many unique domain names, excluding those not considered potential DNS tunneling domain names.



(a) Prefiltered Domain Names

(b) Filtered Domain Names

Figure 15: Unique Domain Names before and after custom rules

Figures 14 and 15 illustrate the differences between the total, prefiltered, and filtered unique domain names. These figures highlight the number of unique domain names at each stage. Figure 15(b) shows that the number of unique domain names and queries potentially associated with DNS tunneling is significantly lower than in Figures 14 and 15(a).

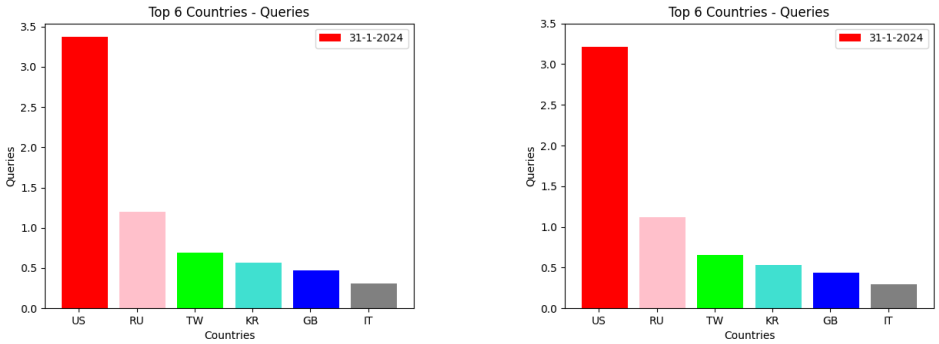
6.4 Third Phase: Specific Domain Name Measurements

We conduct this measurement focusing on a particular company's domain name in the Netherlands due to intriguing DNS queries observed on a specific date. We aim to understand the behaviour of this specific domain name, which may involve DNS tunneling techniques.

6.4.1 Top 6 Countries

Initially, we examine the top six countries appearing in the prefiltered data and during the implementation of custom rules in earlier subsections 6.2.1 and 6.2.3. Figures 16(a) and 16(b) depict the same top six countries. Otherwise, we noticed no difference in the volume of DNS queries between the prefiltered and filtered data. Additionally, we observe that the United States(cc US) remain steadily in the first position as in previous measurements in subsections 6.2 and 6.3. Subsequently, the significant point in this case is the appearance of Russia (cc RU) in the second position in both Figure 16(a) and 16(b), which is also presented in our previous measurements in subsections 6.2.1 and 6.3.1. Finally, Figure 16 follows countries that are not commonly used in previous

measures, such as Taiwan (cc TW), South Korea (cc KR), the United Kingdom (cc GB), and Italy (cc IT).



(a) Prefiltered Top 6 countries

(b) Filtered Top 6 countries

Figure 16: Prefiltered and Filtered Top 6 Countries

Figure 16 demonstrates no difference between the prefiltered and filtered data in the top six countries. As noted in subsections 6.2.1 and 6.3.1, the consistency of the top six countries, except for the United States (cc US) and Russia (cc RU), consistently holding the first positions, aligns with our previous measurements. Finally, we conclude that the United States (cc US) and Russia (cc RU) could be the top countries with potential DNS tunneling queries, impacting the potential targets in .nl traffic.

6.5 Summary of Measurements

Our analysis reveals a notable contrast in DNS query volumes between total queries and those filtered on specific dates. The United States consistently ranks among the top countries, indicating its significant influence on .nl traffic. This trend remains evident across different filtering stages, with the United States maintaining a leading position in total and filtered data. On specific dates, the ratio of DNS queries from the United States increased from 0.026% to 0.031% between the total and filtered data, underscoring its substantial role in potential DNS tunneling activities.

