Fast Flux Network-Based Detection of Malicious DNS Domains

Sucheta Chandra[1], Moutushi Singh[2], Malay Gangopadhyaya[3], Sriparno Chakraborty[1], Ishika Chowdhury[1]

[1]Department of Computer Science & Application, Institute of Engineering & Management (IEM), University of Engineering & Management (UEM).
[2]Department of Information Technology & Computer Science, Institute of Engineering & Management (IEM), University of Engineering & Management (UEM).
[3]Department, Electronics & Communication Engineering, Institute of Engineering & Management (IEM), University of Engineering & Management (UEM).

Contributing authors: sucheta.chandra@iem.edu.in; moutushi.singh@iem.edu.in; malay.ganguly@iem.edu.in; Sriparno.Chakraborty2023@iem.edu.in; Ishika.Chowdhury2023@iem.edu.in;

Abstract

This paper introduces an innovative approach to network security, emphasizing detailed algorithm descriptions, visualization techniques, and real-time monitoring for effective threat detection. The method ensures scalability and adaptability across diverse environments, with threshold-based malicious detection and graph structures enhancing security analysis. By focusing on DNS and fast flux networks, the approach provides a dynamic, real-time analysis of domain-IP relationships, identifying suspicious activity such as frequent IP changes. The integration of graph analysis offers valuable insights into network behavior, enabling timely responses to potential security threats and maintaining a secure network environment.

Keywords: Network Security,DNS,Fast Flux Networks,Graph Analysis

1 Introduction

The Domain Name System, or DNS,[1] is fundamental to the operations of the Internet, as it acts as a directory that transposes human-friendly domain names, such as google.com, into their machine-friendly forms, known as IP addresses. This translation ensures that users have fast access to web pages and services. At the same time, DNS may be used nefariously by certain malicious actors. One of these techniques is Fast Flux Networks [2],[3] in which IP addresses are rapidly rotated under the same domain.

This creates a network of nodes that are in constant flux, making it nearly impossible for security professionals and automated systems to track the source of malicious activity. The nature of fast flux makes it heavily used in most phishing campaigns and the distribution of malwares as well as botnets because it obfuscates where the real server is. They can always beat the defenders tracking or blocking such infrastructure by a constant change in IP addresses. Network graph analysis is one powerful method for attack detection. This is based on modeling relationships between entities such as domains, IP addresses,[4] and other components in the network as a graph. Every node in the graph represents an entity, and the edges connecting them signify relationships. From this relationship, security experts can find the patterns that appear to be anomalous or suspicious, like repeated changes in the IP addresses for a specific domain or nodes having anomalous interconnection. For example, a domain may continue to change IP addresses frequently or have other connections with domains, which is suspicious and thus may indicate it is part of a fast flux network or possibly involved in cybercrime or cyber-attacks. This type of network graph analysis is more useful since it can identify tactics that evade traditional security controls, like fast flux networks, which are dynamic. It will soon allow the analyst to identify the threats and start investigations about them, thus making a faster response to malicious activities and reduced damage from attacks. Network graph analysis can be used by organizations to improve their ability to detect and mitigate complex cyber threats that rely on dynamic, shifting network infrastructures.

2 Literature Survey

The paper "Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features" addresses the challenges of detecting malicious domains used in activities like phishing, malware distribution, and command-and-control operations. It proposes a new method combining DNS data, lexical analysis, blacklisting, and web-based features, utilizing logistic regression to classify domains. The researchers achieve 60% accuracy, suggesting future improvements such as dataset expansion, passive DNS analysis, and multi-classifier models for enhanced detection of malicious activities like spam, phishing, and malware.[1].The paper "DNS dataset for malicious domains detection" presents a dataset of around 9,000 domain names, split between malicious and non-malicious domains, designed for supervised machine learning analysis. It encompasses characteristics of domain name, IP addresses, geolocation, and WHOIS information, and is designed to be used in real-time DNS firewall solutions.[5].The paper "Detection Of Malicious Domain Name Based On DNS Data Analysis" discusses methods for detecting malicious domains, focusing on DNS hijacking, poisoning, and various types of cybersquatting. It brings to light the issues with advanced attack techniques and calls for scalable detection algorithms, such as those using platforms like Apache Hadoop for big data processing.[6].Another paper on the same topic explores active and passive DNS data analysis methods for detecting malicious domains, emphasizing the challenges posed by advanced techniques like Domain-Flux and IP-Flux, and the need for more efficient and standardized detection algorithms.[2].The paper "Countering Malicious Processes with Process-DNS Association" introduces PDNS, an endpoint DNS monitoring system that integrates DNS and process-level features for detecting stealthy attacks. PDNS demonstrated high detection accuracy with low false positives, even in the presence of encrypted DNS traffic.[3] "DNS graph mining for malicious domain detection" introduces a graph-based approach that makes use of DNS data and belief propagation algorithms. This approach gains over 98% accuracy and is scalable for the purpose of detecting malicious domains, even using limited labeled data.[7]. "A Survey of Machine Learning for Computer Architecture and Systems" deals with the concept of machine learning (ML) in computer architecture, which applies to performance modeling, resource management, and hardware design. It indicates issues like big data and

