On the Provision of Network-Wide Cyber Situational Awareness via Graph-Based Analytics

Martin Husák[1,2(✉)], Joseph Khoury[2], Đorđe Klisura[2], and Elias Bou-Harb[2]

[1] Institute of Computer Science, Masaryk University, Brno, Czech Republic
husakm@ics.muni.cz
[2] The Cyber Center for Security and Analytics, The University of Texas, San Antonio, TX, USA
{joseph.khoury,dorde.klisura,elias.bouharb}@utsa.edu

Abstract. In this paper, we posit how semi-static (i.e., not changing very often) complex computer network-based intelligence using graph-based analytics can become enablers of Cyber Situational Awareness (CSA) (i.e., perception, comprehension, and projection of situations in a cyber environment). A plethora of newly surfaced cyber security researchers have used graph-based analytics to facilitate particular down tasks in dynamic complex cyber environments. This includes graph-, node- and edge-level detection, classification, and others (e.g., credit card fraudulent transactions as an edge classification problem). To the best of our knowledge, very limited efforts have consolidated the outputs of heterogeneous computer network monitoring and reconnaissance tools (e.g., Nmap) in enabling actionable CSA. As such, in this work, we address this literature gap while describing several use cases of graph traversal, graph measures, and subgraph mining in vulnerability and security state assessment, attack projection and mitigation, and device criticality estimation. We highlight the benefits of the graph-based approaches compared to traditional methods. Finally, we postulate open research and application challenges in graph-based analytics for CSA to prompt promising research directions and operational capabilities.

Keywords: Cyber security · Cyber situational awareness · Graph-based analytics · Large and complex network · Network security management

1 Introduction

Computer networks have become a critical asset in the interconnected world. As such, the sheer number and diverseness of connected devices hinder its security management, deteriorate incident response processes, and ultimately thwart operative Cyber Situational Awareness (CSA) (i.e., situational awareness in

P. Collet et al. (Eds.): CCE 2023, LNCS 13927, pp. 167–179, 2023.
https://doi.org/10.1007/978-3-031-44355-8_12

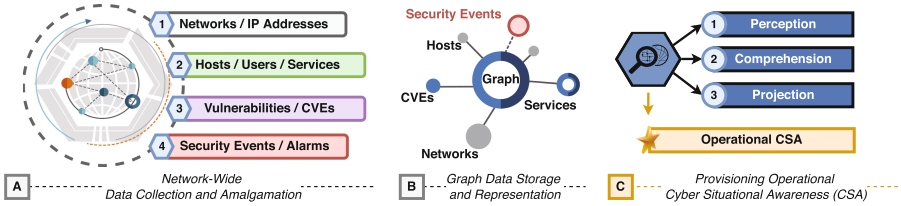

Fig. 1. Provisioning network-wide CSA via graph-based analytics: *(A)* collect and amalgamate network-wide data using heterogeneous tools for computer network monitoring and reconnaissance, *(B)* leverage graph-based analytics to store, visualize, and query the data, *(C)* leverage this data to provision operational CSA for defensive measures, incident responses, and network forensics.

cyberspace). Besides, the constantly changing threat landscape effectively renders network defense a tedious procedure that involves viable CSA coupled with continuous decision-making, actions, and improvements. The concept of CSA defines three levels that need to be achieved to protect the network effectively, *(i)* the `perception` of the elements in the environment within a volume of time and space, *(ii)* the `comprehension` of their meaning, and ultimately *(iii)* the `projection` of their status in the near future [9]. In the context of large and complex networks, graph-based analytics is fairly extensible, can be straightforwardly visualized, and is comprehensible for human analysts, which makes them an excellent choice for the *comprehension* level of CSA. For these reasons, researchers adopted graph-based data representation and storage (e.g., in the form of graph databases) to store data on computer networks, devices, vulnerabilities, and other security-relevant entities.

