Detection and Mitigation of Malicious Activities Based on DNS Query Analysis

Georgi Markov
Department of Information Technologies
Nikola Vaptsarov Naval Academy
Varna, Bulgaria
g.markov@naval-acad.bg

Borislav Nikolov
Department of Information Technologies
Nikola Vaptsarov Naval Academy
Varna, Bulgaria
nikolov@naval-acad.bg

***Abstract*-With the increasing number of cyber threats and the growing complexity of attacks on network infrastructures, the need for effective methods to detect malicious activities has become critically important. One of the key attack vectors is the Domain Name System (DNS), which plays a fundamental role in internet communication. Although DNS is essential for every end user, it often remains unnoticed and unprotected, making it vulnerable to abuses such as DDoS attacks, attack surface reconnaissance, and data exfiltration. The aim of this study is to develop a method for automated analysis of DNS traffic to enable early detection of suspicious patterns and prevent potential attacks. To achieve this, open-source tools, publicly available databases, and log files from a real authoritative DNS server are utilized. The methodology includes analysing the frequency and type of DNS queries, as well as evaluating the IP addresses from which they originate. The results of the analysis demonstrate that automated processing of DNS logs allows for the identification of anomalous query patterns associated with malicious activities. Systematic monitoring of DNS traffic provides an opportunity for early threat detection and faster implementation of protective measures. The proposed approach enhances cybersecurity mechanisms by strengthening threat intelligence capabilities and automating the detection process. This underscores the significance of the research and the necessity of continuously improving protection methods in the dynamic landscape of cybersecurity.**

Keywords- DNS query, automation, logging, Python script, cybersecurity.

I. Introduction

The Domain Name System (DNS) is a fundamental component of modern computer networks, particularly in Internet-connected environments. It allows users to access network resources and services without the need to memorize numerical IP addresses, facilitating seamless interaction between clients and hosts. From the perspective of public institutions and commercial enterprises, DNS serves as a critical infrastructure element, providing accessibility and visibility for online services to end users.

DNS operates as a hierarchical and distributed naming system. Organizations that wish to establish a public domain must configure and maintain an authoritative DNS server, which is responsible for resolving Fully Qualified Domain Names (FQDNs) to corresponding IP addresses. To achieve this, authoritative DNS servers maintain a local database of resource records, including forward lookup zones and, in many cases, reverse lookup zones. Key resource records include A/AAAA (host addresses), PTR (reverse lookup), SOA (Start of Authority), DS (Delegation Signer), MX (Mail Exchanger), SRV (Service Locator), NS (Name Server), and TXT records, which provide various forms of metadata [1] – [4].

Despite its critical role, DNS is increasingly being exploited in cyber-attacks. Malicious actors frequently attempt to exfiltrate data by performing unauthorized DNS zone transfers or leveraging lesser-known DNS record types to obtain sensitive information. Attackers also take advantage of the fact that DNS traffic is often unrestricted by organizational firewalls, allowing it to serve as a covert channel for data exfiltration, botnet communication, and command-and-control (C2) operations [5] – [7]. Studies highlight that DNS tunnelling techniques have been widely adopted by cybercriminals to bypass traditional security measures, making DNS security an essential aspect of modern cybersecurity frameworks [8] – [12].

Given the increasing sophistication of DNS-based threats, there is a growing need for automated detection and mitigation mechanisms. Existing research explores various approaches for identifying suspicious DNS activities, such as anomaly detection methods based on statistical analysis, machine learning, and rule-based filtering systems [13] – [14]. However, many of these methods require significant computational resources or lack adaptability to evolving attack patterns.

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol2.8582

The growing number of cyber threats leveraging DNS infrastructure underscores the necessity of real-time monitoring and automated threat mitigation solutions. This study aims to develop a practical, automated approach for analysing DNS query patterns to detect and mitigate malicious activities targeting authoritative DNS servers. Unlike conventional detection methods, which often rely on static signatures or reactive security measures, the proposed approach integrates real-time log analysis, rule-based filtering, and automated threat intelligence validation using public databases.