interpretability of ML models in the design of a system. [8].Paper "Discovering Malicious Domains through Passive DNS Data Graph Analysis" focuses on the analysis of global associations between domains through a domain-resolution graph. This method has been shown to have high true positive rates (over 95%) and low false positives, thus providing a more resilient mechanism for detecting malicious domains.[9].The research paper "Identifying Malicious Domains based on Passive DNS Data Graphs Analysis" also uses graph-based inference techniques to detect malicious domains. This technique also attains high true positive rates and low false positives, focusing on global associations to counter evasion strategies.[10].The paper "Monitoring the Initial DNS Behavior of Malicious Domains" discusses early DNS behaviors of malicious domains and postulates that monitoring initial DNS activities may be a worthwhile strategy for early detection.[11].he paper "Detecting Malware Based on DNS Graph Mining" presents a malware detection approach by building a DNS graph and using belief propagation to detect infected nodes. The method produced high accuracy to detect malware hosts and command-and-control servers.[4]. In "Discovering Malicious Domains through Passive DNS Data Graph Analysis" present a scalable approach using global associations between domains to detect malicious domains, supplementing feature-based methods.[12].

3 Proposed methodology

3.1 Data Collection

We begin firstly with resolving the domain names into their respective IP addresses using the 'socket' library. The list of most visited and frequently accessed domains are monitored in real time. Continuous queries on these domains ensure that the system is always updated about the latest mapping between domain names and their resolved IP addresses. This constant surveillance is the basis of the analysis, through which any anomalies or frequent changes in IP resolutions might indicate possible security breaches or attacks.

3.2 Data Pre-processing

The program keeps two important data structures: one to record the resolved and most recent IP addresses of each domain and another that keeps track of the number of changes observed up to a certain point in time. Both these dictionaries serve as real-time references. Whenever an IP of a domain gets re-resolved, they can be compared quickly. In case the IP of a domain changes, the new IP is noted, and the change count for that particular change is incremented. Domains that exceed a predefined threshold of IP changes are flagged for further investigation, as frequent changes might indicate malicious activity or potential domain hijacking.

3.3 Graph Construction

The networks and their resolved IP addresses are represented as nodes in a directed graph, constructed using the NetworkX library. The association between a domain and its corresponding IP is represented as edges in the graph. First, the graph is built mapping each network to its current IP. Over time, as IP changes are detected, the graph evolves dynamically by removing outdated edges and creating new ones that reflect updated mappings. The graph structure then provides a visual and analytical framework for studying relationships and changes in the domain-IP network. A fast flux network is characterized by a domain name rapidly switching between multiple IP addresses, often leveraging a network of compromised machines to obscure malicious activities. In this implementation, the graph construction simulates and detects such behavior by dynamically updating the domain-IP relationships over time.Domains and IP addresses are nodes in a directed graph. A directed edge between a domain node and an IP node means that the domain resolves to the IP. An emulated fast

flux network is formed when one domain node connects to multiple IP nodes, either sequentially or concurrently, to represent the frequent change in the domain's IP resolution.Updates from every identified IP change trigger the graph construction. The old edge connecting the domain to its previous IP is removed and replaced by an edge connecting the same domain to the new updated IP, leading to continuous evolution of the graph structure that eventually creates a visualization representation of the fast flux behavior.This dynamic graph, above all, underscores a simple but defining feature of fast flux networks: frequent, rapid reassignment of IP addresses to a domain. These patterns, when observed, often prompt further scrutiny since they are commonly seen in malicious operations, including phishing attacks, botnets, or malware distribution.