To that extent, in this paper, we posit the provision of network-wide CSA (and namely its **comprehension** level) via graph-based analytics. Figure 1 illustrates three major steps to achieve this quest. First, in *(A)* we rely on various monitoring and reconnaissance tools (e.g., Nmap [21]) to gather timely information on a computer network and devices (i.e., network/hosts/users/services information, IP addresses, vulnerabilities, Common Vulnerability and Exposures (CVEs), security events). Second, in *(B)* we make use of graph-based analytics (e.g., Neo4j Graph Data Platform [24]) to store and visualize the collected data. Third, in *(C)* we put forward methodical cyber security tasks involving graph-based analytics to achieve operational CSA in practice and ultimately facilitate the preparation of network defenses, planning of preventive actions, and speeding-up incident responses and network forensics.

The remainder of this paper is organized as follows. Section 2 presents background information and summarizes related work. Section 3 presents a selected set of imperative cyber security tasks using graph-based analytics. Section 4 discusses the open issues and formulates the challenges for future work. Section 5 concludes the paper.

2 Background Information and Related Work

Whilst, graph-based analytics are very well known to the cyber security community [1], yet, very limited efforts have been put in the context of semi-static graphs, i.e., graphs where new information is sporadic or recurrent with little to no obvious pattern changes. Attack graphs are one example that has been used for decades to model cyber-attacks and calculate their impact [17,23]). Accordingly, we primarily investigate in this work graph-based analytics for cyber security tasks in the context of semi-static graphs.

The primacy of identifying critical nodes/threats in a network necessitates the usage of advanced graph-based analytics including centrality algorithms (i.e., degree, betweenness, *PageRank*, and closeness). Degree centrality is used to count the edges that connect a node to others [3], while closeness centrality gauges a node's typical separation from every other node in the network. Moreover, betweenness centrality constitutes the extent to which a node lies on the shortest paths between all pairs of nodes in a network [5]. Furthermore, the *PageRank* algorithm ranks nodes according to their significance and is effective in discovering critical nodes in a network [7].

2.1 Tooling and Perception of the Cyber Environment

With the emergence of lateral movement and attacks targeting whole networks, there was a need to grasp complex heterogeneous data on computer networks in a single, comprehensive database. The conceptual works like Cauldron [15] enabled to keep track of hosts, services, users, security events, and other entities in a single database. CyGraph [25] became a well-known implementation of the graph-based approach for cyber situational awareness. CRUSOE [12] is a recently published toolset inspired by CyGraph but based on empirical data provided by common tools instead of perfecting the data structure and analysis.

Such network-wide graphs allow for assessing risks to the organization operating the network, optimizing the network defenses, or facilitating incident response. However, it is still tedious to fill the whole database with exact data, which would allow using all the analytical features. For example, the CRUSOE [12] uses common tools to autonomously monitor the network traffic, actively scan devices in the network, fingerprint running hosts and services, and disclose vulnerabilities. Such data were periodically updated and stored in a joint database along with static information on the network segmentation, organization structure, details on vulnerabilities from external sources, history of cyber security incidents, and other local knowledge.

2.2 Comprehension and Knowledge Building

The graph databases, i.e., graph-oriented NoSQL database management systems, such as Neo4j [24] and specialized graph-querying languages like Cypher and Gremlin and natural choices for storing and querying large graph data.

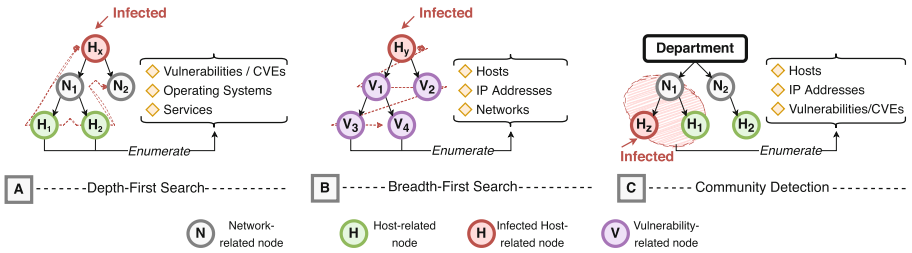

Fig. 2. Graph Traversal and Community Detection Algorithms: *(A)* depth-first search, *(B)* breadth-first search, *(C)* community detection algorithms applied to selected cyber security tasks.