This paper contributes to the field by presenting an efficient, open-source-based solution for analysing DNS traffic, dynamically generating firewall rules, and preventing future cyber threats. Through periodic execution, the proposed system adapts to new attack vectors and reduces the risk of repeated exploits. The comparative analysis of literature sources further highlights the necessity of combining statistical, behavioural, and rule-based techniques for enhanced DNS security.

By addressing both theoretical and practical aspects of DNS security, this study provides a foundation for improving network resilience against cyber threats while ensuring minimal disruption to legitimate traffic.

II. Materials and methods

A. Defining the scope of DNS analysis

Before proceeding with the analysis of DNS queries, the scope of the analysis should be determined. Depending on the purpose of the authoritative DNS server, different types of queries can be observed, as well as different percentage ratios between them.

To determine the scope of our research, as well as to confirm the functionality of the developed approach for automating the analysis of DNS queries, we will use the Bulgarian Naval Academy's (NVNA) authoritative public DNS server.

The Bulgarian Naval Academy is a higher educational institution with a technical profile in the fields of military affairs, maritime sciences, and information technologies. The school is part of the Ministry of Defence of Bulgaria. Nevertheless, it provides education for many foreign students, mainly from Turkiye and Greece. The school maintains very good international contacts with educational institutions and business companies from other NATO and/or European Union member or partner countries.

Regarding its presence in the Internet space, the Bulgarian Naval Academy owns the Internet domain (*naval-acad.bg*), as well as a class C public network (*194.XXX.XXX.0/24*). The authoritative DNS server for this domain uses BIND9 software [15]. Local clients with private IP addresses use the same DNS server to resolve internet names.

Given the specified profile of this educational institution, several reasonable assumptions can be made regarding legitimate Internet traffic to the domain naval-acad.bg, which in turn can provide a basis for analysis and classification of suspicious or malicious traffic.

For example, legitimate DNS queries can be considered to be all those directed to the authoritative DNS server of the domain naval-acad.bg for resolving the FQDN names of publicly available Internet services provided by the academy, such as the official web page, e-mail, student status system, etc. It can also be reasonably assumed that such queries will essentially come from IP addresses from "friendly" or "partner" countries. Any other type of query can be classified as suspicious and subject to additional analysis.

With the framework thus set for implementing automated analysis of DNS queries, the flowchart presented in Fig. 1 can be followed. The flowchart can be adapted to other types of authoritative DNS servers.

Fig. 1 presents the procedure for performing the preparatory part of the analysis. We assume that the necessary DNS software has already been installed (in our case, this is BIND9, but the principle of operation will be identical for other DNS servers).

By default, BIND9 does not record separate event logs for DNS queries it processes. Audit logging must be enabled by adding an option to its configuration file.

Once audit logging is enabled, it is necessary to determine where the logs will be stored and how they will be managed. If this configuration is not performed, all audit events related to the BIND9 operation will be recorded in the main system log (in most cases, this is the /var/log/syslog file). The syslog file is quite dynamic, so we chose an approach in which audit logs from BIND9 will be recorded in separate files. This allows us to perform an easier analysis of the recorded events. Also, these log files can be shared with third parties for additional evaluation and analysis, if necessary, without providing any other information [16] – [17].

We have chosen an approach where different events are recorded in separate log files. As a result, we have 17 separate categories [18] of audit records, which allows us to perform detailed and quick analysis of different types of queries or other types of events related to the operation of the BIND9 DNS server. In the process of work, it was found that four categories of audit records are mainly used:

- default.log – audit records that do not fall within the scope of any of the other defined categories. A review of this log file for the server in question shows that it records events related to queries that are outside the scope of its defined forward and reverse lookup zones.
- lame-servers.log – audit records in case a problem is detected in the resolution chain to other DNS servers.
- queries.log – main log file for recording incoming queries.
- security.log – contains information about rejected queries for zone transfers or cached

information. It should be noted that the configuration of the DNS server in question prohibits performing zone transfers or cached information, but this information is useful for detecting malicious activity against the server and the domain it serves.