3.4 Graph Analysis Technique

The system monitors the IP change frequency of each domain. If more than a threshold of predefined value changes are monitored within a period of time for the same domain (for instance, 10), it will indicate some malicious activities, including phishing and DNS spoofing, which indicate anomalous behavior. The nature of the graph of adding or deleting an edge visually depicts the anomaly to target suspicious domains to carry out targeted analysis.

3.5 Identification of Malicious Nodes

The domains that haphazardly change their resolved IP addresses with respect to time are marked as possibly malicious and highlighted in the graph. These nodes are colored red and distinguished from the safe domain nodes, which are colored light blue, and IP nodes, which are colored light green. Distinct colors will easily allow problematic nodes to be identified at a glance, allowing for quicker decisions and further investigation into the flagged domains. This identification process plays a crucial role in proactive threat detection and network security.

3.6 Visualization of the Malicious Nodes

NetworkX is used to visualize the graph in a more user-friendly representation of domain-IP relationships. The graph layout is optimized for clarity by positioning nodes using a spring layout algorithm that spreads them out for better visibility. Each node is color-coded according to its type and status: safe domains are light blue, flagged domains are red, and IP nodes are light green. This visual distinction allows users to easily understand the structure and focus on areas of concern, such as domains with high IP change frequencies.

3.7 Result of the Graph Analysis

The real-time analysis of the domain-IP graph provides valuable insights into network behavior. The program dynamically tracks and visualizes changes, identifying domains that exhibit suspicious activity, such as frequent IP changes. The flagged nodes serve as a warning for potential threats, enabling security teams to investigate and respond promptly. The evolving graph provides a wide and continuous view of the landscape of the domain-IP, allowing no suspicious activity to be masked and ensuring maintenance of a secured network environment.

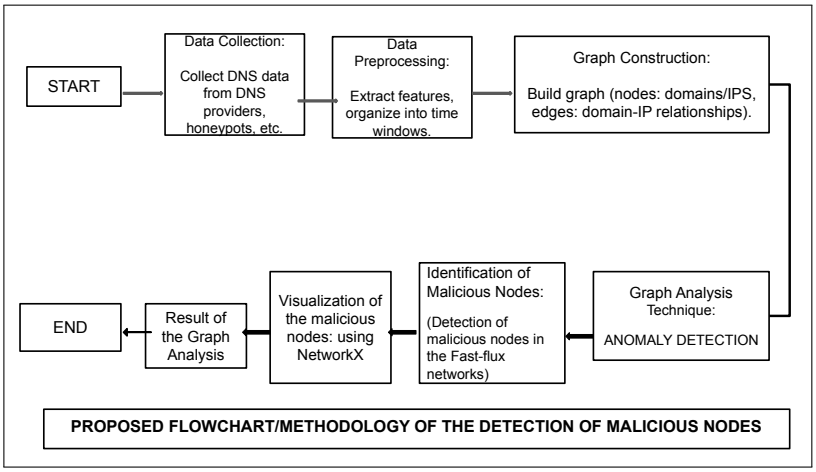

Fig. 1: FLOWCHART OF THE DETECTION OF MALICIOUS NODES

3.8 Algorithm

Algorithm 1 Initialization

Define `get_ip(domain)` to resolve a domain to its IP address
Define the list of domains to monitor:
`domains = [‘‘google.com’’, ‘‘facebook.com’’, ‘‘twitter.com’’, ...]`
Create an empty directed graph `G = nx.DiGraph()`
Initialize `old_ips` dictionary with current IPs of the domains:
`old_ips = {domain: get_ip(domain) for domain in domains}`
Initialize `change_counts` dictionary with zero for each domain:
`change_counts = {domain: 0 for domain in domains}`
Add nodes and edges for each domain and its corresponding IP to the graph:
for each `domain` in `domains` **do**
Add node for the domain `G.add_node(domain, type=‘domain’)`
Add node for the domain's IP `G.add_node(old_ips[domain], type=‘ip’)`
Add edge from domain to IP `G.add_edge(domain, old_ips[domain])`
end for
Define threshold `threshold = 10` for flagging malicious domains based on IP changes

