Hierarchical Modeling of Cyber Assets in Kill Chain Attack Graphs

Lukáš Sadlek, Martin Husák, Pavel Čeleda
Masaryk University, Brno, Czech Republic
sadlek@mail.muni.cz, husakm@ics.muni.cz, celeda@fi.muni.cz

***Abstract*—Cyber threat modeling is a proactive method for identifying possible cyber attacks on network infrastructure that has a wide range of applications in security assessment, risk analysis, and threat exposure management. Popular modeling methods are kill chains and attack graphs. Kill chains divide attacks into phases, and attack graphs depict attack paths. A difficult issue is how to hierarchically model categories of cyber assets that should be used in threat models due to the variety of cyber systems in the current networks. This task should be addressed to provide automation of realistic threat modeling and interoperability with public knowledge bases, such as MITRE ATT&CK. In this paper, we propose a hierarchical modeling methodology for representing cyber assets in kill chain attack graphs. We illustrate its practical application on MITRE D3FEND's Digital Artifact Ontology. Moreover, we define how cyber assets with related attack techniques should be transformed into logical facts and attack rules. We implemented proof-of-concept software modules that can process data obtained from network and host-based monitoring together with attack rules to generate attack graphs. We evaluated the approach with data from a cyber exercise captured in a network of a digital twin organization. The results show that the approach is applicable in real-world networks and can reveal ground-truth attacks.**

***Keywords*—attack graph, kill chain, cyber threat scenario, MITRE ATT&CK, MITRE D3FEND**

I. Introduction

The need for automated approaches that can analyze the security of cyber systems and predict what cyber threats could be used by attackers increases with the ever-increasing complexity and variety of such systems. This fact was emphasized, e.g., by Gartner, which claims that continuous threat exposure management will help organizations decrease their probability of experiencing data breaches by three times by 2026 [1].

Possible attack paths that occur in the network are depicted by attack graphs. According to one of the first definitions [2], they contain vertices representing control of assets by the attacker's influence and directed edges representing attack steps. Their automated generation requires defining preconditions and results of each attack step. However, it is complicated to maintain the consistency of numerous preconditions and results, including the right level of detail. Moreover, researchers often focus on specific attack techniques according to their coverage [3]. Therefore, prerequisites are considered one of the research challenges in the automated generation of attack graphs [4]. We need a modeling methodology that defines how to create preconditions and results in a uniform way.

The previous work created a methodology for generating kill chain attack graphs (KCAGs) [5]. However, it supported mainly attack techniques from MITRE ATT&CK and used only selected cyber assets. In this paper, we aim to extend it. We focus on two research questions:

1) *Can we systematically express hierarchical categories of cyber assets in kill chain attack graphs?*
2) *Can we apply the automated generation of kill chain attack graphs on realistic data from a digital twin network?*

Our contribution consists of extending KCAGs to support trees of cyber asset categories. We define how attack techniques should be mapped to cyber assets and used in rules for generating KCAGs. The generation of graphs allows adjusting levels of details about cyber assets based on their hierarchies, which saves effort in creating attack rules. We evaluated the approach on realistic data from a cyber exercise.

This paper is divided into six sections. Section II describes related work of cyber threat modeling and relevant research. The hierarchical modeling of cyber assets is proposed in Section III. Section IV describes the generation of KCAGs based on asset hierarchies. Section V provides the results of the evaluation for data from a cyber exercise. Section VI concludes the paper.

II. Background and Related Work

The attack graphs are a well-known attack modeling technique that has been used for over two decades [2]. *Kaynar et al.* [6] surveyed attack graph generation and usage. Their use cases include network hardening, security assessment, and metrics. A widely known tool for attack graph generation is MulVAL, which uses logic programming to specify facts and rules for generating attack graphs [7]. Its algorithm has polynomial time complexity, which complicates scalability and generation of attack graphs over large infrastructures. Therefore, scalability is considered a research challenge [4].