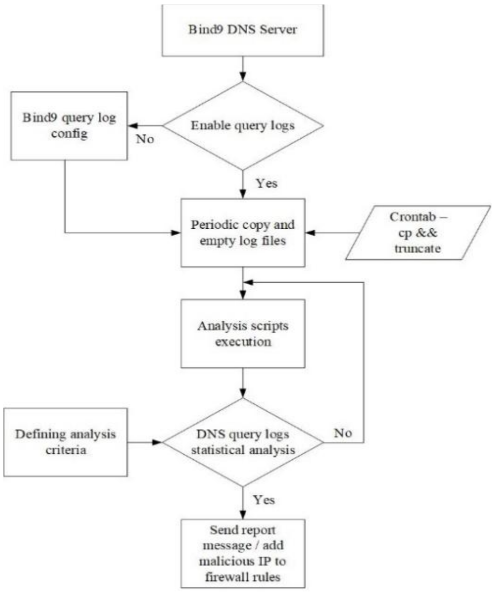

Bind9 DNS Server
Enable query logs
No
Bind9 query log config
Yes
Periodic copy and empty log files
Crontab – cp && truncate
Analysis scripts execution
Defining analysis criteria
DNS query logs statistical analysis
No
Yes
Send report message / add malicious IP to firewall rules

Fig. 1. DNS query analysis process flowchart.

The analysis we perform is on the audit records from these four log files. If malicious activity is detected, the other log files can be used by administrators for a more detailed analysis of the events. For this reason, the storage of three time versions of the log files has been determined, which should be a sufficiently long period of time to organize their storage for a longer period.

We have adopted an audit logs analysis approach that uses periodic copies of log files. For this purpose, we have configured a periodic task that copies all BIND9 log files, regardless of the number of records in them, and then resets the contents of the files to the main storage location. The latter is done to avoid duplication of analysis queries. The periodic task is configured to run every 30 minutes. The period is chosen by considering the average duration of WWW use by ordinary users, during which DNS queries are generated. As shown [19], the average duration of WWW use is about two hours per day. We assume that a session will hardly exceed 30 minutes, and by a session, we mean visiting one specific web page or those that are related to it. To exclude missing sessions that fall within the scope of two consecutive executions of the periodic task, we also store log files from the previous execution. In this way, the subsequent analysis is performed over 60 minutes.

B. DNS query logs analysis script development

We perform the analysis itself using Python script, and as evaluation criteria, we have defined the occurrence of the following events:

- A zone transfer query.
- A cache transfer query.
- An ANY type query.
- Queries from a single client exceed the percentage distribution of the same type of queries among all clients. For thresholds of the percentage distribution of queries, we use the data from [1].

The defined criteria should be periodically assessed and adjusted if necessary [20] – [24].

For individual clients for whom the volume of queries generated by them raises doubts about the purpose of their activity, an additional check is performed in a public database [25] – [26]. This evaluation is done using a separate Python script [27] – [29].

If suspicious clients are identified during the statistical analysis, information about their activity and the result of the check in the public database are sent to the email address of the DNS server administrator, and the client's IP address is added to a blacklist and BIND9 will automatically reject all subsequent queries from this client.

Fig. 2 presents the workflow of the created script for analysing DNS queries. The script is configured to run periodically after completing the log file copying.

The first step of its execution is to read log files from the specified working folder. This is followed by generating summary information for all clients and types of received queries. This summary information is saved in a separate file to be added to the analysis during the next script execution cycle to avoid missing client sessions that fall on the border between the 30-minute log files.

The script summarizes data from the current and previous cycles and compares the results to predefined criteria to determine suspicious activity. The current criteria are:

- IP address sends a zone transfer query;
- IP address sends a query of type “ANY”;
- IP address sends a query of type “SOA”;
- IP address sends more than 10% of all received queries;
    - IP address sends more than 2% of all received “PTR”, “A”, “AAAA”, or “TYPE65” queries;
    - The percentage distribution of queries from a single IP address exceeds any of the following thresholds – 40% type “A”, 10% type “AAAA”, 5% type “PTR”, 15% type “TYPE65”. This rule applies when there are at least 20 queries generated from a single IP address.

The next step in executing the script is to perform an online check against a publicly available database of the IP addresses identified in the previous step. It checks whether these IP addresses have already been reported for malicious activity. This check is performed only for public IP addresses.