Nevertheless, there is a need to structure them. CRUSOE data model [19] is an example of such a model, independent of the underlying technology.

An interesting observation is that the data about the network stored in a graph are, in essence, knowledge graphs. Likewise, the graph-based data models of CyGraph or CRUSOE can be considered ontologies of the domain. Knowledge graphs allow for reasoning over the data and are the subject of research on artificial intelligence [6,28]. The ontologies allow for understanding the domain and categorizing various attack types and defense options [13,30].

2.3 Existing Surveys

Readers further interested in the outlined topics are kindly referred to several surveys. The earliest one by Akoglu et al. [1] surveyed graph-based techniques for anomaly detection in diverse domains, including network traffic analysis. The attack graphs and their construction were exhaustively covered by Kaynar et al. [17]. The application of graphs in network-wide situational awareness was covered by Noel et al. [26]. Bowman and Huang [4] reviewed the challenges of the application of Graph AI in cyber security. Atzmüller and Kanawati [2] provided an overview of explainability for complex network analysis in cyber security. In the sequel, we discuss selected cyber security tasks using graph-based analytics.

3 Selected Cyber Security Tasks Using Graph-Based Analytics

In this section, we present a selected set of imperative cyber security tasks that is amenable to graph-based analytics. Specifically, we highlight the advantages of these approaches in the context of complex network-related intelligence to achieve operational CSA, namely incident comprehension, prevention, and response. The selected use cases coupled with the graph-based analytic solutions provide a one-way example of addressing these cyber security tasks, particularly, we tailor these instances to the CRUSOE data model [12,19]. Readers are kindly referred to the related work for exact specification of the data and graph construction.

3.1 Finding Similar Hosts in Close Proximity by Graph Traversal

Graph traversal is a simple but efficient method for analyzing graph data, including for cyber security purposes. In a previous work [11], we elaborated on a typical situation in incident response. Let's assume a user reports a machine infected with ransomware. Ransomware infections can spread rapidly and can cause significant harm to the organization, so it is of utmost importance to mitigate it fast. We may start a graph traversal from the node representing the infected device to look up for similar devices in close proximity that are exposed to the infection. Since it may not be clear what type of ransomware infected the device and if it spreads autonomously or via infected files in emails or data storage, we may simultaneously look up devices in the same subnet or location, within the same department, or used by the same user.

Following Fig. 2 *(A)*, we run the Depth-First Search (DFS) algorithm from the infected node (host) H_x. The query looks-up nodes representing other devices within the same network N_1. A constraint on the length of the path between the source and destination node or types of nodes and edges to traverse can be applied. The found devices H_1 and H_2 are then scored by their similarity to the infected one using the enumerated intelligence such as OS and service fingerprint or common history of security incidents [11].

The described query enables fast early warning to users and administrators of devices immediately threatened by ransomware, which may prevent further infections. Later on, when the forensic analysis of the malware returns how it spread, the results can be filtered to include only those with paths related to the attack vector. An alternative use for this approach is network forensics. The recommendation of similar devices in close proximity can direct the investigators in the analysis of an attack involving lateral movement, i.e., an attack technique involving a breach of a third machine to get better access to the actual target.

