IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 December 2022, accepted 11 March 2023, date of publication 15 March 2023, date of current version 23 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3257721

SURVEY

A Survey of MulVAL Extensions and Their Attack Scenarios Coverage

DAVID TAYOURI, NICK BAUM, ASAF SHABTAI, AND RAMI PUZIS
Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Be'er Sheva 8410501, Israel

Corresponding author: David Tayouri (davidtay@post.bgu.ac.il)

This work was supported in part by the CONCORDIA Project from the European Union's Horizon 2020 Research and Innovation Program under Grant 830927.

ABSTRACT Organizations employ various adversary models to assess the risk and potential impact of attacks on their networks. A popular method of visually representing cyber risks is the attack graph. Attack graphs represent vulnerabilities and actions an attacker can take to identify and compromise an organization's assets. Attack graphs facilitate the visual presentation and algorithmic analysis of attack scenarios in the form of attack paths. MulVAL is a generic open-source framework for constructing logical attack graphs, which has been widely used by researchers and practitioners and extended by them with additional attack scenarios. This paper surveys all of the existing MulVAL extensions and maps all MulVAL interaction rules to MITRE ATT&CK Techniques to estimate their attack scenarios coverage. This survey aligns current MulVAL extensions along unified ontological concepts and highlights the existing gaps. It paves the way for the systematic improvement of MulVAL and the comprehensive modeling of the entire landscape of adversarial behaviors captured in MITRE ATT&CK.

INDEX TERMS Attack graphs, attack scenario coverage, MITRE ATT&CK, MulVAL, network risk assessment.

I. INTRODUCTION

With the growth in the number of cyber attacks and their increasing complexity, cyber security risk assessment has become more essential [1], [2]. To improve their cyber security, organizations must identify their business-critical elements and protect them. For every possible threat, there may be several countermeasures; since it is infeasible to implement all countermeasures, organizations should assess the risks to their systems, prioritize these risks, and identify the security measures that will best reduce the threats to an acceptable level [3].

Different attack modeling techniques can be used to perform a risk assessment and present the risks visually, including misuse sequence diagrams (a use case method) [4], cyber kill-chain (a temporal method) [5], and fault trees (a graph-based method) [6]. A popular method of visually representing cyber risks is the attack graph. An attack graph is a risk assessment method representing attack states, transitions between them, and the related enterprise network vulnerabilities [7]. Attack graphs organize identified vulnerabilities into attack paths, composed of sequences of actions an attacker can take to reach and compromise system assets. Attack graphs can also help identify the attack paths most likely to succeed. As a consequence, attack graphs enable security administrators to prioritize an organization's network risks and decide which vulnerabilities to patch first.

Most attack graphs suffer from scalability challenges when modeling large networks [8]. Some frameworks address these challenges by adding assumptions such as the delete-free relaxation in logical attack graphs [9]. Nevertheless, attack graphs have two main advantages over other risk assessment methods. First, an attack graph models the interactions between vulnerabilities (multi-stage attacks) and the attacker's lateral movements (multi-host attacks) instead of focusing on individual vulnerabilities. Second, for the preconditions, consequences, and severity, attack graph risk assessment considers the effect of the exploitation of vulnerabilities on the specific target environment.

Different types of attack graphs have been proposed, including attack trees [10], state graphs [11], exploit dependency graphs [12], logical attack graphs [9], and multiple

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

prerequisite attack graphs [13] (a brief overview of attack graphs is presented in Section II). In this research, we focus on the logical attack graph - a directed graph in which leaves represent facts about the system, the internal nodes represent actions (attack steps) and their consequences (privileges), and the root represents an attacker's final goal. MulVAL is a well-known open-source framework for constructing logical attack graphs [14]. In addition to its scalability and extensibility, MulVAL is commonly used by researchers; as of February 2022, we identified 938 academic publications that mention MulVAL. A description of the MulVAL framework is presented in Section III-A.

To generate an attack graph, MulVAL requires four main inputs: security domain knowledge, such as CVE (Common Vulnerabilities & Exposures); information regarding the environment state, such as the principals and network and host configuration; the security policy; and reasoning rules. MulVAL's reasoning engine relies on interaction rules, which describe how facts and privileges are used to achieve attack goals. The original MulVAL framework provided a set of interaction rules representing a limited attack set. Since MulVAL was introduced in 2005, interaction rules have been added to represent additional attack scenarios. Researchers interested in using all MulVAL interaction rules would need to comprehensively review the literature and search through the hundreds of papers that mention MulVAL. Our first goal was to review these papers to collect the additional MulVAL interaction rules.

To identify all academic publications presenting MulVAL extensions, we performed a systematic literature review (see Section III-C). Of the 938 papers, we identified 38 extended MulVAL with additional interaction rules (see Section III-D). We provide a list of all of the MulVAL interaction rules we found in the literature (which are referred to as MulVAL rules in this paper).[1] The entire list of rules will enable the generation of attack graphs covering more attack scenarios.

To provide a comprehensive assessment of the risks faced by an organization's network, attack graphs should be able to present as many attack scenarios as possible. Thus, our second goal was to evaluate the extent to which the current MulVAL extensions cover known attack scenarios. The comprehensiveness and completeness of a set of interaction rules can be assessed using a knowledge base of known tactics, techniques, and procedures (TTPs). MITRE [15], which is the de facto standard of cyber threat modeling taxonomies, is a globally-accessible evidence-based knowledge base of TTPs; MITRE ATT&CK is described in Section IV-A. To evaluate the extent to which the MulVAL rules can represent different attacks, we systematically mapped all of the MulVAL rules to MITRE ATT&CK Techniques. Another essential benefit of mapping all of the MulVAL rules to ATT&CK Techniques is that mapping enables actionable insights: Techniques' Detection and Mitigation can be used to detect and mitigate the risks represented by the attack paths built with MulVAL rules.

Fig. 1 presents the relationships between the different entities of the enterprise cyber ecosystem: attackers try to attack enterprise networks; enterprises perform a risk assessment to prioritize the risks and allocate the resources to handle them; risk assessment can be achieved by using an attack graph generation tool, such as MulVAL, for which there are several inputs; among the inputs are reasoning rules, which can be mapped to MITRE ATT&CK to disclose the coverage of TTPs and enable the coverage of more TTPs (using an attack ontology).