Algorithm 2 Monitoring and Visualization Loop

while True **do**
  Wait for 60 minutes `time.sleep(3600)`
  for each `domain` in `domains` **do**
    Get the current IP address `new_ip = get_ip(domain)`
    if `new_ip != old_ips[domain]` **then**
      Print the IP change: `IP for domain changed from old_ips[domain] to new_ip`
      Update `old_ips[domain]` to `new_ip`
      Increment the change count `change_counts[domain] += 1`
      if `change_counts[domain] > threshold` **then**
        Print a warning: `Warning: domain has changed IPs multiple times. Potentially malicious.`
      end if
      Remove the old edge `G.remove_edge(domain, old_ips[domain])`
      Add a new edge from domain to the new IP `G.add_edge(domain, new_ip)`
    end if
  end for
  Create a plot with size `plt.figure(figsize=(12, 10))`
  Define node positions using `pos = nx.spring_layout(G)`
  Create a color map for nodes based on domain type and change counts:
  for each `node` in `G.nodes(data=True)` **do**
    if `node[1]['type'] == 'domain'` **then**
      if `change_counts[node[0]] > threshold` **then**
        Add `'red'` to the color map for potentially malicious domains
      else
        Add `'lightblue'` to the color map for safe domains
      end if
    else
      Add `'lightgreen'` to the color map for IP nodes
    end if
  end for
  Draw the graph with `nx.draw(G, pos, with_labels=True, node_color=color_map, node_size=1500, edge_color='gray')`
  Set the title of the plot `plt.title("Network Graph of Domains and IP Addresses")`
  Display the plot `plt.show()`
end while

4 Discussion

The proposed methodology offers an innovative approach to detecting malicious domains through continuous domain resolution monitoring and dynamic graph construction, addressing the growing security threats posed by phishing, malware distribution, and botnet activities. By resolving frequently visited domains to their IP addresses using the 'socket' library, the system ensures that it is always updated with the latest mappings, flagging domains with significant fluctuations in their resolved IP addresses. This is a key indicator of potential malicious activity, such as domain hijacking or attempts to obfuscate the true origin of cyber-attacks. The system employs two dictionaries to track the resolved IP addresses and monitor the frequency of IP changes, flagging domains that exceed a predefined threshold for further investigation. Frequent IP switching, which is a characteristic of fast flux networks, is often associated with malicious activities like botnets or phishing schemes. The dynamic graph construction using the NetworkX library is crucial in detecting such behaviors. By modeling domains as nodes and their corresponding IPs as edges in a directed graph,

the system continuously updates the graph structure as IP addresses change over time, thereby identifying fast-flux patterns where a domain rapidly switches between multiple IP addresses to obscure its origin. This technique provides a visual and analytical framework for studying the relationships between domains and IP addresses. When a domain node connects to multiple IP nodes, the system detects fast flux behavior and flags the domain for potential malicious activity. The algorithm also tracks the number of IP changes over time, and suspicious domains are flagged if they exceed a predefined threshold. This is a strong indicator of malicious intent, as it is typical of fast flux networks used to evade detection and carry out attacks. The approach, which incorporates belief propagation algorithms for analyzing DNS data and dynamically updating the graph, ensures that the system can identify anomalous behavior in real time. By identifying domains that exhibit suspicious patterns, such as frequent changes in IP address, and correlating them with known attack vectors, the methodology enhances the ability to detect threats proactively. The real-time nature of the system makes it suitable for integration into cybersecurity infrastructure, providing an effective means of identifying and responding to malicious domains and activities that could otherwise remain hidden in the vastness of the internet.

5 Result

It actually monitors IP addresses over time to determine when domain sets have made changes. Using Python, a script will actually help to flag up any irregular patterns that indicate a possible threat. First, the code detects the respective malicious corresponding IP address, placing the information inside a dictionary; it then prepares a directed graph with 'networkx', the nodes representing the domains or their corresponding IP address.The monitoring process runs in an infinite loop and checks the IP addresses every hour. If an IP address has changed, the graph should update by removing the old edge, which represents the previous IP address, and adding a new edge for the new IP address. The number of changes in IP address for each domain is tracked and, if it exceeds 10, it is considered potentially malicious. This is set up because changes in IPs too frequently may indicate abnormal behavior, such as phishing or other forms of cyberattacks.Such monitoring may therefore detect suspected anomalies and show them against a number of domains associated with unorthodox changes in IPs which might present potential phishing and similar malevolent threats hence helping rapidly spot such problems to take early intervention.