3.2 Vulnerable Asset Discovery via Graph Traversal

One of the key motivating use cases of the CRUSOE toolset was large-scale vulnerability assessment [12]. The devices in the network are fingerprinted by Nmap or other common tools that generate output in the form of CPE strings, a structured identification of the system's vendor, major and minor version, patches, or edition. The same CPE strings can also be found in vulnerability databases to enumerate vulnerable systems or their specific configurations. Thus, there exists a mapping between a description of a vulnerability and a fingerprint of a device, which can be used to infer which and how many devices in the network are vulnerable.

For example, in CRUSOE, the relation can be represented as a path consisting of an edge between a *Host* and *Fingerprint* and *Fingerprint* and *CVE*, where CVE is a common vulnerability identifier. The nodes and edges in the CRUSOE graph are inserted automatically by tools periodically checking the vulnerability databases and network scanning tools. Figure 2 *(B)* depicts the use of the Breadth-First Search (BFS) algorithm to query and list potential vulnerabilities

and CVEs, for instance, V_1, V_2, V_3, and V_4 associated with a recent infection on H_y. By extending the query, one can enumerate how many vulnerable hosts are there in each subnet or under the control of a specific administrator. Such a summary was found to be one of the most valuable features of CRUSOE by practitioners [12]. The advantage of the graph-based approach here is the very low complexity of inserting new connections and queries as simple as enumerating neighbours of the neighbouring node.

An elegant graph-based matching of vulnerable configuration was proposed by Tovarňák et al. [31]. Matching the CPE strings of vulnerabilities and device fingerprints are usually done via a brute-force approach. However, doing so on a large scale calls for more efficient algorithms. The authors decompose the CPE strings into a graph model and provide a query to find all matches between vulnerable CVEs and asset configurations in a single graph traversal.

3.3 Network Segmentation via Community Detection and FSM

Graph-based analytics can also be used to implement network segmentation and an important strategy to isolate malicious entities from a graph network and ultimately restrict the propagation of potential security threats [33]. Figure 2 *(C)* depicts a community detection algorithm, namely, the Girvan-Newman algorithm which is used herein to accomplish network segmentation [8]. Specifically, once a community has been identified with an infected host H_z, its corresponding network N_1 can be isolated by deploying firewalls (or fortifying existing ones) to prevent communication with other communities. As such, this approach can isolate potential security risks and prevent them from spreading throughout the entire network. Additionally, enumerating such communities can help identify and characterize specific vulnerabilities and CVEs associated with the infections.

Furthermore, Frequent Subgraph Mining (FSM) is a subfield of graph mining that can offer additional capabilities for cyber security tasks by identifying frequently occurring patterns in a graph using DFS and BFS algorithms [16]. The application of FSM to graph-based network data offers a wide range of cybersecurity benefits. We might discover, for instance, that a certain set of nodes and edges occurs more frequently than we would anticipate by chance, such as a subgraph representing a collection of devices connecting to a particular server or using a particular resource. Then, by examining these subgraphs, we could look for patterns or motifs that would point to malicious behavior. For instance, if it turns out that a specific subgraph is linked to well-known malware or attack vectors, we may utilize this knowledge to create more specialized detection and prevention strategies, such as adding firewall rules to restrict access to particular servers or resources.

3.4 Node Criticality Estimation via Graph Centrality

A motivating example for using graph centrality measures is a frequent question of network security management - *How important is a particular machine for the organization?* Answering such a question properly requires a knowledge of the