Significant efforts were recently put into building knowledge bases for cybersecurity, most notably by the MITRE Corporation. MITRE ATT&CK is a knowledge base of adversarial tactics, techniques, and common knowledge [8] and is widely used in the literature for cyber risk assessment. See, for example, the work of *Ahmed et al.* [9]. MITRE D3FEND is a knowledge graph of possible countermeasures in cybersecurity and provides Digital Artifact Ontology (DAO) that connects adversarial techniques and countermeasures using its entities [10]. The D3FEND matrix divides countermeasures according to types of actions into groups, e.g., model, harden, and isolate. Another notable framework is the Cyber Defense Matrix (CDM) [11].

978-3-903176-66-9

CDM helps organize cybersecurity technologies by putting them into the matrix, where one axis covers various assets (devices, apps, networks, data, users) and the other covers functions (identify, protect, detect, respond, recover), thus clarifying which areas are covered.

Our paper proposes an implementation using operationally relevant input data sources, namely Syslog and IPFIX. IP Flow Information Export (IPFIX) is a protocol for the transmission of IP flows representing network-based data [12]. IP flows are sets of packets with the same properties (e.g., source and destination IP address, source and destination port, and transport protocol) passing through an observation point during a specific time window. Host-based monitoring uses log events. An advantage is the use of some common log formats, such as the widely-used Syslog [13], which is standard logging for Unix-based systems and commonly found on other operating systems and network devices. The Syslog messages typically contain a timestamp, originator process ID, domain name or IP address, and the message, and can use several levels of severity.

In a closely related work from 2018, *Johnson et al.* [14] introduced Meta Attack language (MAL) for modeling cyber threats and attack simulation. It can process the representation of domain entities in class diagrams. It provides a strong expressing ability for cyber threat modeling, but it does not check that defined cyber threats cannot form sequences that violate the ordering of kill chain phases. *Xiong et al.* [15] introduced a domain-specific language called enterpriseLang based on MAL. However, since our focus exceeds the enterprise domain, enterpriseLang misses important entities, such as credentials, files, and processes. *Gylling et al.* [16] proposed the integration of cyber threat intelligence feeds into attack defense graphs (ADGs), thus enriching the ADGs and strengthening the infrastructures against current adversarial methods.

Recently, *Zenitani* [17] pointed out that there is not much information about how prerequisites and results of attack techniques should be described. In our opinion, this is caused by the common way of applying attack graphs. They are applied mainly for ad-hoc use cases, such as security assessment. For these use cases, researchers define some set of rules that are necessary to consider. According to *Tayouri et al.* [3], who surveyed extensions and defined rules for generating attack graphs using the MulVAL generator, less than a quarter of ATT&CK techniques were expressed.

KCAGs combine the advantages of kill chain models and attack graphs [5]. KCAGs contain five types of vertices – levels of privileges, attack techniques, asset properties, countermeasures, and attack goals. Attack techniques were populated with ATT&CK techniques and subtechniques, divided according to violated security properties.

III. Modeling of Cyber Assets

In this paper, we extend the methodology from [5] with the possibility of modeling hierarchies of cyber assets. The KCAGs contain two types of vertices containing cyber assets – *levels of asset control* and *attack goals*. Since attack goals are levels of control appearing at the end of attack paths, we use *levels of asset control* or *level vertices* in the following text.

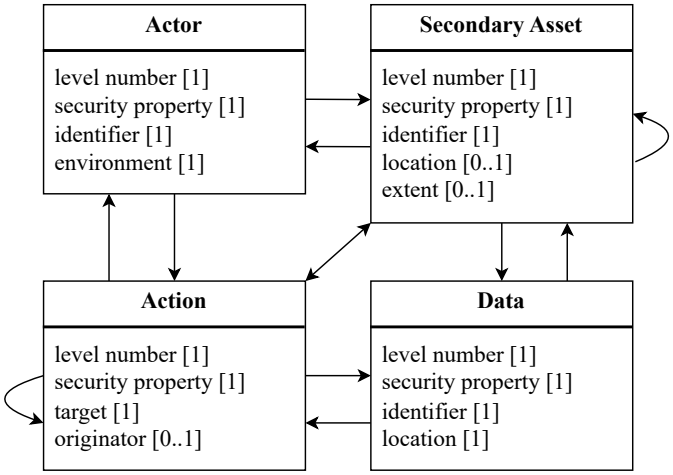

Fig. 1. Asset types allowed as level vertices. Attack techniques can have only prerequisite and result vertices connected by arrows.