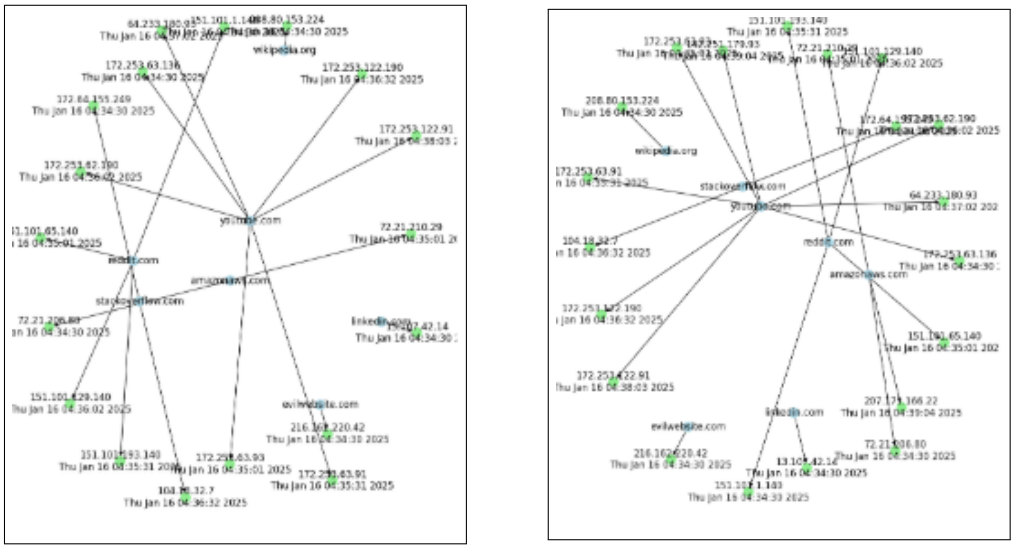

Fig. 2: Formation of Network Graph at Different Time Stamps

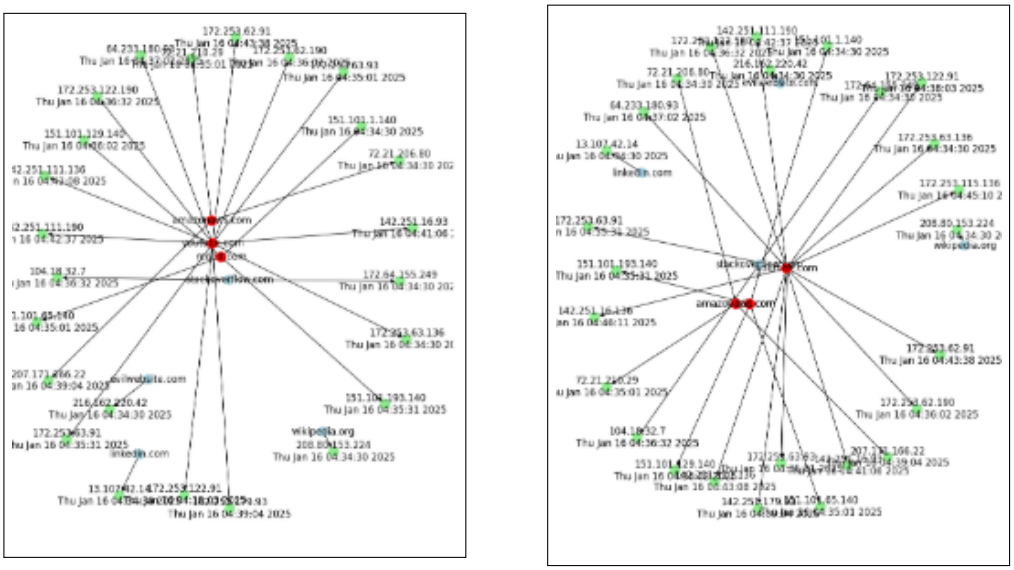

Fig. 3: Detection of Malicious Domain at Different Time Stamp

6 Comparative Analysis

Parameters	**Proposed Paper**	[5]	[2]	[3]	[10]
Presence of Algorithm Description	✓	✓			✓
Use of Visualization Techniques	✓				
Real-Time Monitoring Capability	✓		✓	✓	
Scalability and Adaptability	✓		✓	✓	✓
Threshold-Based Malicious Detection	✓				✓
Integration of Graph Structures	✓				
Fast Flux Detection	✓		✓		✓
Continuous Updates	✓		✓	✓	
Ease of Implementation	✓	✓			✓
Graph Analysis	✓				