Russia displays a notable presence in the DNS query data, particularly within the prefiltered datasets. On 5-12-2023, Russia had a high ratio of 0.23% for prefiltered data, which dropped to 0.03% for filtered data, resulting in a combined ratio of 14.83%. This combined ratio decreased to 9.53% by 6-2-2024, with the prefiltered ratio declining to 0.15% and the filtered ratio to 0.014%. These variations highlight a significant shift in Russia's DNS query ratios over time. The observed decrease in the combined ratio suggests that while Russia's role in DNS tunneling is initially prominent, it becomes less pronounced when data is further refined. Emphasises the need for a thorough analysis of DNS query dynamics to accurately assess Russia's involvement in DNS tunneling and potential security threats.

The analysis of query types reveals that "TXT" records exhibit a high frequency of potential DNS tunneling queries, as detected by our custom filters. On 5-9-2023, the ratio of "TXT" to "A" queries was 0.203%, which remained consistent between the total and filtered data. This ratio slightly increased to 0.215% on 3-10-2023, indicating that "TXT" records become more prominent when isolating potential DNS tunneling queries. Applying these filters also uncovers a substantial difference in the distribution of unique IP addresses between total queries and filtered DNS data. Specifically, IPv4 addresses

decreased from 1.4 million to 18,000, while IPv6 addresses dropped from 250,000 to 5,000. This reduction reflects a concentration of potential DNS tunneling among fewer IP addresses.
After applying the filters, the histograms from the second phase of our analysis show a significant decrease in unique domain names, from approximately 80,000 to around 3,500. This sharp reduction highlights the effectiveness of our custom rules in isolating potential DNS tunneling domains and reducing the overall number of unique domain names associated with such activity.
Our findings underscore the impact of prefiltered rules and custom filters on .nl traffic. The United States and Russia emerge as key contributors to potential DNS tunneling queries, with significant ratios observed in their DNS query volumes. The "TXT" record ratio also plays a crucial role in identifying suspicious activity. The substantial reduction in unique IP addresses and domain names post-filtering further demonstrates the effectiveness of our targeted filtering approach in detecting and analyzing DNS tunneling.

7 Ethics

This study analyzed DNS queries received from .nl authoritative name servers, which contain sensitive details such as IP addresses, hostnames, second-level domains (domain names), Autonomous Systems, and the origin of DNS queries, including the name of the autonomous system provider.
Our method for detecting, analyzing, and measuring DNS tunneling[9, 10, 11, 12] is based on rules that do not pinpoint individual persons or companies. Instead, these custom rules focus on identifying features of DNS tunneling beyond the second-level domain of DNS queries. While some analyzed data requires personal information, we have obscured this information using asterisks (*), as shown in earlier sections. Additionally, in the measurement process, we only count the origin and IP addresses of DNS queries that do not contain personal details.
The ENTRADA tool adheres to a privacy framework[36] to ensure that data is used solely for maintaining and enhancing the security and stability of the .nl domain.
Subsequently, this research has been reviewed and validated by SIDN's privacy board, confirming that it aligns with ENTRADA's privacy policy.
In the course of this research, we do process personal information, such as IP addresses. However, we do not disclose this sensitive information in this thesis. Generally, the IP addresses in our dataset are associated with recursive resolvers, which serve thousands of end users. We do not attempt to identify individuals behind these IP addresses.
In conclusion, our final findings do not expose any personal information that could identify individuals or companies.

8 Conclusion

In this research, we explored the use of DNS tunneling techniques and tools[23], focusing on their distinct features within networks and how they can be leveraged for detection within the .nl top-level domain. Our study explores detection mechanisms to identify DNS tunneling activity in the volume of .nl traffic, answering research question R1. This question focuses on understanding which DNS tunneling techniques are employed, their distinct features within the network, and how these features can be used for detection on a TLD (.nl). Detection of DNS channels involves analyzing DNS query features within TLD (.nl) traffic using payload and traffic analysis, including high entropy, encoded types, uncommon resource records, and NXDOMAIN error types.