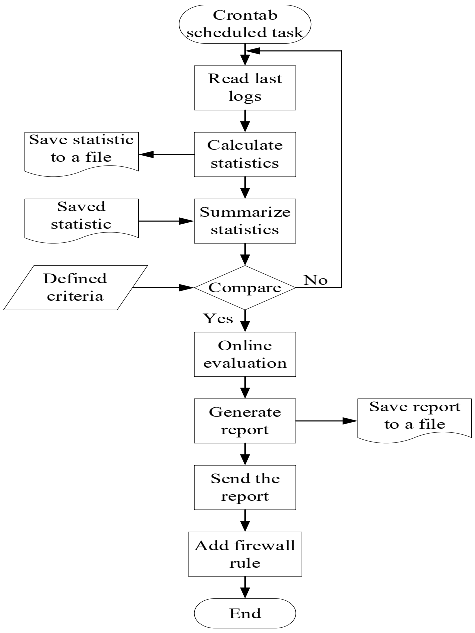


Fig. 2. DNS query analysis script flowchart.

The summary report is then generated and stored in a file. The file is then emailed to the DNS server administrators.

The script also generates inbound traffic rules to be added to the server's firewall. In this case, it is iptables. These rules aim to drop any type of traffic from the specified IP addresses. Information about the generated rules is added to the email sent to administrators, who have the opportunity to review them and, if necessary, modify or remove them.

This completes the script execution cycle and the next moment for its execution is awaited.

III. Results and discussion

We have assumed that reporting malicious IP addresses is not a continuous process, but usually requires a certain response time, both on the part of the reporter and on the part of the public databases. For this reason, we have configured checking suspicious IP addresses to be performed once every 24 hours. For this purpose, we use an additional file with information accumulation, which records all IP addresses checked in the public database for each script execution cycle. The contents of this file are reset every day.

Using such a file prevents us from duplicating the generated iptables rules for a single IP address. Without such stored information, every two consecutive script execution cycles would cause the generation of two iptables blocking rules that concern the same IP address. We have adopted a threshold of 75% "bad" reputation of a checked IP address for a rule to be generated for its blocking.

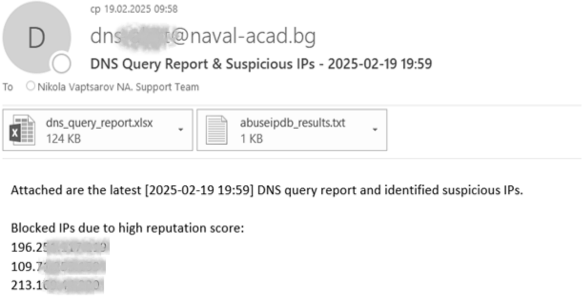
The execution of each script cycle ends with sending an email to the DNS server administrators (Fig. 3).

This email contains the following information:

- Date and time of script execution;
- List of IP addresses for which blocking iptables rules have been generated;
- Attached file containing complete statistics of processed DNS logs in Excel spreadsheet format. Part of the content of this file is presented in Fig. 4.
- Attached file containing a report of the checked IP addresses in the public database with the score received for them (Fig. 5). This file contains information about all checked IP addresses with their scores, not just the blocked ones.

The script being executed colors the data for each client IP address when it matches any of the specified conditions described above, using the following color coding:

- Zone transfer queries – “Red” (FF0000);
- Queries of type “ANY” – “Light Red” (FF9999);
- Queries of type “SOA” – “Orange” (FF9900);
- Other matched rules – “Yellow” (FFFF00).



ср 19.02.2025 09:58

D

dns[redacted]@naval-acad.bg

DNS Query Report & Suspicious IPs - 2025-02-19 19:59

To ○ Nikola Vaptsarov NA. Support Team

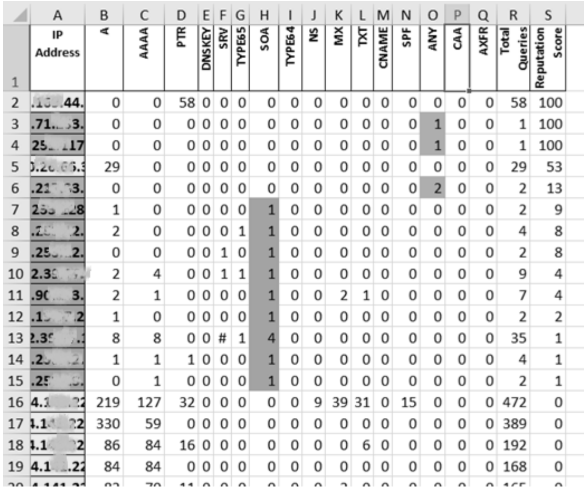
dns_query_report.xlsx
124 KB

abuseipdb_results.txt
1 KB

Attached are the latest [2025-02-19 19:59] DNS query report and identified suspicious IPs.