Table 1: Comparison of Parameters Across Papers

7 Conclusion and Future Work

The proposed paper will demonstrate a very comprehensive and innovative approach in which it addresses the key parameters important to its effectiveness. It includes detailed descriptions of algorithms, making sure that they are clear and reproducible, and visualization techniques have been added to further facilitate the understanding of the model. Real-time monitoring capabilities make the paper applicable in real life, scalability and adaptability making sure that it works fine in diverse environments. The use of threshold-based malicious detection and the integration of graph structures also very clearly shows that there is a firm commitment towards security and advanced methods of analysis. Overall, this paper provides a well-rounded framework which combines theoretical depth with practical relevance, thus making it applicable to the real world. The future scope of this proposed paper focuses on several promising avenues for further research and development. Enhancements in real-time monitoring capabilities to allow support for ever more dynamic environments may be done such as Internet of Things (IoT) networks or large-scale distributed systems, wherein the speed and

complexity of the data are continually increasing. The second area for improvement may include scaling the algorithm better so that as the data is increasing, the system will still work efficiently and effectively, especially when applied to the cloud and edge computing environment. Moreover, inclusion of machine learning or artificial intelligence models could help in enhancing malicious detection accuracy since the system learns and adapts to new threats without human intervention. Further investigations of hybrid graph structures, introducing temporal or weighted edges could better provide more nuanced insights in the changing nature of security threats. The integration of further sophisticated visualization techniques and dashboards would also improve the user experience and decision-making process in real-time security management.

References

[1] Palaniappan, G., S, S., Rajendran, B., Sanjay, Goyal, S., B S, B.: Malicious domain detection using machine learning on domain name features, host-based features and web-based features. Procedia Computer Science **171**, 654–661 (2020) https://doi.org/10.1016/j.procs.2020.04.071 . Third International Conference on Computing and Network Communications (CoCoNet'19)

[2] Zhang, K., Ji, W., Li, N., Wang, Y., Liao, S.: Detection of malicious domain name based on dns data analysis. Journal of Physics: Conference Series **1544**(1), 012169 (2020) https://doi.org/10.1088/1742-6596/1544/1/012169

[3] Bilge, L., Kirda, E., Balzarotti, D., Kruegel, C., Vigna, G.: Exposure: A passive dns analysis service to detect and report malicious domains. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2011). https://kangkookjee.io/wp-content/uploads/2021/06/pdns-ndss2019.pdf

[4] Zou, F., Zhang, S., Rao, W., Yi, P.: Detecting malware based on dns graph mining. International Journal of Distributed Sensor Networks **11**(10), 102687 (2015) https://doi.org/10.1155/2015/102687 https://doi.org/10.1155/2015/102687

[5] Marques, C., Malta, S., Magalhães, J.P.: Dns dataset for malicious domains detection. Data in Brief **38**, 107342 (2021) https://doi.org/10.1016/j.dib.2021.107342

[6] Aljawarneh, S., Aldwairi, M., Yassein, M.B.: Detection of malicious domain name based on DNS data analysis. ResearchGate (2020)

[7] Tran, H., Nguyen, A., Vo, P., Vu, T.: Dns graph mining for malicious domain detection. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 4680–4685 (2017). https://doi.org/10.1109/BigData.2017.8258515

[8] Jia, J., Dong, Z., Li, J., Stokes, J.W.: Detection of malicious dns and web servers using graph-based approaches. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2625–2629 (2021). https://doi.org/10.1109/ICASSP39728.2021.9415092

[9] Khalil, I., Yu, T., Guan, B.: Discovering malicious domains through passive dns data graph analysis. In: ASIA CCS '16: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2897845.2897877 . https://doi.org/10.1145/2897845.2897877

[10] Perdisci, R., Corona, I., Giacinto, G.: Early detection of malicious flux networks via large-scale passive dns traffic analysis. IEEE Transactions on Dependable and Secure Computing **9**(5), 714–726 (2012)

[11] Hao, S., Feamster, N., Pandrangi, R.: Monitoring the initial dns behavior of malicious domains. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference. IMC '11, pp. 269–278. Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2068816.2068842 . https://doi.org/10.1145/2068816.2068842

[12] M, A., G, S.V., Krishnan, M.R., U, P.S.A., Pal, S., Vazhayil, A., Sridharan, G., Poornachandran, P.: Malicious node identification for dns data using graph convolutional networks. In: 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), vol. 7, pp. 104–109 (2022)