Assets are divided into four categories – *actors*, *secondary assets*, *actions*, and *data* [5]. Data includes all categories of data – in transit, at rest, and in use. Secondary assets are mainly technologies supporting the primary organization's assets – missions. Examples are networks, devices, operating systems, and applications. Actors should be humans, their accounts, and non-person entities that do not belong to secondary assets, e.g., organizations. Actions can be system processes, implemented functions, and communication between entities. They should not be malicious actions expressed by attack techniques.

The categories should contain a numeric specification of level and security property [5]. They should have some identifier that distinguishes instances of these classes (see Figure 1). For example, a filename is an identifier for files, and a triple (hostname, protocol, and port) is an identifier for network services. However, actions do not have identifiers since process IDs are not necessary in KCAGs. Instead, the non-compulsory originator of action is needed, e.g., username for authentication.

In a similar way, environment, location, and target aim to express the same pieces of information adjusted to individual asset types. In the case of data, location determines where the data is stored, processed, or transmitted to. The attributes allow copying pieces of information from one to a subsequent level of asset control in KCAGs, e.g., an actor can be an originator of the following action. The secondary assets also contain a non-compulsory extent that further refines the scope of breached security properties, e.g., user level of privileges.

Each of the four classes of assets in Figure 1 should have its hierarchy of inheriting classes. It can be based on a comprehensive ontology for a cybersecurity domain. In our case, we used a subset of entities from the MITRE D3FEND's DAO [10], which we consider exhaustive enough. Considered taxonomies with their cyber assets and categories are listed in Table I. Figure 2 contains a subset of entities expressed graphically.

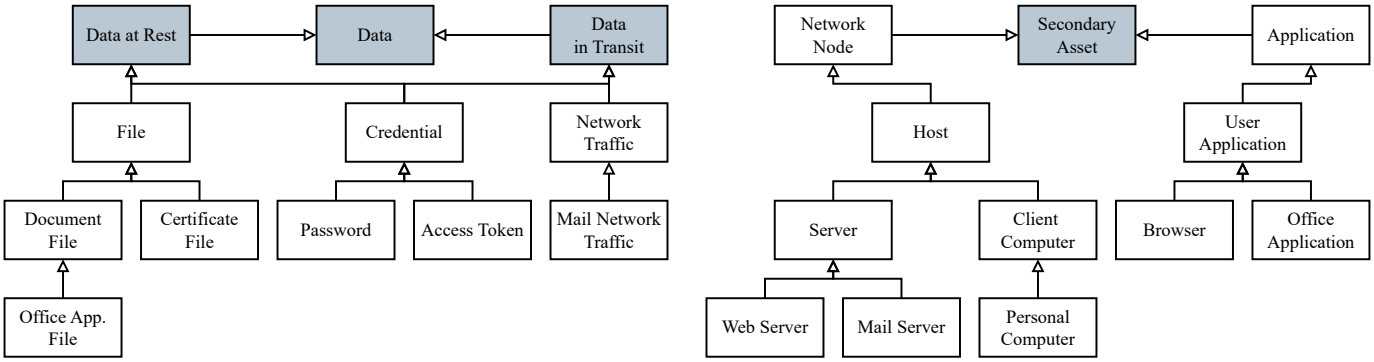

Fig. 2. Hierarchies of classes based on Digital Artifact Ontology for data and secondary assets. Filled rectangles represent classes that were not present in the ontology.