Blocked IPs due to high reputation score:
196.25[redacted]
109.7[redacted]
213.1[redacted]

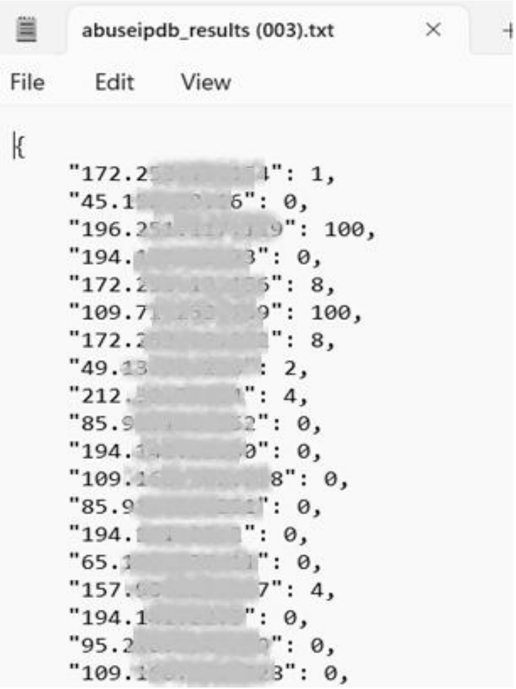
Fig. 3. Content of a received email with a report on the script execution.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	IP Address	A	AAAA	PTR	DNSKEY	SRV	TYPE65	SOA	TYPE64	NS	MX	TXT	CNAME	SPF	ANY	CAA	AXFR	Total Queries	Reputation Score
2	.1[redacted]44.	0	0	58	0	0	0	0	0	0	0	0	0	0	0	0	0	58	100
3	.71.[redacted]3.	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	100
4	25[redacted]17	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	100
5	[redacted]	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	53
6	.2[redacted]3.	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	13
7	2[redacted]28	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	9
8	.2[redacted]2.	2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	4	8
9	.25[redacted]2.	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2	8
10	2.3[redacted]	2	4	0	0	1	1	1	0	0	0	0	0	0	0	0	0	9	4
11	.9[redacted]3.	2	1	0	0	0	0	1	0	0	2	1	0	0	0	0	0	7	4
12	.1[redacted]2	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	2
13	2.3[redacted]	8	8	0	0	#	1	4	0	0	0	0	0	0	0	0	0	35	1
14	.2[redacted]	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	4	1
15	.2[redacted]	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1
16	4.1[redacted]2	219	127	32	0	0	0	0	0	9	39	31	0	15	0	0	0	472	0
17	4.1[redacted]22	330	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	389	0
18	4.1[redacted]2	86	84	16	0	0	0	0	0	0	0	6	0	0	0	0	0	192	0
19	4.1[redacted].22	84	84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	0

Fig. 4. Content of the Excel file with statistics of processed logs.

The public IP addresses in Fig. 3, Fig. 4, and Fig. 5 are intentionally masked in this paper.



```
{
    "172.2[redacted]": 1,
    "45.1[redacted]6": 0,
    "196.[redacted]": 100,
    "194.[redacted]": 0,
    "172.[redacted]": 8,
    "109.7[redacted]": 100,
    "172.[redacted]": 8,
    "49.[redacted]: 2,
    "212[redacted]": 4,
    "85.[redacted]": 0,
    "194.[redacted]": 0,
    "109.[redacted]8": 0,
    "85.9[redacted]": 0,
    "194.[redacted]": 0,
    "65.[redacted]": 0,
    "157.[redacted]7": 4,
    "194.1[redacted]": 0,
    "95.2[redacted]": 0,
    "109.[redacted]": 0,
```

Fig. 5. Contents of the file with the checked IP addresses and their reputation score.