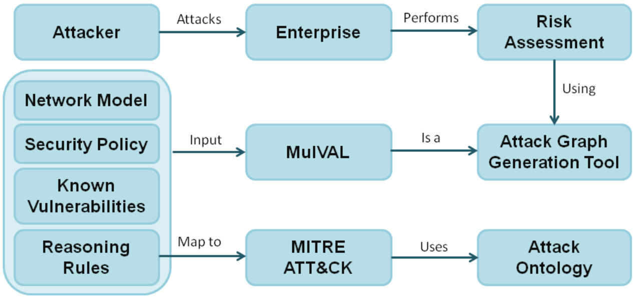


FIGURE 1. Relationships among the entities comprising the enterprise cyber ecosystem.

The contributions of this work are:

- We survey all of the MulVAL extensions found in the literature and provide the list of all published MulVAL rules.
- We map all available MulVAL rules to MITRE ATT&CK Techniques and summarize the attack coverage capabilities of existing MulVAL extensions.

II. ATTACK GRAPHS

An attack graph (AG) is a model that enables researchers and security administrators visually represent events that may lead to a successful attack scenario. Various AGs have been proposed in prior research. Hong et al. [16] conducted a survey reviewing all of the modeling techniques and AG generation tools presented in the literature. In this section, we describe the most common AG representations, including the attack tree (AT), state graph (SG), exploit dependency graph (EDG), logical attack graph (LAG), and multiple prerequisite attack graph (MPAG) representations. Fig. 2 depicts these representations (in blue) and their supported AG generation tools (in red) on a timeline graph, along with the number of citations (y-axis). We also review the common uses of AGs and the main challenges of modeling attacks with an AG.

The need for different AG representations stems from their use in diverse cyber domains and applications. For example, AG-based network security assessment methods can be utilized by modeling zero-day network resilience in an AG by defining a new zero-day safety metric that counts how many unknown vulnerabilities would be required to compromise

[1] The list of MulVAL rules is available at https://github.com/dtayouri/MulVAL-MITRE/blob/main/%E2%80%8F%E2%80%8FMulVAL%20Interaction%20Rules.xlsx

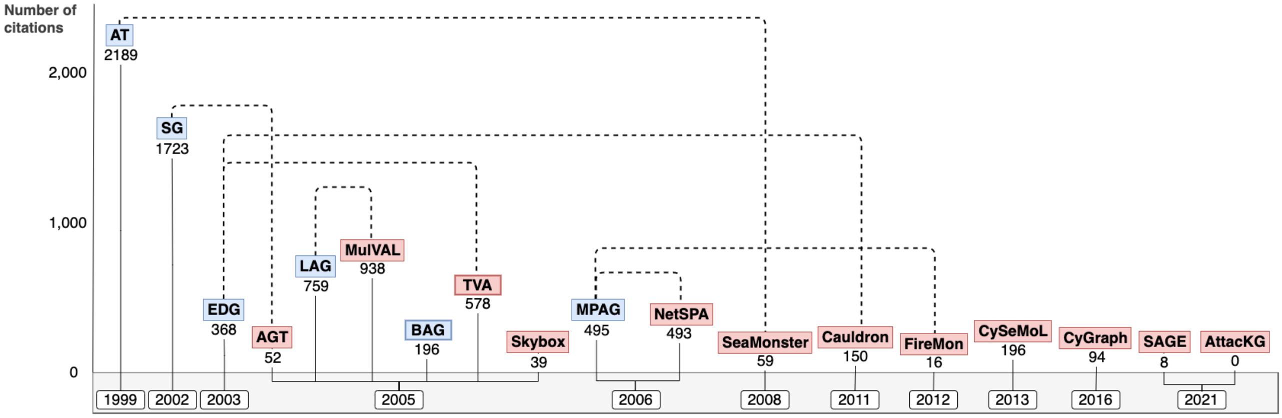


FIGURE 2. Evolution of attack graph generation methods: the publication year and the number of citations of AG representations (blue) and the respective AG generation tools (red).

network assets [17]. Noel et al. [18] used AGs to solve the sensor-placement problem, optimally placing IDS (intrusion detection system) sensors by covering the entire AG using the fewest number of sensors. Roschke et al. [19] presented an AG-based IDS (an alert correlation algorithm capable of analyzing dependencies between vulnerabilities) aggregating similar alerts and deciding if an isolated alert is a part of an ongoing multi-step attack. Liu et al. [20] used AGs for forensic analysis by showing that security administrators can prove, for example, that a series of IDS alerts are not isolated but rather correspond to a sequence of attacks in a coherent attack scenario. Wang et al. [21] used AGs to solve the minimum network hardening problem by constructing the set of specific vulnerabilities that should be patched to eliminate the attack paths leading to a given critical asset while minimizing the cost involved in removing those vulnerabilities.

a: ATTACK TREE (AT)

Tree-based graphical attack models are widely used to model network security [22], [23], [24], [25]. The most known tree-based representation, first published in 1999 by Schneier [10], was the AT. The root node of an AT represents the attacker's goal, and leaf nodes represent the attacker's sub-goals. Although an AT does not enumerate all possible system states, it still depends on the number of events. As a result, its main disadvantage is its poor scalability. In addition, modification of the AT nodes near the root node may change the entire tree. SeaMonster is a commonly used open-source AT generation tool based on the Eclipse framework [26]. SeaMonster focuses on helping developers during the software development lifecycle by providing three different viewpoints: existing vulnerabilities in the software, what causes the vulnerabilities, and possible countermeasures.

b: STATE GRAPH (SG)

In 2002, Sheyner et al. [11] presented the Attack Graph Toolkit, which is based on an SG. The Attack Graph Toolkit utilizes the SG in which each node represents a global network state, and edges correspond to attack actions initiated by the intruder. State enumeration-based approaches for AG representation suffer from degraded scalability. Enumerating all possible attack scenarios means dealing with a large state and action space, representing each possible system state as a node and each change of state caused by a single action taken by the attacker as an edge, resulting in a state space explosion [27]. First introduced in 2002, Ammann et al. [28] proposed a more scalable approach for AG representation called the monotonicity assumption. The authors addressed the scalability problem by assuming that the preconditions of an attack are not invalidated by the successful execution of another attack. Applying this assumption reduces the AG generation complexity from the exponential state space to the polynomial.

c: EXPLOIT DEPENDENCY GRAPH (EDG)