There are twelve allowed pairs of prerequisite and result vertices for attack techniques depicted in Figure 1 by arrows. Each technique considered during attack graph generation should be mapped to one of these pairs. Examples of attack techniques mapped to all combinations of asset types are listed in Table III. For more details, see Section IV.

The methodology should also express that the violated security property of one asset implies the violated security property of another cyber asset without using any ATT&CK techniques. For this purpose, we use *lists of influence* that consist of inheritance, composition, and other relationships applied on the attack path when there is a need to adjust the asset details, e.g., replace server by web server in Figure 2, filesystem by individual file, and a remote service (SSH, RDP) account by a local system account. One level can be influenced by another from its list of influence when having the same location. Table II contains examples of items from these lists.

IV. KCAG Generation with Asset Hierarchies

KCAG generator implemented in [5] was extended in this paper to support hierachies of cyber assets in three steps. The first step was to create predicates in the ruleset file for each class of cyber assets from Figure 2. Each of them has its specific attributes, according to Figure 1. Examples are a *privileged user account*, a *document file*, and a *password* listed in Listing 1 in the predicates section. During the second step, we created two types of rules – substitution rules and attack rules. The substitution rules were created based on lists of influence from Table II. The majority of them correspond to inheritance relationships from Figure 2. An example of a substitution rule that substitutes a *file* into a *document file* is shown in Listing 1 in the rules section.

The attack rules describe the conditions and results of ATT&CK techniques. Each technique should be mapped to cyber assets controlled by the attacker before and a cyber asset that is impacted after execution of the technique, such as in

```
% PREDICATES
privilegedUserAccount(_level, _property, _identity,
    _host).
documentFile(_level, _property, _host, _filename).
password(_level, _property, _username, _identity,
    _host, _application).

% RULES
interaction_rule(
    (documentFile(Level, Property, Host,
        Filename) :-
      file(Level, Property, Host, Filename)),
    rule_desc('Substitution', 1.0)).

interaction_rule(
    (application(2, availability, Host, Software) :-
      vulnerableAsset(Host, Software, CveId, remote,
          appAvailabilityLoss),
      networkService(Host, Software, Protocol, Port,
          _),
      networkResourceAccess(2, authentication, Host,
          Protocol, Port)),
    rule_desc('T1499.004 - Application or system
        exploitation', 1.0)).
```

Listing 1: Examples of predicates from Digital Artifact Ontology and rules containing one ATT&CK technique.

TABLE I
Relevant DAO taxonomies with possible cyber assets.

DAO Taxonomy	Examples of Assets	Category
Digital events	Open file, Network resource access	Action
Files	Document file, Office application file	Data
Network nodes	Host, Web server	Sec. asset
Software	Application, Browser	Sec. asset
Network traffic	Network session	Data
System calls	Open file, Create file	Action
User accounts	Local user account, Default user account	Actor
Credentials	Credential, Password	Data
Resources	File, Document file	Data

TABLE II
Content of the lists of influence for asset keys.

Asset Key	Influenced Types of Assets
Host	Privileged user account, File, Client computer
Service application	Local user account, Server
Password	Account
File	Document file, Configuration file

TABLE III
EXAMPLES OF ATTACK TECHNIQUES FOR EACH COMBINATION OF SOURCE AND DESTINATION ASSET TYPES FROM FIGURE 1. EACH TECHNIQUE CAN HAVE OTHER POSSIBLE PAIRS OF SOURCE AND DESTINATION ASSETS.