Additionally, in the measurements part in section 6, we observe that the DNS tunneling techniques leave a small amount of potential DNS tunneling queries involved in .nl traffic. This shows us that the DNS tunneling techniques aimed to operate are hidden in regular DNS traffic. Thus, we need to implement rules which previously identified footprints in DNS traffic at the authoritative name servers.
We examined the State-of-the-Art rules for detecting DNS tunneling techniques, answering Research Question R2. By exploring this question, we assessed the effectiveness of various existing detection methods and their application in identifying DNS tunneling activities. These rules include hostname entropy, encoded types beyond the second-level domain (domain name), uncommon record types, and the volume of NXDOMAIN responses in DNS queries. These rules were applied to DNS testbeds and .nl traffic to detect potential DNS tunneling queries, as seen in subsection 4.4.
Furthermore, we explored adapting detection techniques to higher levels in the DNS hierarchy, answering Research Question R3. We explored how detection techniques can be adapted to higher levels in the DNS hierarchy. We developed custom rules to be adaptable across different levels of the DNS hierarchy. By focusing on the leftmost labels beyond the second-level domain, we enable their use in various top-level domains (TLDs) and root DNS servers. The effectiveness of transforming state-of-the-art rules into custom rules analyzes the DNS data by disabling the QNAME minimization and using DNS queries only, as explained in subsection 4.3. This provides us with raw DNS data for improved analysis and detection of DNS tunneling.
Additionally, the custom rules included new characteristics from Rules 5 and 6 based on our observations on the DNS testbed. In developing custom detection rules, we provided the feasibility of adapting detection techniques to various levels of the DNS hierarchy. These custom rules efficiently detected potential DNS tunneling queries by incorporating all relevant features identified in DNS testbeds.
In the unique validation case that included the implementation of the actual DNS tunneling progress in .nl traffic, we successfully identified common features of DNS tunneling by applying our custom detection rules, which were developed and tested on a DNS testbed. The validation process involved a genuine user utilizing the Iodine DNS tunneling tool on .nl traffic under controlled conditions, deliberately disabling QNAME minimization to capture comprehensive DNS query data. By applying our detection rules to this data, we detected DNS tunneling queries with characteristics consistent with those observed in our DNS testbed. This successful detection validates the effectiveness of our State-of-the-Art rules, confirming their applicability in real-world scenarios and their potential for broader use in identifying DNS tunneling activities, answering the research question R4. In addressing Research Question R4, we examined whether the transformation of detection techniques leads to rules that effectively identify DNS tunneling.
Our final analysis focused on DNS tunneling techniques and identified that the United States consistently has the highest volume of activities within .nl traffic. This suggests a prominent role in potential DNS tunneling, as the high activity volume points to substantial involvement in these techniques rather than merely reflecting overall traffic volume. We also observed that other countries, like Russia, show varying ratios of DNS tunneling activities by analyzing these ratios rather than just total query volumes. Additionally, DNS tunneling queries often utilize "TXT" records, making it essential to monitor specific query types for more effective detection of DNS tunneling techniques. Overall, our findings answer research question R5, which examines the characteristics of identified DNS tunneling attempts and underscores the effectiveness of targeted filtering and detailed analysis in identifying such attempts. Additionally, we observed a significant volume of suspected DNS tunneling traffic originating from the United States and Russia.

9 Future Work

Future work can explore several key areas to enhance DNS tunneling detection and mitigation strategies:

- DNS Tunneling with QNAME Minimization
- Extension of Measurements
- Custom Rules on ccTLDs
- Custom Rules on Root DNS Servers
- Machine Learning Detection of DNS tunneling

By addressing these areas, we aim to advance the detection and prevention of DNS tunneling, thereby fortifying the resilience of digital infrastructures against evolving cyber threats.

- **DNS tunneling with QNAME Minization** While QNAME minimization enhances security by truncating DNS queries to reveal only necessary information, thereby reducing the risk of data exfiltration or abuse, it also poses a challenge for detecting potential DNS tunneling. This truncation limits the data available for analysis, making it more difficult to identify and track malicious activities within DNS traffic.