In 2003, Noel et al. [12] presented the EDG, which enumerates all possible exploit sequences while considering the monotonicity assumption. Each exploit or dependency appears only once, and all exploits contribute to the attack goal. As a result, there are no edges between independent exploits, and the AG size is quadratic to the number of exploits. However, enumerating all of the possible states of the attack using EDG is still an exponentially complex task. To address this limitation, a heuristic method can be used. In 2005, Jajodia et al. [29] proposed the Topological Vulnerability Analysis (TVA) tool, which is based on EDG. This tool uses two types of nodes: exploit and security condition nodes. Exploit nodes represent attack actions, and condition nodes represent either attack pre-conditions or post-conditions. The graph is built backward from the attacker's goal to the initial exploit. As a result, they do not include exploits generated in the forward dependency graph, and all of the exploits are relevant to the predefined attack goal. There is also an enterprise version of TVA called Cauldron, which provides additional visualizations, data integration features, automatic generation of mitigation recommendations, etc. [30].

d: LOGICAL ATTACK GRAPH (LAG)

In 2005, Ou et al. [9] introduced the LAG, a directed graph, which can also be represented as a tree. Due to the monotonicity assumption, the LAG size is polynomial in the size of the network being analyzed. A LAG can be generated using MulVAL's AG generation tool [31]. A description of this AG is provided in Section III-A.

e: BAYESIAN ATTACK GRAPH (BAG)

First proposed by Liu and Man [32], the BAG is a directed acyclic graphical model where the nodes represent different security states that an attacker can acquire, and the directed edges represent the dependencies between these security states. The potential attack paths are modeled by assigning conditional probability tables to edges, enabling the use of Bayesian inference methods. There is no generation tool available for BAG.

f: MULTIPLE PREREQUISITE ATTACK GRAPH (MPAG)

In 2006, Ingols et al. [13] presented the MPAG. The MPAG uses three types of nodes: state nodes, prerequisite nodes, and vulnerability nodes. State nodes describe the attacker's level of access on a specific host, prerequisite nodes can represent the reachability group or a set of credentials, and vulnerability nodes express a particular vulnerability on a specific host. MPAG node aggregation reduces the number of edges compared to a method in which state nodes point directly at vulnerability instance nodes since many state nodes can imply the same set of attacks. Several AG generation tools use MPAGs, such as NetSPA (network security planning architecture) [33] and FireMon [34], which is a commercial attack generation tool based on NetSPA. Both tools provide useful functionalities for security administrators, such as AG security assessment, prioritization of the vulnerabilities found, and suggestions on how to deal with the weaknesses discovered; however, these tools also have some limitations. For example, as an MPAG has many loops, this type of AG is difficult to understand.

In addition to the types of AGs mentioned above, some commonly used AG generation tools are worth mentioning.

g: SKYBOX

In 2005, Skybox View was presented by Skybox Security[2] as a solution for vulnerability and threat management. Skybox View is not an open-source product, so its underlying AG representation is not publicly available. However, like other commercial AG generation tools, it provides organizations with an end-to-end automated vulnerability management workflow and vulnerability discovery, assessment, prioritization, and remediation.

h: CySeMol

In 2013, Holm et al. [35], [36] presented the cyber security modeling language (CySeMoL), which is a modeling language and AG tool that can be used to estimate the cyber security of enterprise architectures. CySeMoL includes theoretical information on how attacks and defenses relate quantitatively; thus, security expertise is not required of its users. Users only need to model their system architecture and specify its characteristics to enable calculations.

i: CyGraph

In 2016, MITRE presented CyGraph, a graph-based AG generation tool [37]. This four-layer tool uses TVA/Cauldron as its network infrastructure and security posture layers. These layers import network topology information and search for vulnerabilities that might be exploited in cyber attacks. The other layers are cyber threats and mission dependencies, which describe the potential cyber threats and capture dependencies among various mission components.

j: CTI-BASED ATTACK GRAPH

In 2021, Nadeem et al. [38] presented SAGE, a framework for constructing AGs from cyber threat intelligence (CTI) instead of system vulnerabilities. In the same year, Li et al. [39] presented AttacKG, a method for extracting structured AGs from CTI reports and identifying the attack techniques.

Whereas we have described the most common AG representations and generation tools, there are also other types of AGs, such as DeepAG [40], which integrates AGs with deep learning techniques.

The main challenges in modeling an AG are visualization and scalability. Recently, Lallie et al. [41] surveyed 180 graphical attack representations proposed in the literature and concluded that more research is needed to standardize the representations. The scalability of each AG type is reviewed as part of Table 1, which compares the attack generation tools described above. In this research, we focus on the MulVAL framework, which uses a logical attack graph and will be described in Section III-A.

III. MulVAL EXTENSIONS

A. THE MulVAL FRAMEWORK

MulVAL (multi-host, multi-stage vulnerability analysis language) is an open-source publicly available logic-based attack graph generation tool [31]. MulVAL is based on the Datalog modeling language, a subset of the Prolog logic programming language. In MulVAL, Datalog is used to represent two types of entities:

- *Facts*: network topology and configuration, security policy, and known vulnerabilities
- *Rules*: also known as interaction rules, define the interactions between components in the network

Facts and rules are defined by applying a predicate p to some arguments: $p(t_1, \ldots, t_k)$. Each t_i can be either a constant or a variable. Datalog syntax indicates that a constant is an identifier that starts with a lowercase letter, whereas a variable begins with an uppercase letter. A wildcard expression can be defined by the underscore character ('_'). A sentence in

[2] https://www.skyboxsecurity.com/; the tool's name is changed to Vulnerability Control

TABLE 1. Comparison of common attack graph generation and visualization tools.