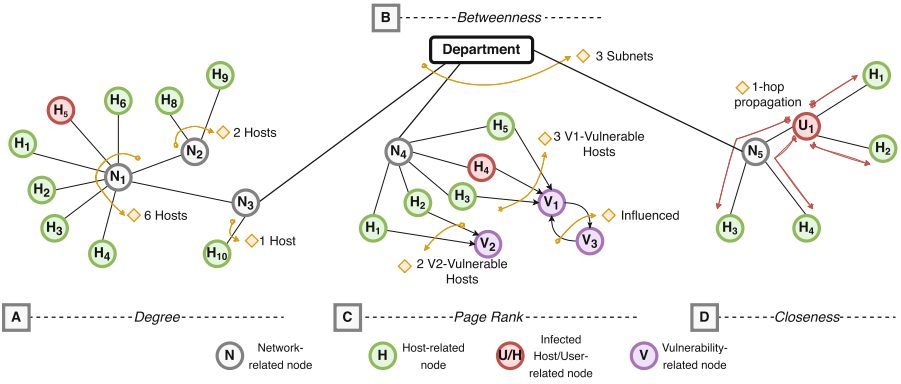

Fig. 3. Graph Centrality Algorithms: *(A)* degree, *(B)* betweenness, *(C)* page rank, *(D)* closeness algorithms applied to specific cyber security tasks.

local environment, which might not be available in large networks and organizations [34]. However, approximations via network measurement help in quickly assessing the importance of a node (device) and aid in identifying a node's role within the network and how that role impacts the network as a whole [20].

To build the network topology graph, multiple hosts in the network use a scanning tool like *Nmap* with the traceroute option to scan the network [21]. Each scan provides a tree structure, with the scanning machine as the host, and these trees are merged into one graph. The more observation points are used, the more is the resulting graph similar to the actual network topology.

The resulting graph can be subjected to a number of centrality measure techniques for understanding the role and impact of different nodes on the network. In the following, we describe four of these techniques, presented in Fig. 3.

To understand the dynamics of a network and select high-value targets, we can identify significant nodes using the degree centrality technique. Nodes with high centrality degrees are crucial to the network's operation and may be potential attack targets [3]. In Fig. 3 *(A)*, we show a scenario where the host H_5 is infected. In this case, we must first isolate the node N_1 as it has the highest centrality degree, that is, is associated with the largest number of nodes (hosts). This will enhance the network security because if N_1 gets compromised, the malicious activity might easily propagate to other nodes and networks in the subnet, all way to the department.

Furthermore, we employ a betweenness centrality approach to find nodes critical to information flow or attack paths via the network [5]. We determine the shortest path between every pair of nodes in the network and subsequently assign a score to each node based on the number of shortest pathways that traverse through it. In Fig. 3 *(B)*, the node *Department* has the highest betweenness centrality as it serves as a bridge between the subnets N_3, N_4, and N_5.

We may also employ the *PageRank* algorithm to measure a node's influence in a network based on the quality of its connections [7]. Influential nodes, which

have many incoming links, also share some of their influence with the nodes they are connected to. As a result, *PageRank* can identify nodes with a broader impact than just their direct connections.

In Fig. 3 *(C)*, nodes V_1, V_2 and V_3 represent certain vulnerabilities V_1, V_2 and V_3. We mark node H_4 to be infected we note it has a vulnerability V_1. Then we note that nodes H_3 and H_5 are nodes with the same vulnerability V_1. Using the *PageRank* algorithm we mark influenced node V_3 to be a vulnerability that is similar to V_1. If in such an environment attacker was able to exploit vulnerability V_1, there is a high probability of exploiting V_3 as well. That being said, in the case of compromised node H_4, it is necessary to check all the nodes that share vulnerability V_1 and then nodes that share vulnerability V_3, and secure them.

Finally, we employ closeness centrality to identify potentially vulnerable systems that are in close proximity to a noteworthy threat. As such, Fig. 3*(D)* demonstrates the relevance of closeness centrality in the context of a graph-based network. Specifically, we mark U_1 to be a user node with the highest closeness centrality. Let's say the user opens a fishing email on its local machine U_1 and the machine gets compromised. Then, the malicious propagation can be the fastest to other nodes within the subnet N_5, as it only takes the attacker one hop from U_1 to compromise all other nodes in N_5.

4 Open Issues and Challenges

Herein, we formulate open issues and research challenges that we face in the development and deployment of the tools enabling us to view the network security properties as a complex network.