Source Asset	Src. Type	Destination Asset	Dst. Type	ID	Technique
Local user account	Actor	Execute command	Action	T1548.003	Abuse elevation control mechanism: Sudo and sudo caching
Network resource access	Action	Account	Actor	T1078.001	Valid accounts: Default accounts
Send email (by attacker)	Action	Open file (by user)	Action	T1204.002	User execution: Malicious file
Network resource access	Action	ARP cache	Data	T1557.002	Adversary-in-the-Middle: ARP cache poisoning
File	Data	Copy file	Action	T1560.001	Archive collected data: Archive via utility
Network resource access	Action	Application	Sec. asset	T1499.004	Endpoint DoS: Application or system exploitation
Host	Sec. asset	Network resource access	Action	T1563	Remote service session hijacking
Registry	Data	Service, Host	Sec. asset	T1543.003	Create or modify system process: Windows service
Host	Sec. asset	Password	Data	T1003.008	OS credential dumping: /etc/passwd and /etc/shadow
Service application	Sec. asset	Host	Sec. asset	T1133	External remote services
Local user account	Actor	Host	Sec. asset	T1203	Exploitation for client execution
Command line interface	Sec. asset	Account	Actor	T1078.001	Valid accounts: Default accounts

Table III. For example, the ATT&CK technique representing endpoint denial of service by application or system exploitation (T1499.004) has *network resource access* as the source asset and *application* as the destination asset. These assets appear in the attack rule for the technique in Listing 1. It is consequently necessary to enrich the source and destination assets from Table III with properties of assets that are required and optionally countermeasures that were not employed. Therefore, the attack rule for T1499.004 in Listing 1 requires that the impacted application was vulnerable (see predicate *vulnerableAsset*) and the attacker could access it (see predicate *networkService*).

We applied hierarchies of assets on a ruleset that was created without it. As a result, we used almost four times more predicates for level vertices. It reduced the number of manually created rules for attack techniques by one third. However, it is necessary to emphasize that manual enumeration of rules was hardly exhaustive before concerning possible cyber assets. The ruleset in the proof-of-concept implementation [18] contains rules for almost 60 ATT&CK techniques and subtechniques and more than 20 substitutions based on the lists of influence. It can be prepared in approximately two working days of net time with additional time to tune any inconsistencies.

During the third step, we prepared a transformation of IP flows gathered using IPFIX protocol and Syslog events to the generator's input file that conforms to the syntax defined by the ruleset. IP flows are used to populate facts about network services, open ports, and IP addresses. The purpose of IP flows is also to approximate the firewall rules indicated by transmitted communication. We counted with denied access by default. The Syslog events provided facts about installed software on hosts. Attack goals present in the input file are critical cyber assets that can be revealed using any method determining the criticality of cyber assets.

V. EVALUATION OF ATTACK GRAPH GENERATION

We evaluated KCAG generation on IP flows in IPFIX format and Syslog events from a cyber exercise with six defensive teams [19], which protected their identical networks of digital twin organizations against attackers but behaved in a different way. The networks contained public servers (e.g., mail and web servers), internal servers (e.g., database server, file server, and domain controller), and user desktops with Ubuntu and CentOS divided into several segments. Moreover, we used a graph database containing vulnerabilities for the operating systems of hosts populated by the CVE connector from CRUSOE [20]. The count of IP flows and the generator's input facts are listed in Table IV. There were hundreds of thousands of Syslog events. We also determined thirteen possible attack goals according to executed cyber attacks that represented the ground truth.

The attack graphs were generated for each team in units of seconds. The small height of the hierarchy of cyber assets in Figure 2 did not influence the performance. The approach revealed four or five attack goals for all teams except for team number five (see Table IV). The team's actions probably did not provide enough input data about their network. Targets of attack paths were mainly personal computers and a mail server. We did not find all attack goals due to not using all ATT&CK techniques. We also used a limited count of vulnerabilities and only essential properties from the input data.

Ten different substitution rules were applied in all six KCAGs. Approximately half of them substituted assets (even

TABLE IV
PROPERTIES OF KCAG GENERATION FOR INDIVIDUAL TEAMS. PATHS CONTAIN UP TO 18 VERTICES. $Paths_a$ DENOTES THAT WE FOCUS ON DIFFERENT ASSETS. $Paths_t$ MEANS DIFFERENT TECHNIQUES IN PATHS. SUBSTITUTIONS WERE BASED ON LISTS OF INFLUENCE. RATIO OF PATHS THAT CONTAIN SUBSTITUTIONS IS COMPUTED FOR $Paths_a$.