Name	Developers	Accessible	AG Type	Scalability	Intuitive Level	Year	No. of References	Paper Search	Tool Search
Attack Graph Toolkit	Carnegie Mellon University	Open source	SG	Poor, Exponential	Fair	2005	52	"Scenario graphs applied to security": 16	["Attack Graph Toolkit"]: 52
MulVAL	Kansas State University	Open source	LAG	O(N^2) - O(N^3)	Good	2005	938	"A scalable approach to attack graph generation": 757	["MulVAL"]: 938
TVA	George Mason University	Not open source, difficult to obtain	EDG	O(N^3)	Good	2005	578	"Topological analysis of network attack vulnerability": 578	["Topological Vulnerability Analysis"]: 547
Skybox View	Skybox Security, Inc.	Commercial Software	Unknown	O(N^3)	Good	2005	39	"Proactive Security for a Mega-Merger": 39	["skybox view" "attack graph"]: 15
NetSPA	Massachusetts Institute of Technology	Not open source, difficult to obtain	MPAG	O(NlgN)	Fair	2006	493	"Practical attack graph generation for network defense": 493	["NetSPA"]: 357
SeaMonster	Norwegian Univ. of Science and Technology and SINTEF research foundation	Open source	AT	Polynomial	Fair	2008	59	"SeaMonster: Providing tool support for security modeling": 37	["seamonster" "attack tree"]: 59
Cauldron	PROINFO Company, George Mason University	Commercial Software	EDG	O(N^3)	Good	2011	150	"Cauldron mission-centric cyber situational awareness with defense in depth": 142	[Cauldron "attack graph"]: 150
FireMon	FireMon, Massachusetts Institute of Technology	Commercial Software	MPAG	O(NlgN)	Good	2012?	16	No paper	["firemon" "attack graph"]: 16
CySeMoL	Royal Institute of Technology, Stockholm, Sweden	Not open source, difficult to obtain	Unknown	Polynomial?	Not Provided	2013	196	"The Cyber Security Modeling Language: A Tool for Assessing the Vulnerability of Enterprise System Architectures": 168	["cysemol"]: 196
CyGraph	MITRE	Not open source, difficult to obtain	Unknown	Scales well$^{(a)}$	Very Good$^{(b)}$	2016	94	"CyGraph: graph-based analytics and visualization for cyber security": 94	["cygraph" "attack graph"]: 46
SAGE	Delft University of Technology, Netherlands, Rochester Institute of Technology, US	Open source	Alert-driven	NA$^{(c)}$	Good	2021	8	"Alert-driven Attack Graph Generation using S-PDFA": 2	["SAGE" "attack graph"]: 8
AttacKG	Zhejiang University, National University of Singapore, Northwestern University	Open source	CTI-based	NA$^{(c)}$	Fair	2021	0	"Attackg: Constructing technique knowledge graph from cyber threat intelligence reports": 0	["AttacKG" "attack graph"]: 0

$^{(a)}$Graph database complexity depends on the part of the graph traversed by the query, not the total number of nodes in the database

$^{(b)}$CyGraph includes graph dynamics, layering, grouping, filtering, and hierarchical views

$^{(c)}$Alert-driven and CTI-based AG generators don't refer to network size; therefore, their scalability is irrelevant here

MulVAL is defined as Horn clauses of literals:

$$L_0 : -L_1, \ldots, L_n$$

L_0 is defined as the head, and $L_1, \ldots, L_n$ are defined as the body of the sentence, respectively. Each L_i in the body can be either a fact or an interaction rule. If the body ($L_1, \ldots, L_n$) literals are true, then the head (L_0) literal is also true. A sentence with an empty body is called a fact. For example, the following fact states that there is an identified vulnerability `CVE-2002-0392` in the `httpd` service running on `webServer01` instance:

```
vulExists(webServer01, "CVE-2002-0392", httpd).
```

A sentence with a nonempty body is called a rule. For example, the rule in Listing 1 says that if a `User` has ownership of `Path` on `Host`, and if an owner of `Path` on `Host` has the specified `Access`, then the `User` on `Host` can have the specified `Access` to `Path`.

```
localFileProtection(Host, User, Access, Path) :-
    fileOwner(Host, Path, User),
    ownerAccessible(Host, Access, Path).
```

Listing. 1. Interaction rule example.

Fig. 3 presents an example of a LAG generated by MulVAL: a code execution attack via a remote service (sshd) performed by using a compromised user account. In MulVAL, the graph representation is constructed as follows:

- Fact nodes (rectangles), also called primitive facts, represent the asset state, configuration, or network condition that must exist for the attack to exploit the vulnerability.
- Privilege nodes (diamonds), also called derived facts, represent the attack impact, e.g., the information or assets obtained by an attacker.
- Action nodes (circles), also called derivation or exploit nodes, represent the actions an attacker should perform to gain some privileges.

To execute an exploit, which means performing some action, the attacker needs all the privileges and facts that lead to that action. As a result, an action node will lead to a single privilege node.

As depicted in Fig. 4, MulVAL facts (which appear in blue) are constructed from:

- Vulnerabilities
  - Known vulnerabilities: CVEs registered in publicly available vulnerability databases, such as the NVD (National Vulnerability Database) [42], VulDB (Vulnerability Database) [43], WhiteSource Vulnerability Database [44], etc.
  - Unknown vulnerabilities: MulVAL facts can be used for simulating unknown vulnerabilities and testing network resilience against zero-day exploits. The following fact enables the simulation of unknown bugs:

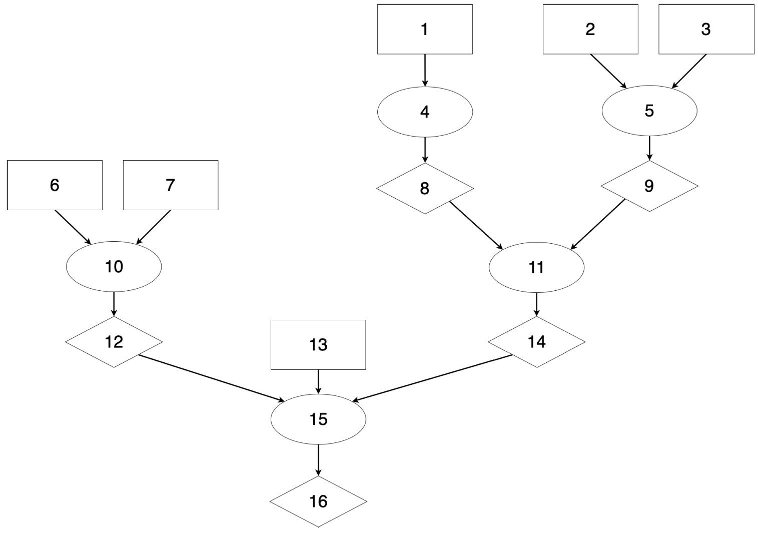
```
bugHyp(Host, Prog, ExploitRange, ExplConseq)
```

- Infrastructure: the infrastructure setup, containing information regarding the current environment state, such as network configuration (e.g., network topology, firewall rules), service configuration, accounts, installed software, principals, and data bindings (symbolic names).
- Security policy: the security policy loaded into the reasoning engine.

The vulnerability and infrastructure configuration required can be collected using custom scripts or existing tools and services such as Nessus [45] vulnerability scanner, host-based OVAL [46] agents, etc. The reasoning engine estimates the effect of the identified vulnerabilities on the system. This estimation is performed by applying the defined set of interaction rules to the generated facts. The MulVAL framework provides a default set of various interaction rules [14]. These rules are represented as action nodes in the LAG and can be categorized into two types:

- Environment rules (in yellow): describing additional security-related facts (see Definition 3 in Section III-B). For example, index 4 in Fig. 3 identifies `sshd` as a login service.
- Adversarial behavior (in red): describing an attack technique. For example, index 15 in Fig. 3 enables the attacker to apply a code execution technique.