4.1 Need to Learn a New Paradigm

It is vital to remind that the graph-based or complex network-based view on network security is a novel paradigm for many cybersecurity experts. Just like with the emergence of stream-based data analysis in the past decade, the new paradigm enabled novel views on the problems and ways to resolve them. However, the practitioners had to adopt the principles of stream-based data analytics to make full use of them.

We have already observed a generally very positive attitude towards the graph-based representation of cyber security data [12]; the graph-based visualization is highly comprehensive even without any background knowledge. However, the adoption of new query languages and data processing paradigms can be slow, and the embracement of the methods of graph theory may be even slower among practitioners.

4.2 Dataset

A lack of well-known, high-quality datasets is a common issue of cyber security research. The existing datasets contain mostly network traffic or attack traces,

and only a few approach the network defense [22]. A few examples of a dataset created for the needs of CSA research include the MM-TBM [14] and CYSAS-S3 [22]. Nevertheless, the first one contains a simple network topology but focuses on attack traces, while the later one is focused on mission-oriented situational awareness and decision-making rather than the technological background. As far as we know, there is no dataset for preventive or defensive network-wide situational awareness. The closest are the network topologies and scenarios for various cyber competitions [32].

We aim to generate a dataset stored as a graph in our future work. The most viable option at the moment seems to be a graph representation of an existing network topology (e.g., graph storage using combinatorial embedding [18]). Another option includes existing tools such as [12,25] to collect data on a live network. However, such data would require proper anonymization of all entries (e.g., IP addresses, domain names, department names, contacts on users and administrators). Anonymizing the most critical or important nodes in the network would require extreme caution. There is a high risk that any omission would allow the de-anonymizing of the whole dataset and possibly compromise the network in which the dataset was collected.

4.3 Unified Ontology

Since the graph-based representation of the network also serves as a knowledge graph for network defense and the graph-based data models as ontologies, it is important to establish a common ontology to facilitate knowledge transfer and research collaboration and enable the integration of tools. Unfortunately, a mature unified technology in the cyber defense domain is not yet available and widely adopted despite significant research efforts.

So far, the relevant ontologies target specific applications like vulnerability management [29] and cyber threat intelligence [28]. Attempts to develop a unified ontology exist in the form of UCO [30] or STUCCO [13], but have yet to achieve a wider spread in the application domain and remain a research topic. Data models used in CyGraph [25] or CRUSOE [19] are the closest to the topics of this manuscript but are not used as stand-alone ontology outside of the tools they support. Thus, there is still an open call for an ontology.

4.4 Application of Graph Neural Networks and Graph AI

Machine learning is, without a doubt, a significant driving force in research on data analysis, regardless of domain. The graph-based data can be processed by a class of machine learning approaches referred to as graph neural networks (GNNs) or Graph AI. We see an emerging trend of the application of GNNs for intrusion detection [27], vulnerability assessment [10] or reasoning over knowledge graphs [6]. However, applications in the area of CSA are lacking.

At the moment, there is a need to identify promising GNN-based approaches applicable to the available data. Even though the complex graphs representing computer networks contain up to tens of thousands of nodes, the nodes are of

certain types and there might not be enough nodes of each type or subgraphs to train the GNNs or Graph AI for a particular purpose. A great benefit would be the use of distributed or federated learning techniques to train the models in multiple networks simultaneously. The privacy considerations of network security management call for such an approach anyway. The graphs are rather static, so the trained models would also not be obsolete that fast as it happens in network traffic analysis, threat intelligence, or other cyber security applications [4]. Nevertheless, explainability remains an open issue [2].

The link prediction, i.e., predicting which entities (nodes) will create a relation (edge), is a widely-used technique worth mentioning here. It suits dynamic graphs, which change rapidly over time. The more static graph representing the computer networks does not offer enough opportunities to observe dynamic changes and to train the ML-based models for link prediction. Therefore, even though we can imagine link prediction for rapid assessment of a newly observed device or vulnerability, we argue such approaches are more suitable for intrusion detection or network traffic analysis.