	T1	T2	T3	T4	T5	T6
Flows	66.5k	116.9k	63.4k	88.7k	78.3k	30.8k
Facts	1,103	1,110	1,151	1,512	1,154	912
Vertices	651	700	873	869	544	593
Edges	1,269	1,416	1,770	1,786	1,133	1,134
$Paths_a$	16.6k	107.0k	16.9k	164.4k	144.7k	1.7k
$Paths_t$	206	904	194	913	856	168
Goals	4	4	5	4	1	4
Substit.	83	92	109	103	64	96
$Ratio_{sub}$	0.36	0.26	0.37	0.27	0.28	0.51

TABLE V
ONE OF THE ATTACK PATHS PRESENT IN KCAGS AND EXPRESSED BY ITS LEVEL VERTICES AND TECHNIQUES.

No.	Description of Asset or Technique	Type
1	External actor located on the Internet.	Level
2	T1595 – Active Scanning	Technique
3	Network resource access to a host.	Level
4	T1078 – Valid Accounts	Technique
5	User account on the host.	Level
6	T1543 – Create or Modify System Process	Technique
7	Host – obtained root privileges.	Level
8	Substitution of assets	Technique
9	Root account on the host.	Level
10	T1489 – Service Stop	Technique
11	An application is not available.	Goal

multiple times) according to inheritance relationships. The remaining rules substituted other related cyber assets (see Table V for an example). Table IV shows the number of substitutions. We would miss a fraction of paths without using the lists of influence. Moreover, ten level predicates were used in KCAGs but only five out of them were not created from substitutions. The large number of vertices in Table IV is caused by attributes of these predicates since they create a lot of possibilities. In general, cyber assets from KCAGs represented accounts, software, files, kinds of hosts, and access to resources. All KCAGs used 21 ATT&CK techniques.

Postprocessing large KCAGs with lateral movements can be time-prohibitive. However, we can focus on shorter attack paths followed by attackers with a higher probability and containing all techniques from KCAGs. Paths with up to eighteen vertices and up to nine attack techniques were enumerated in tens of seconds on a personal computer with 64 GB RAM, 16 CPU cores, and a processor's clock speed of 2.5 GHz. High counts of paths for such small networks in Table IV were caused by sharing subpaths and reordering some techniques. The shortest path that was revealed consisted of two attack techniques – scanning of network infrastructure and network denial of service. Table V contains an example of a longer attack path with four ATT&CK techniques and one substitution.

VI. CONCLUSION

This paper extended kill chain attack graphs that depict attack paths in a network using MITRE ATT&CK with hierarchical categories of cyber assets from MITRE D3FEND's Digital artifact ontology. At the highest level, hierarchies contain four asset types and their attributes. Moreover, the approach allows adjusting levels of details about assets. As a result, this approach contributes to the automation of cyber threat modeling and projects future security posture based on vulnerabilities.

The evaluation indicated that the automated generation of graphs with hierarchies of assets was applicable to real-world input data from a cyber exercise. Graphs of nontrivial size were generated in reasonable execution time and contained a reasonable count of attack goals with respect to the ground truth and coverage of attack techniques. A proof-of-concept implementation and evaluation results are available in supplementary materials [18]. A promising future work is to investigate large language models to create a ruleset for automated generation based on the proposed approach and public knowledge bases.

ACKNOWLEDGMENT

This project has been funded by the European Union as part of the Horizon Europe Framework Program (HORIZON), under the grant agreement 101119681, Resilmesh.

REFERENCES

[1] "How to Manage Cybersecurity Threats, Not Episodes," Gartner, 2023, accessed: Apr 19, 2024. [Online]. Available: https://www.gartner.com/en/articles/how-to-manage-cybersecurity-threats-not-episodes

[2] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 workshop on New security paradigms*, 1998, pp. 71–79, doi: 10.1145/310889.310919.