Unlike CVEs, principals, network configuration, etc., the set of rules defining the adversary's behavior and the environment's mechanics rarely change. Rules can be extended to represent known tactics, techniques, and procedures (TTPs). However, procedures are a highly detailed description of a technique and are rarely modeled in the LAG. In addition, rules can be used to represent different IT advances such as near-field communication or cloud technologies [47], [48], [49]. Tactics (which appear in green in Fig. 4) describe the short-term goals of the attacker. They are represented as privilege nodes that are created by adversarial behavior. Each of these nodes advances the attacker toward the final



Index	Description
1	networkService(sshServer, sshd, tcp, 22, sshd).
2	located(victim, internet).
3	hacl(internet, sshServer, tcp, 22).
4	RULE 55: ssh is a login service.
5	RULE 58: Direct network access.
6	malicious(attacker).
7	incompetent(victim).
8	logInService(sshServer, tcp, 22).
9	netAccess(victim, sshServer, tcp, 22).
10	RULE 51: Incompetent user.
11	RULE 54: Access a host through a login service.
12	principalCompromised(victim, attacker).
13	hasAccount(victim, sshServer, Access).
14	canAccessHost(victim, sshServer).
15	RULE 52: Once the credentials of a principal are compromised, an attacker can compromise accounts of the principal on any machine the attacker has access to.
16	exeCode(attacker, sshServer, Access).

FIGURE 3. MulVAL example: code execution attack graph.

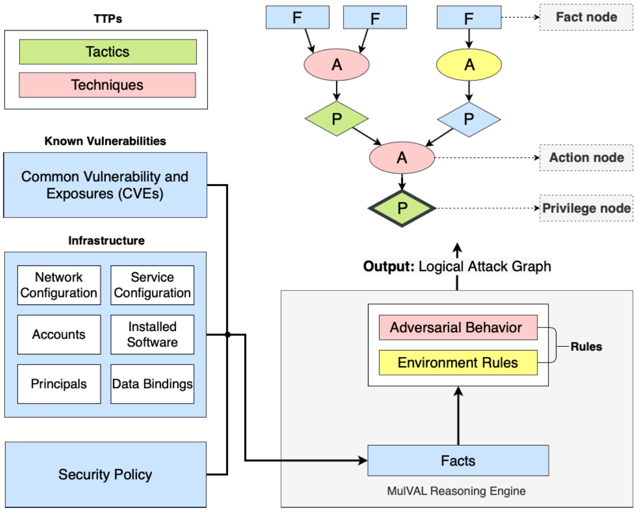


FIGURE 4. The MulVAL framework.

goal, which is achieved in the last privilege node. Techniques (in red) describe the attacker's actions.

Being generic and extensible, LAGs support many threat models characterized by attackers' goals, capabilities, and resources. Attackers may have arbitrary goals represented as assets [31]. LAGs may also support the various levels of attacker capabilities if they are defined as preconditions for exploits [50], [51]. However, due to the common delete-free relaxation in LAG solvers, modeling attacker resources may be challenging.

MulVAL uses the XSB (Extended Stony Brook) environment [52], which supports a declarative style of logic programming of Datalog programs called table execution. XSB enables effective dynamic programming that avoids the recomputation of previously calculated facts, thus enabling the reasoning engine to scale well with the size of the network.

In 2013, Yi et al. [53] compared several academic and commercial attack graph generation tools (TVA, Attack Graph Toolkit, NetSPA, MulVAL, Cauldron, FireMon, and Skybox View). The authors concluded that MulVAL is the most extendable and scalable framework; commercial tools may be more scalable and user-friendly; however, they are not open-source and are thus less suitable for academic research. In our review, we add five additional attack graph generation tools to the comparison. Table 1 is based on the comparison made by Yi et al. [53], with the addition of SeaMonster, CySeMoL, CyGraph, SAGE, and AttacKG, and four additional columns: Year (the year in which the tool was first published), Number of References (the larger value of the following two columns), Paper Search (the number of Google Scholar citations for the tool's main paper between 2005-2021), and Tool Search (the results of a search of the tool's keyword(s) in Google Scholar between 2005-2021).

Table 1 shows that MulVAL has several advantages:

- Availability: it is open-source.
- Scalability: its execution time is $O(n^2)$ relative to the size of the network [9].
- Extensibility: its underlying reasoning engine is written in a logical programming language, which enables users to extend functionality by writing custom rules.
- Compatibility: it leverages public vulnerability resources, which are continuously updated.
- Broad acceptance: as depicted in Table 1, MulVAL is the tool most referred to by researchers.

Therefore, in this study, we focus on the MulVAL attack graph framework and its reasoning engine in particular.

B. DEFINITIONS

Our first goal in this paper is to conduct a thorough survey and identify all papers extending MulVAL and adding new interaction rules to describe new attacks. We begin by providing some formal definitions, using the "Exploitation for Privilege Escalation" MITRE ATT&CK Technique. Each Technique can be implemented using one or more attack procedures. For example, the exploitation for privilege escalation technique can be implemented by executing code on a host where software with a vulnerability exists or by injecting a command into a host with a bad configuration.

The first procedure can be represented by MulVAL rules as presented in Listing 2:

```
execCode(Prin, Host, Perm) :-
    malicious(Prin),
    execCode(Prin, Host, Perm2),
    vulExists(Host, Software, localExploit, privEsc),
    setuidProgram(Host, Software, Perm).
```

Listing. 2. A technique procedure expression.

The description of an ATT&CK Technique implies one or more attack procedures. Such procedures may include interactions between multiple entities, such as users or computer resources. If a set of interaction rules encodes all relevant interactions to describe an attack procedure implied by a Technique, we say that this set of rules expresses the Technique.

Definition 1 (Expressing a technique): A set of MulVAL interaction rules $SIR = \{R_1, R_2, \ldots, R_n\}$ **expresses** a MITRE ATT&CK Technique if SIR is a minimal set that is sufficient to represent a Technique's procedure in an attack graph. The reason for the minimal set is efficiency and clarity. When there is a SIR expressing a Technique, MulVAL **covers** this Technique.

For example, according to the description of the Exploitation for Privilege Escalation Technique, "Adversaries may exploit software vulnerabilities to collect elevate privileges. Exploiting a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or the kernel itself to execute adversary-controlled code." Listing 2 **expresses** a procedure of this Technique. Since each Technique may have different procedures, it can be expressed with different sets of interaction rules (SIRs).