5 Conclusion

In this paper, we outlined the application of graph-based representation of the data, namely in the form of complex networks, in cyber security with a special focus on cyber defense and cyber situational awareness. In a series of use cases, we illustrated how can we achieve a deeper understanding of a cyber security situation in a network via selected graph algorithms and approaches used in complex network analysis. We illustrated how to provide vulnerability or security state assessment using simple graph traversal and use the results for attack projection and mitigation. Graph centrality measures, link prediction, and subgraph mining were shown to be applicable in advanced security assessment, such as device criticality estimation, prediction of its security state or belonging to a community of common attack targets.

Moreover, we identified several open issues and challenges we may face in future research and development and transferring the research into practice. Namely, we identified the lack of datasets and unified ontology and obstacles practitioners might face when embracing a novel paradigm. On the contrary, we see great potential in the application of graph neural networks in this domain. The open issues will be the subject of our future work, alongside further research and development and empirical evaluation and analysis of the outlined approaches to the presented use cases.

Acknowledgments. This research was supported by OP JAK "MSCAfellow5_MUNI" (No. CZ.02.01.01/00/22_010/0003229).

Scientific Validation. This paper has benefited from the remarks of the following reviewers:
- Pierre Parrend, EPITA Strasbourg, France
- Sofiane Lagraa, Fujitsu, Luxembourg
- Nidà Meddouri, LRE, EPITA, Kremlin-Bicêtre, France
The conference organisers wish to thank them for their highly appreciated effort and contribution.

References

1. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Disc. **29**(3), 626–688 (2014)
2. Atzmueller, M., Kanawati, R.: Explainability in cyber security using complex network analysis: a brief methodological overview. In: Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference. EICC '22, pp. 49–52. ACM (2022)
3. Bavelas, A.: Communication patterns in task-oriented groups. J. Acoust. Soc. Am. **22**(6), 725–730 (1950)
4. Bowman, B., Huang, H.H.: Towards next-generation cybersecurity with graph AI. SIGOPS Oper. Syst. Rev. **55**(1), 61–67 (2021)
5. Brandes, U.: A faster algorithm for betweenness centrality. J. Math. Sociol. **25**(2), 163–177 (2001)
6. Dasgupta, S., Piplai, A., Ranade, P., Joshi, A.: Cybersecurity knowledge graph improvement with graph neural networks. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 3290–3297 (2021)
7. De, S., Sodhi, R.: A PMU assisted cyber attack resilient framework against power systems structural vulnerabilities. Elect. Power Syst. Res. **206**, 107805 (2022)
8. Despalatović, L., Vojković, T., Vukicević, D.: Community structure in networks: Girvan-Newman algorithm improvement. In: 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 997–1002. IEEE (2014)
9. Endsley, M.R.: Situation awareness global assessment technique (SAGAT). In: Aerospace and Electronics Conference, 1988. NAECON 1988, Proceedings of the IEEE 1988 National, pp. 789–795. IEEE (1988)
10. He, H., Ji, Y., Huang, H.H.: Illuminati: towards explaining graph neural networks for cybersecurity analysis. In: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), pp. 74–89 (2022)
11. Husák, M.: Towards a data-driven recommender system for handling ransomware and similar incidents. In: 2021 IEEE International Conference on Intelligence and Security Informatics (ISI) (2021)
12. Husák, M., Sadlek, L., Špaček, S., Laštovička, M., Javorník, M., Komárková, J.: CRUSOE: a toolset for cyber situational awareness and decision support in incident handling. Comput. Secur. **115**, 102609 (2022)
13. Iannacone, M., et al.: Developing an ontology for cyber security knowledge graphs. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference. CISR 2015. ACM (2015)