[3] D. Tayouri, N. Baum, A. Shabtai, and R. Puzis, "A Survey of MulVAL Extensions and Their Attack Scenarios Coverage," *IEEE Access*, vol. 11, pp. 27 974–27 991, 2023, doi: 10.1109/ACCESS.2023.3257721.

[4] A.-M. Konsta, A. Lluch Lafuente, B. Spiga, and N. Dragoni, "Survey: Automatic generation of attack trees and attack graphs," *Computers & Security*, vol. 137, p. 103602, 2024, doi: 10.1016/j.cose.2023.103602.

[5] L. Sadlek, P. Čeleda, and D. Tovarňák, "Identification of Attack Paths Using Kill Chain and Attack Graphs," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, doi: 10.1109/NOMS54207.2022.9789803.

[6] K. Kaynar, "A taxonomy for attack graph generation and usage in network security," *Journal of Information Security and Applications*, vol. 29, pp. 27–56, 2016, doi: 10.1016/j.jisa.2016.02.001.

[7] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer," in *USENIX security symposium*, vol. 8. Baltimore, MD, 2005, pp. 113–128.

[8] "MITRE ATT&CK," The MITRE Corporation, 2015 – 2023, accessed: Jun 21, 2024. [Online]. Available: https://attack.mitre.org/

[9] M. Ahmed, S. Panda, C. Xenakis, and E. Panaousis, "MITRE ATT&CK-driven Cyber Risk Assessment," in *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 2022)*. New York, NY, USA: ACM, 2022, doi: 10.1145/3538969.3544420.

[10] "MITRE D3FEND," The MITRE Corporation, 2022, accessed: Jun 21, 2024. [Online]. Available: https://d3fend.mitre.org/

[11] S. Yu, "Cyber Defense Matrix," 2022, accessed: Jun 21, 2024. [Online]. Available: https://cyberdefensematrix.com/

[12] P. Aitken, B. Claise, and B. Trammell, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011, Sep. 2013, doi: 10.17487/RFC7011.

[13] R. Gerhards, "The Syslog Protocol," RFC 5424, Mar. 2009, doi: 10.17487/RFC5424.

[14] P. Johnson, R. Lagerström, and M. Ekstedt, "A Meta Language for Threat Modeling and Attack Simulations," in *Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018)*. New York, NY, USA: ACM, 2018, doi: 10.1145/3230833.3232799.

[15] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," *Software and Systems Modeling*, vol. 21, no. 1, pp. 157–177, 2022, doi: 10.1007/s10270-021-00898-7.

[16] A. Gylling, M. Ekstedt, Z. Afzal, and P. Eliasson, "Mapping Cyber Threat Intelligence to Probabilistic Attack Graphs," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2021, pp. 304–311, doi: 10.1109/CSR51186.2021.9527970.

[17] K. Zenitani, "Attack graph analysis: An explanatory guide," *Computers & Security*, vol. 126, p. 103081, 2023, doi: 10.1016/j.cose.2022.103081.

[18] L. Sadlek, M. Husák, and P. Čeleda, "Supplementary Materials: Hierarchical Modeling of Cyber Assets in Kill Chain Attack Graphs," Zenodo, Sep. 2024, doi: 10.5281/zenodo.13735708.

[19] D. Tovarňák, S. Špaček, and J. Vykopal, "Traffic and log data captured during a cyber defense exercise," *Data in Brief*, vol. 31, p. 105784, 2020, doi: 10.1016/j.dib.2020.105784.

[20] M. Husák, L. Sadlek, S. Špaček, M. Laštovička, M. Javorník, and J. Komárková, "CRUSOE: A toolset for cyber situational awareness and decision support in incident handling," *Computers & Security*, vol. 115, p. 102609, 2022, doi: 10.1016/j.cose.2022.102609.