Definition 2 (Partial expression): If the number of rules in the expressing set $|SIR| > 1$, any subgroup ‘$SIR' \subset SIR$ **partially expresses** the Technique.

In the above example, the following rule **partially expresses** the Technique:

```
vulExists(Host, Software, localExploit, privEsc)
```

It should be mentioned that the same interaction rule can be used in different sets (SIRs) and partially express different Techniques.

Definition 3 (Environment rule): An **environment rule** is a predicate describing a security-related configuration, a formal software vulnerability, or a security policy defined by system administrators. Environment rules are used as input to MulVAL. An environment rule can be a primitive predicate, which will be referred to as a **primitive environment rule** (or simply a **fact**) or a derived predicate, which will be referred to as a **derived environment rule**.

For example, the following predicate is a **primitive environment rule (fact)** describing that a service `Prog` is running on `Host` as `User` and listening on `Port` of `Protocol`:

```
networkService(Host, Prog, Protocol, Port, User).
```

Listing 3 is a **derived environment rule** describing that if a `Prog` running on `Host` depends on `Library`, which has a vulnerability, then the `Prog` has the same vulnerability.

```
vulExists(Host, Prog, Consequence) :-
    vulExists(Host, Library, Consequence),
    dependsOn(Host, Prog, Library).
```

Listing. 3. Derived environment rule example.

Definition 4 (Building block): A derived predicate is called a **building block** if it is a general attack step that can be used in many SIRs, i.e., it can partially express many Techniques.

Listing 4 is a **building block** describing that if a user `Prin` has access to a `Host` from any source computer (`_Src`) on a `Port` of `Protocol`, and the `Host` enables login service in the same port and protocol, then user `Prin` can log in to the `Host`.

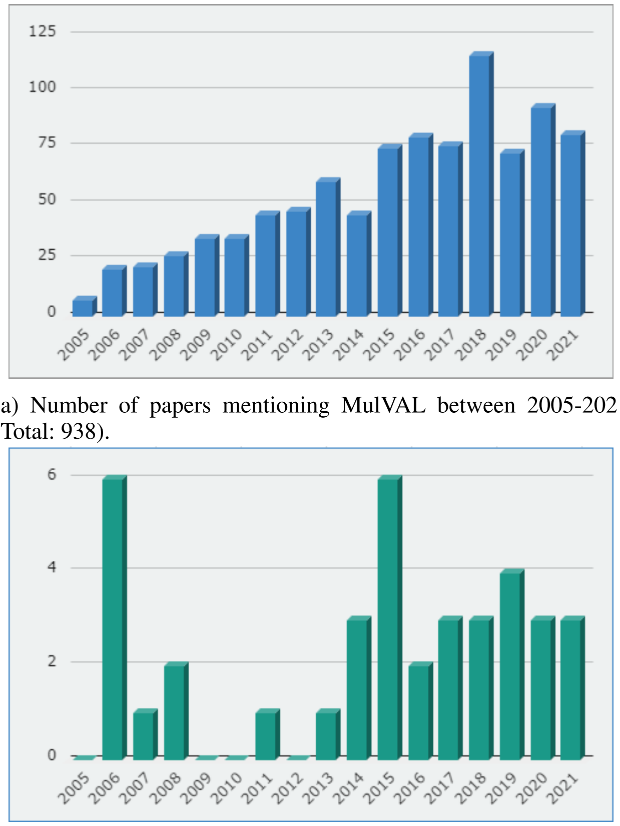
```
canLogInHost(Prin, Host) :-
    logInService(Host, Protocol, Port),
    netAccess(Prin, _Src, Host, Protocol, Port).
```

Listing. 4. Building block example.

This is a building block since accessing a host can be a step in many attack procedures (SIRs).

C. SEARCH METHOD

To find all of the MulVAL extensions, we performed a systematic literature review. Our review aimed to identify all academic papers that present MulVAL extensions. Since the papers do not always explicitly mention the fact that they are extending MulVAL, we searched Google Scholar for a single phrase “MulVAL,” excluding patents and quotes. Since the original paper describing MulVAL was published in 2005, the search was limited to the years 2005-2021. We found 938 papers available online on February 2022. The next step was to identify and remove all of the papers that mentioned MulVAL but did not add any new interaction rules. By manually examining these papers, we identified a set of 38 papers in which the authors presented an extension to the MulVAL framework either by introducing new interaction rules or by describing a method for defining new interaction rules. Fig. 5a shows the number of papers mentioning MulVAL each year from 2005 to 2021, and Fig. 5b indicates the number of papers extending MulVAL during that time.



(a) Number of papers mentioning MulVAL between 2005-2021 (Total: 938).

(b) Number of papers extending MulVAL between 2005-2021 (Total: 38).

FIGURE 5. A timeline of MulVAL publications.

D. EXTENSION FINDINGS

Of the 38 papers that presented an extension to the MulVAL framework, 21 papers defined new interaction rules representing new attack procedures. Together, the base MulVAL paper and these papers defined a total of 349 predicates: 92 environment rules (describing security-related configuration information, formal vulnerabilities, or security policies defined by system administrators) and 257 interaction rules defining new attack procedures. Table 2 presents a list of all of the papers extending MulVAL, including the extension's field, the methodology used, the number of times the paper

TABLE 2. Papers extending MulVAL.