14. Ioannou, G., Louvieris, P., Clewley, N.: MM-TBM evaluation datasets (2018). https://dx.doi.org/10.21227/8dt8-gx46, iEEE Dataport
15. Jajodia, S., Noel, S., Kalapa, P., Albanese, M., Williams, J.: Cauldron mission-centric cyber situational awareness with defense in depth. In: 2011 - MILCOM 2011 Military Communications Conference, pp. 1339–1344 (2011)
16. Jiang, C., Coenen, F., Zito, M.: A survey of frequent subgraph mining algorithms. Knowl. Eng. Rev. **28**(1), 75–105 (2013)
17. Kaynar, K.: A taxonomy for attack graph generation and usage in network security. J. Inf. Secur. Appl. **29**, 27–56 (2016)
18. Klisura, Đ.: Embedding non-planar graphs: storage and representation. In: Proceedings of the 2021 7th Student Computer Science Research Conference, p. 57 (2021)
19. Komárková, J., Husák, M., Laštovička, M., Tovarňák, D.: CRUSOE: data model for cyber situational awareness. In: Proceedings of the 13th International Conference on Availability, Reliability and Security. ARES 2018. ACM (2018)
20. Laštovička, M., Čeleda, P.: Situational awareness: detecting critical dependencies and devices in a network. In: Tuncer, D., Koch, R., Badonnel, R., Stiller, B. (eds.) AIMS 2017. LNCS, vol. 10356, pp. 173–178. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60774-0_17
21. Lyon, G.F.: Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure, Com LLC (US) (2008)
22. Medenou, R.D., et al.: CYSAS-S3: a novel dataset for validating cyber situational awareness related tools for supporting military operations. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. ARES 2020. ACM (2020)
23. Nassar, M., Khoury, J., Erradi, A., Bou-Harb, E.: Game theoretical model for cybersecurity risk assessment of industrial control systems. In: 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–7. IEEE (2021)
24. Neo4j Inc: Neo4J Graph Data Platform (2023). https://neo4j.com. Accessed 21 Feb 2023
25. Noel, S., Harley, E., Tam, K.H., Limiero, M., Share, M.: CyGraph: graph-based analytics and visualization for cybersecurity. In: Handbook of Statistics, vol. 35, pp. 117–167. Elsevier (2016)
26. Noel, S.: A review of graph approaches to network security analytics. In: Samarati, P., Ray, I., Ray, I. (eds.) From Database to Cyber Security. LNCS, vol. 11170, pp. 300–323. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04834-1_16
27. Pujol-Perich, D., Suarez-Varela, J., Cabellos-Aparicio, A., Barlet-Ros, P.: Unveiling the potential of graph neural networks for robust intrusion detection. SIGMETRICS Perform. Eval. Rev. **49**(4), 111–117 (2022)
28. Sarhan, I., Spruit, M.: Open-cykg: an open cyber threat intelligence knowledge graph. Knowl.-Based Syst. **233**, 107524 (2021)
29. Syed, R.: Cybersecurity vulnerability management: a conceptual ontology and cyber intelligence alert system. Inf. Manag. **57**(6), 103334 (2020)
30. Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A.: UCO: a unified cybersecurity ontology. UMBC Student Collection (2016)
31. Tovarňák, D., Sadlek, L., Čeleda, P.: Graph-based CPE matching for identification of vulnerable asset configurations. In: 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 986–991 (2021)
32. Tovarňák, D., Špaček, S., Vykopal, J.: Traffic and log data captured during a cyber defense exercise. Data Brief **31**, 105784 (2020)

33. Wagner, N., et al.: Towards automated cyber decision support: a case study on network segmentation for security. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–10. IEEE (2016)
34. Zand, A., Houmansadr, A., Vigna, G., Kemmerer, R., Kruegel, C.: Know your Achilles' heel: automatic detection of network critical services. In: Proceedings of the 31st Annual Computer Security Applications Conference. ACSAC 2015, pp. 41–50. ACM (2015)