- **Extension of Measurements** Future research should extend the measurements in section 6 to focus on potential DNS tunneling queries, offering insights to understand and mitigate this cyber threat. Expanding the analysis to encompass a wide range of top-level domains (TLDs) can provide insights into the prevalence and distribution of potential DNS tunneling activity across different domains. Moreover, by examining temporal trends in DNS tunneling queries over an extended measurement period, we can identify evolving patterns and emerging tactics employed by malicious actors, enabling proactive measures to mitigate risks and enhance network security.

- **Custom rules on ccTLDs** The custom rules designed to identify DNS tunneling techniques[9, 10, 11, 12, 13, 14] and tools[23] aren't limited to just .nl traffic. They can be applied universally by companies or organizations managing country code top-level domains (ccTLDs) worldwide. These rules focus on analyzing the leftmost labels of DNS queries beyond the second-level domain (registered domain) present in DNS traffic. By implementing these rules across all ccTLDs, we can effectively detect potential DNS tunnelling tools, ensuring higher security across the DNS infrastructure.

- **Custom rules on root DNS servers** Implementing the custom rules on root DNS servers is a viable option. These rules are designed to be applicable across all country code top-level domains (ccTLDs), and they target the leftmost labels of DNS queries, which come after the second-level domain. This approach enables the detection of DNS tunneling at the highest level of the DNS infrastructure.
  We can effectively identify and mitigate DNS tunnelling activities by deploying these custom rules at the root DNS servers. This proactive measure may help maintain a robust and resilient DNS infrastructure.

- **Machine Learning Detection of DNS tunneling** Machine learning[12] approach can enhance our ability to detect and mitigate DNS tunneling. Machine

learning algorithms establish normal behaviour patterns by analyzing large volumes of DNS traffic data. They could continuously monitor DNS queries, identifying anomalies and potential tunneling activities in real-time. Additionally, machine learning automates the extraction of critical features from DNS queries, such as query length and frequency, aiding in distinguishing legitimate from malicious traffic. These models can adapt over time, learning from new data and updates to recognize evolving threats. Machine learning may reduce false positives, improving efficiency in security responses. Leveraging machine learning for DNS tunneling detection strengthens global DNS infrastructure protection.

References

[1] RFC 1035, "Domain Names - Implementation and Specification", P. Mockapetris, the Internet Society (November 1987)

[2] Adam Ali.Zare Hudaib, Esra'a Ali Zare Hudaib, "DNS Advanced Attacks and Analysis" (2014)

[3] RFC 768, J. Postel, ISI, 28 August 1980, "User Datagram Protocol"

[4] RFC: 793, "TRANSMISSION CONTROL PROTOCOL", DARPA INTERNET, PROGRAM PROTOCOL SPECIFICATION, September 1981

[5] Request for Comments: 9364, Internet Engineering Task Force (IETF), P. Hoffman, ICANN, February 2023, DNS Security Extensions (DNSSEC)

[6] Request for Comments: 6347, Internet Engineering Task Force (IETF), E. Rescorla, RTFM, Inc., N. Modadugu, Google, Inc., January 2012, Datagram Transport Layer Security Version 1.2

[7] Network Address Translation,
`https://en.wikipedia.org/wiki/Network_address_translation`

[8] Two-Way Communication,
`https://en.wikipedia.org/wiki/Two-way_communication`

[9] Greg Farnham, "Detecting DNS Tunneling" (2013).

[10] Jiangang Hou, Xin Li, Kun Zhao, Wei Liang, Yanmiao Li, Tongqing Jiang, Zhi Liu, "A Survey of DNS Tunnel Detection" (2022).

[11] Mahmoud Sammour, Burairah Hussin, Mohd Fairuz Iskandar Othman, Mohamed Doheir, Basel AlShaikdeeb, Mohammed Saad Tablib, "DNS tunneling: A Review of Features" (2018).