Paper	Year	Venue	Extension Field	Methodology	No. of Citations	No. of Env. Rules	No. of SIRs
[31]	2005	Princeton University	NA (base MulVAL paper)	NA	61	35	26
[9]	2006	ACM Conference	Framework Improvements	NA	757	0	0
[54]	2006	CINNABAR Networks	Framework Improvements	NA	13	0	0
[55]	2006	NATO Security, IOS	Access Rules for Apache	Ad Hoc	1	0	3
[56]	2006	Princeton University	Access Rules for Windows	Ad Hoc	3	8	1
[57]	2006	Princeton University	Access Rules for Windows	Ad Hoc	39	0	4
[58], [59]	2006, 2007	Princeton University	Access Rules for Windows	Ad Hoc	4	18	8
[60]	2008	Kansas State University	Enterprise	Ad Hoc	12	1	2
[61]	2008	ACM Conference	Framework Improvements	NA	58	0	0
[62]	2011	Springer	Enterprise	Ad Hoc	52	0	2
[63], [64]	2013, 2015	IEEE	Mobile Connectivity	Ad Hoc	51	0	1
[65], [66], [67], [68]	2014, 2015	IEEE	Framework Improvements	NA	37	0	0
[69], [70]	2014, 2018	Springer	Cloud Computing	Methodical	9	6	4
[71]	2015	Electrical Eng. & Informatics	Framework Improvements	NA	19	0	0
[72]	2015	IEEE	Enterprise	Ad Hoc	4	1	1
[73], [74]	2015, 2016	Springer	Infected USB	Ad Hoc	0	0	1
[75]	2016	IEEE	Enterprise	Ad Hoc	11	2	3
[76]	2017	IEEE	Rule Generation	Methodical	7	0	0
[77]	2017	Springer	Cloud Computing	Methodical	13	0	2
[49]	2017	Springer	Cloud Computing	Methodical	6	0	0
[78]	2018	Eastern Washington Univ.	Framework	NA	0	0	0
[79]	2018	Springer	Graph Connectivity	NA	11	0	0
[48]	2019	CentraleSupélec	Cloud Computing	Methodical	1	8	5
[80]	2019	IEEE	Framework Improvements	NA	0	0	0
[81]	2019	IEEE	Enterprise	Ad Hoc	8	0	8
[82]	2019	ACM Conference	Enterprise	Methodical	8	1	6
[47]	2020	IEEE	Enterprise	Methodical	12	0	60
[83]	2020	DiVA	Data Criticality	Methodical	0	3	11
[84]	2020	IEEE	3D Printer	Ad Hoc	2	10	54
[85]	2021	IEEE	Framework Improvements	Methodical	1	0	1
[86]	2021	ACM Conference	Rule Generation	Methodical	0	0	0
[87]	2021	arXiv Preprint	Machine Learning	Methodical	0	0	53
Total						92	257

has been cited, and the number of environment and interaction rules. We have classified the interaction rules into the following six categories: framework improvements, access rules, enterprise, cloud computing, rule generation, and others.

1) FRAMEWORK IMPROVEMENTS

Ou et al. [9] demonstrated how to produce a derivation trace in the MulVAL logic-programming engine and how the trace can be used to generate a LAG in quadratic time. Bacic et al. [54] extended MulVAL to improve network representation and derive rules more intuitively. Saha [61] extended the MulVAL framework to include complex security policies and extended the LAG concept to justify why a negated subgoal failed. Liu et al. [65], [66], [67], [68] used evidence obtained from security events to construct an attack scenario and build an evidence graph. Sembiring et al. [71] introduced three methods to improve the MulVAL framework: employing the Common Vulnerability Scoring System (CVSS) to calculate the probability of vulnerability variables and the Common Configuration Scoring System (CCSS) to calculate the probability of system security configuration vulnerabilities; introducing the concept of interdependence between vulnerability variables in Bayesian senses; and analyzing the impact of a change in the system security configuration on the probability of vulnerabilities in the context of Bayesian probability. Anderson [78] explored enhancing estimations of factor analysis of information risk vulnerability by modeling interactions between threat actors and assets through attack graphs. Appana et al. [80] proposed applying a ranking algorithm on the mission impact graph based on the MulVAL attack graph. Stan et al. [85] proposed a method that expresses the risk of the system using an extended attack graph model that considers the prerequisites and consequences of exploiting a vulnerability, examines the attacker's potential lateral movements, and expresses the physical network topology as well as vulnerabilities in network protocols.

2) ACCESS RULES

Bhatt et al. [55] presented a model-driven technique for automated policy-based access analysis and added three access rules for Apache. Govindavajhala et al. [56], [57], [58], [59] suggested separating scanning from analysis to reduce the size of code running in privileged mode. They also demonstrated how to extend the MulVAL framework to reason about the security of a network with hosts running disparate operating systems. In particular, they illustrated 39 reasoning rules for Windows to find misconfigurations of the access control lists.

3) ENTERPRISE

Homer et al. [60] presented methodologies that can automatically identify and trim portions of an attack graph that do not help a user understand the core security problems. Ou et al. [62] presented an approach which, given component metrics that characterize the likelihood that individual vulnerabilities can be successfully exploited, computes a numeric value representing the cumulative likelihood for an attacker to succeed in gaining a specific privilege or carrying out an attack in the network. Jilcott [72] presented a technology that automatically maps and explores the firmware/-software architecture of a commodity IT device and then generates attack scenarios for the device. Acosta et al. [75] augmented MulVAL to incorporate network layer misconfigurations. In particular, they presented ARP spoofing and route hijacking scenarios. Khakpour et al. [81] defined several rules for the exploitation and propagation of vulnerabilities. Inokuchi et al. [82] proposed a methodical procedure for defining new interaction rules, and they applied the method to define four categories of behavior: privilege escalation, credential access, lateral movement, and execution. Stan et al. [47] presented an extended network security model for MulVAL that considers the physical network topology, supports short-range communication protocols, models vulnerabilities in the design of network protocols, and models specific industrial communication architectures. They also introduced an extensive list of 60 new interaction rules.

4) CLOUD COMPUTING

Sun et al. [69], [70] referred to two cloud risks: virtual machine (VM) images may be shared between different users, and VMs owned by different tenants may co-reside on the same physical host machine. Sun et al. [77] dealt with the gap between mission impact assessment and cyber resilience in the context of cloud computing. The authors bridged this gap by developing a graphical model that interconnects the mission dependency graphs and cloud-level attack graphs. Albanese et al. [49] proposed building cross-layer Bayesian networks to infer the stealthy bridges between enterprise network islands in clouds. Mensah [48] extended MulVAL to include cloud virtualization vulnerabilities.

5) RULE GENERATION

Jing et al. [76] presented a tool that can parse vulnerability descriptions, as provided in the CVE, to retrieve relevant information for generating interaction rules that can be incorporated into MulVAL. Binyamini et al. [86] presented an automated framework for modeling new attack techniques from the textual description of a security vulnerability. Their framework enables the automatic generation of MulVAL interaction rules from the NVD.

6) OTHERS

Almohri et al. [63], [64] addressed the problem of statically performing a rigorous assessment of a set of network security defense strategies to reduce the probability of a successful large-scale attack in a complex, dynamically changing network architecture. Dong et al. [73], [74] presented common input scenarios for different model-based security assessment tools. Cao et al. [79] proposed a business process impact assessment method, which measures the impact of an attack targeting a business-process-support enterprise network. Zhou [83] extended the security risk analysis with data criticality and introduced 14 new interaction rules. McCormack et al. [84] focused on identifying security threats to networked 3D printers. Bitton et al. [87] extended MulVAL with 54 interaction rules to model attacks on machine learning production systems.