TABLE 1 NUMBER OF IPTABLES RULES GENERATED

Number of days since the initial script execution	Number of blocked IP addresses
Day 1	8
Day 2	8
Day 3	6
Day 4	3
Day 5	3
Day 6	7
Day 7	4
Day 8	1

Table 1 presents data on the generated iptables rules over a specific period, with the first day covering less than a full 24-hour cycle.

IV. CONCLUSIONS

A Python script has been developed to analyse the log files of a BIND9 DNS server. By applying predefined rules and cross-referencing data with a public database, it generates summarized statistics on potential malicious activities targeting the server. The process of creating filtering rules for incoming Internet traffic, based on the assessed reputation of IP addresses initiating DNS queries, has been automated. This automation enhances the server's security by enabling proactive threat mitigation and reducing the risk of repeated attacks.

With the script's periodic execution, iptables rules are dynamically generated to block IP addresses that match any predefined criteria. These rules are applied immediately upon detection of suspicious activity, ensuring real-time protection. A decreasing trend in the number of generated rules has been observed (Table 1), suggesting the effectiveness of the filtering mechanism in reducing malicious traffic over time.

The proposed approach relies exclusively on open-source software, making it accessible, cost-effective, and adaptable for different environments. Future improvements may include the integration of machine learning techniques for adaptive threat detection and enhanced accuracy in identifying malicious behaviour.

ACKNOWLEDGMENTS

This work was supported by the National Scientific Program "Security and Defence", adopted with RMS No. 731/21.10.2021, and financed by the Ministry of Education and Science of the Republic of Bulgaria according to Agreement No. D01-74/19.05.2022.

REFERENCES

[1] J. Ginesin and J. Mirkovic, Understanding DNS Query Composition at B-Root, 2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), Dec 06-09, 2022, Vancouver, WA, USA, pp. 265-270, doi: 10.1109/BDCAT56447.2022.00044

[2] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034, Nov. 1987. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1034 [Accessed February 2, 2025].

[3] P. Mockapetris, "Domain names - implementation and specification," RFC 1035, Nov. 1987. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1035 [Accessed February 2, 2025].

[4] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, "DNS Security Introduction and Requirements," RFC 4033, Mar. 2005. [Online]. Available: https://datatracker.ietf.org/doc/rfc4033/ [Accessed February 2, 2025].

[5] Y. Wang, A. Zhou, S. Liao, R. Zheng, R. Hu and Lei Zhang, "A comprehensive survey on DNS tunnel detection," Computer Networks, vol. 197, 2021. doi: 10.1016/j.comnet.2021.108322

[6] Y. Zhauniarovich, I. Khalil, T. Yu and M. Dacier, "A Survey on Malicious Domains Detection through DNS Data Analysis," ACM Computing Surveys, vol. 51, 2018. doi: 10.1145/3191329

[7] I. Ghafir and V. Prenosil, DNS Traffic Analysis for Malicious Domains Detection, 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), February 19-20, 2015, Noida, India.

[8] A. Ramdas and R. Muthukrishnan, A servey on DNS Security Issues and Mitigation Techniques, Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019), June 27-28, 2019, Secunderabad, India.

[9] K. Borgolte, T. Chattopadhyay, N. Feamster, M. Kshirsagar, J. Holland, A. Hounsel and P. Schmitt, "How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem," SSRN Electronic Journal, 2019. doi:10.2139/ssrn.3427563

[10] I. Dube and G. Wells, An Analysis of the Use of DNS for Malicious Payload Distribution, 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), November 25-27, 2020, Kimberley, South Africa. doi: 10.1109/IMITEC50163.2020.9334104

[11] M. Luo, Q. Wang, Y. Yao, X. Wang, P. Yang and Z. Jiang, Towards Comprehensive Detection of DNS Tunnels, 2020 IEEE Symposium on Computers and Communications (ISCC), July 07-10, 2020, Rennes, France. doi: 10.1109/ISCC50000.2020.9219547

[12] S. Lysenko, K. Bobrovnikova, O. Savenko and R. Shchuka, Technique for Cyberattacks Detection Based on DNS Traffic Analysis, Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, October 06-10, 2020, Kharkiv, Ukraine, pp. 171-182. Available: https://ceur-ws.org/Vol-2732/20200171.pdf. [Accessed February 2, 2025].