[12] Yue Wang, Anmin Zhou, Shan Liao, Rongfeng Zheng, Rong Hu, Lei Zhang, "A comprehensive survey on DNS tunnel detection"(2021)

[13] Jingkun Liu, Shuhao Li, Yongzheng Zhang, Jun Xiao, Peng Chang, Chengwei Peng, "Detecting DNS Tunnel through Binary-Classification Based on Behavior Features" (2017)

[14] Gianni D'Angelo, Arcangelo Castiglione, Francesco Palmieri, "DNS tunnels detection via DNS-images" (2022)

[15] VirtualBox
`https://www.virtualbox.org/`

[16] TCP over DNS tunnels,
https://blogs.infoblox.com/community/analysis-on-popular-dns-tunneling-tools/

[17] BIND9 DNS software
https://www.isc.org/bind/

[18] Unbound software
https://unbound.docs.nlnetlabs.nl/en/latest/

[19] Tcpdump tool,
https://www.tcpdump.org/

[20] Wireshark network protocol analyzer,
https://www.wireshark.org/

[21] RFC 4253, "The Secure Shell (SSH) Transport Layer Protocol", T. Ylonen, SSH Communications Security Corp C, Lonvick, Ed. Cisco Systems, Inc. January 2006

[22] Secure Copy Protocol,
https://en.wikipedia.org/wiki/Secure_copy_protocol

[23] Iodine DNS tunneling tool,
https://code.kryo.se/iodine/

[24] Hadoop Apache library,
https://hadoop.apache.org/

[25] Impala Apache,
https://impala.apache.org/index.html

[26] Heyoka DNS Tunneling tool,
https://heyoka.sourceforge.net/

[27] Shannon Entropy Definition,
https://en.wikipedia.org/wiki/Entropy_(information_theory)

[28] Request for Comments: 4648, S. Josefsson SJD, October 2006, "The Base16, Base32, and Base64 Data Encodings"

[29] Request for Comments: 1002, March 1987, "PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: DETAILED SPECIFICATIONS"

[30] Levensthein Distance,
https://en.wikipedia.org/wiki/Levenshtein_distance

[31] RFC 7816, "DNS Query Name Minimisation to Improve Privacy", S. Bortzmeyer, AFNIC, March 2016

[32] Maarten Wullink, Moritz Müller, Marco Davids, Giovane C. M. Moura, and Cristian Hesselman, "ENTRADA: Enabling DNS Big Data Applications" (2016)

[33] Internet Engineering Task Force (IETF), D. Crocker, Ed., Brandenburg InternetWorking, Request for Comments: 6376, T. Hansen, Ed., AT&T Laboratories ISSN: 2070-1721, M. Kucherawy, Ed. Cloudmark, September 2011, "DomainKeys Identified Mail (DKIM) Signatures"

[34] AWS Instance Naming Queries,
`https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-naming.html`AWS Instance Naming Queries

[35] Azure Instance Naming Queries,
`https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming`

[36] Dr. ir. C.E.W. Hesselman, drs. J.R.P. Jansen, drs. M. Wullink, mr. A.K. Vink en mr. M.M. Simon, "Wetenschappelijk artikel Een privacyraamwerk voor 'DNS big data'- toepassingen"(2014)

10 Appendix 1

Part A
Initially, we install the Iodine server-side (daemon) using terminal commands `sudo apt install iodine`. Next, we configure the Iodine server by editing the configuration file using the following command `sudo vim \etc\default\iodine`. The configuration file includes three parameters:

- `START_IODINED="true"` activates the DNS tunnel.
- `IODINED_ARGS="-c -l 10.0.2.12 172.16.0.1 iodine.sidndam.nl"` includes arguments to configure the DNS tunnel, enabling client-side caching option `-c`, specifies the local IP address `-l` and setting the server's domain name.
- `IODINED_PASSWORD="********"` specifies the password required for both server and client to initiate the DNS tunnel.

Part B
The running of the Iodine DNS tunneling tool involves running the command `sudo iodined -c -f -l 10.0.2.12 172.16.0.1 iodine.sidndam.nl` to initialize the server.
The Iodine client-side implemented using the command `sudo iodine -P ******** -L0 -I1 -f iodine.sidndam.nl`. This command configures the client with the following options, including the password of the DNS tunnel `-P`, the local DNS port of the target machine `-L0`, the interval between DNS requests in milliseconds `-I1` and the operation in the foreground `-f`.