[13] M. Antonakakis et al., Understanding the Mirai Botnet, Proceedings of the 26th USENIX Security Symposium, August 16–18, 2017, Vancouver, BC, Canada, pp. 1093-1110.

[14] C. Xuan, T. Nikolaevich, N. Dam, N. Hoang and D. Long, "Malicious domain detection based on DNS query using Machine Learning," International Journal of Emerging Trends in Engineering Research, vol. 8, 2020, pp. 1809-1814. doi: 10.30534/ijeter/2020/53852020

[15] BIND 9 - Versatile, classic, complete name server software – ISC – isc.org. https://www.isc.org/bind/ [Accessed January 21, 2025].

[16] S. Ma, T. Pang, R. Cui and D. Yang, A Malicious Domain Detection Method Based on DNS Logs, 4th International Conference on Blockchain Technology and Information Security (ICBCTIS), August 17-19, 2024, Wuhan, China. doi: 10.1109/ICBCTIS64495.2024.00051

[17] M. Stevanovic, J. M. Pedersen, A. D'Alconzo and S. Ruehrup, "A method for identifying compromised clients based on DNS traffic analysis," International Journal of Information Security, vol. 16, pp. 115–132, 2017. doi: 10.1007/s10207-016-0331-3

[18] BIND 9 Administrator Reference Manual – ISC – isc.org. https://bind9.readthedocs.io/en/latest/reference.html#the-category-phrase [Accessed January 24, 2025].

[19] Y. Lut, M. Wang, E. M. Redmiles and R. Cummings, How We Browse: Measurement and Analysis of Browsing Behavior, 2024 IEEE 6th International Conference on Cognitive Machine Intelligence (CogMI), October 28-30, 2024, Washington D.C., USA, pp. 257-264. doi: 10.1109/CogMI62246.2024.00041

[20] D. Dimitrova, "Selection and Justification of Criteria for Comparative Analysis of Lightweight Ciphers," Mathematics and Informatics, vol. LXVI, no. 5, 2023, pp. 534–542. doi: 10.53656/math2023-5-8-sel

[21] Y. Dechev, "Research on the impact of online learning on individual learning styles," Mathematics and informatics, vol. 66, no. 2, 2023, pp. 155-169. doi: 10.53656/math2023-2-5-res

[22] M. Maroun and A. Ivanova, "Ontology-based approach for cybersecurity recruitment," AIP Conf. Proc., vol. 2333, 070014, March 2021. doi: 10.1063/5.0042320

[23] Y. Huang, J. Negrete, A. Wosotowsky, J. Wagener, E. Peterson, A. Rodriguez and C. Fralick, Detect Malicious IP Addresses using Cross-Protocol Analysis, 2019 IEEE Symposium Series on Computational Intelligence (SSCI), December 6-9, 2019, Xiamen, China, pp. 664-672. doi: 10.1109/SSCI44817.2019.9003003

[24] M. Sotirov, V. Petrova and D. Nikolova-Sotirova, Personalized Gamified Education: Feedback Mechanisms and Adaptive Learning Paths, 2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS), December 6-7, 2024, Istanbul, Turkiye, pp. 1-6. doi: 10.1109/ISAS64331.2024.10845384

[25] AbuseIPDB - making the internet safer, one IP at a time - AbuseIPDB LLC - abuseipdb.com. https://www.abuseipdb.com/ [Accessed January 28, 2025].

[26] S. Torabi, A. Boukhtouta, C. Assi and M. Debbabi, "Detecting Internet Abuse by Analyzing Passive DNS Traffic: A Survey of Implemented Systems," IEEE Communications Surveys & Tutorials, Volume 20, Issue 4, Fourthquarter 2018, pp. 3389 – 3415, June 2018. doi: 10.1109/COMST.2018.2849614

[27] W. McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter 3rd Edition. Sebastopol, USA: O'Reilly Media; 2022

[28] openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files. https://openpyxl.readthedocs.io/en/stable/index.html# [Accessed January 28, 2025].

[29] J. Whitington, Python from the Very Beginning: With 100 exercises and answers. Cambridge, UK: Coherent Press, 2020