IV. COVERAGE OF ATTACK SCENARIOS IN MulVAL

To estimate the coverage of attack scenarios by MulVAL, we decided to map MulVAL interaction rules to MITRE ATT&CK. Section IV-A describes MITRE ATT&CK, and Section IV-B presents the expressed ATT&CK Techniques in MulVAL.

A. MITRE ATT&CK

The Mitre Corporation (MITRE) is an American nonprofit organization dedicated to bringing innovative ideas into existence in different areas related to safety and security [88]. MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) is a knowledge base of adversarial tactics and techniques based on real-world observations [89]. The ATT&CK knowledge base is used as a foundation for developing specific threat models and methodologies in the cybersecurity community. ATT&CK provides a common taxonomy for both offense and defense and has become a useful conceptual tool across many cybersecurity disciplines, helping improve network and system defenses against intrusions.

MITRE ATT&CK reflects the various phases of an adversary's attack life-cycle and focuses on how external adversaries compromise and operate within computer information networks. ATT&CK is a behavioral model that consists of the following core components:

- Tactics, denoting short-term, tactical adversarial goals during an attack
- Techniques, describing how adversaries achieve tactical goals
- Sub-techniques, describing more specific means by which adversaries achieve tactical goals at a lower level than techniques
- Documented adversary use of techniques, their procedures, and other metadata

ATT&CK has different use cases, including: adversary emulation, red teaming, behavioral analytic development, defensive gap assessment, SOC (security operations center) maturity assessment, and cyber threat intelligence (CTI) enrichment. For example, Oosthoek et al. [90] plotted a sample of 951 Windows malware families on the ATT&CK framework to obtain insights on trends in attack techniques used to target Windows. Maynard et al. [91] created an ATT&CK model of a hacktivist and mapped the threat to critical infrastructure to define better the skills and methods a hacker might employ. To assist developers and administrators in cultivating an attacker mindset, Munaiah et al. [92] used the MITRE ATT&CK framework to characterize an attacker's campaign in terms of Tactics and Techniques. Analysts can use the ATT&CK framework to structure intelligence about adversary behavior, and defenders can structure information about what behavior they can detect and mitigate [93]. By overlaying information from several adversary groups, they can create threat-based awareness of the gaps adversaries exploit. Such analysis also improves CTI actionability for decision-makers.

Besides MITRE ATT&CK, there are other known methods of threat modeling. Shevchenko [94] summarized 12 threat-modeling methods, including Microsoft STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege), PASTA (process for attack simulation and threat analysis), and LINDDUN (linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness, non-compliance). Another threat model is the cyber kill chain, a traditional model used to analyze cyber security threats, malware infecting computer systems, covert and illegitimate channels found on a network, and insider threats [95]. These models help gain an increased understanding of high-level processes and adversary goals. However, the MITRE ATT&CK model is more effective at conveying the individual actions performed by adversaries, how one step relates to another and to tactical adversarial objectives, and how the actions correlate with data sources, defenses, and configurations.

Representing an attack in terms of Tactics, Techniques, and Sub-techniques provides a means of balancing the technical details in the Technique description and the attack goals represented by the Tactics. Tactics represent the ''why'' of an ATT&CK Technique or Sub-technique - the adversary's tactical objective, i.e., the desired outcome of performing an action. Techniques represent the ''how'' - an adversary's actions to achieve a tactical objective. Sub-techniques further break down behaviors described by Techniques. Procedures are the specific implementations that adversaries use to apply Techniques or Sub-techniques. In addition to textual descriptions, metadata, Sub-techniques, and Procedures, a Technique may also include:

- Group - known groups of adversaries that are tracked and reported on in threat intelligence reports.
- Software - tools and malware used by adversaries.
- Mitigation - security concepts and classes of technologies that can be used to prevent a technique from being successfully executed.
- Detection - methods for detecting an adversary's use of a Technique.

The relationships between these concepts are depicted in Fig. 6. The Adversary Group and Software (on the left) are related to the attacker, whereas the Detection and Mitigation (on the right) are related to the defender.

ATT&CK is organized in a series of technology domains – the ecosystem an adversary operates within. MITRE has defined three technology domains: Enterprise, Mobile, and ICS (industrial control system). In this work, we focus on the Enterprise domain. In the major version of MITRE ATT&CK Enterprise from October 2020 (ATT&CK Content version 8.0), two pre-attack Tactics were added, bringing the total number of Tactics to 14 [15]. Regarding ATT&CK's coverage, it is important to note that, generally, coverage of every ATT&CK Technique is unrealistic [96]. Similarly, since each ATT&CK Technique may have many implementation procedures that an adversary can use, and we cannot possibly know all of them, it is unrealistic to cover all procedures for a given technique.

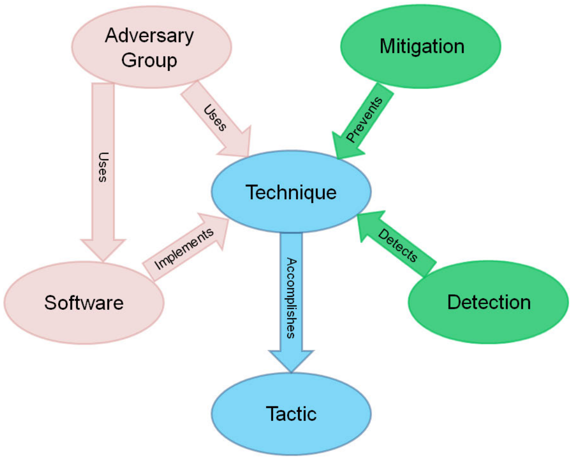


FIGURE 6. Relevant relationships between MITRE ATT&CK concepts.

B. EXPRESSED ATT&CK TECHNIQUES IN MulVAL

Our second goal in this paper is to map MulVAL rules to MITRE ATT&CK Techniques. Mapping between the most commonly-used attack graph generation tool (i.e., MulVAL) and the MITRE ATT&CK threat model will enable researchers and security administrators to handle additional realistic attack scenarios.

During the mapping process, we encountered a problem with Techniques associated with more than one Tactic. These Techniques are presented in Table 3. When implementing interaction rules to express a Technique, in some cases, the implementation can change depending on the Tactic (which is the attack goal). Since each Technique in ATT&CK is supposed to be unique, in cases where the Tactic may affect the Technique, it may be better to define a different Technique for each Tactic. The last column in Table 3 reflects this problem, indicating whether a different Technique for each Tactic is recommended.

In addition, as mentioned above, McCormack et al. [84] defined a list of 64 new interaction rules, focusing on identifying security threats to networked 3D printers, and Bitton et al. [87] defined a list of 54 interaction rules to model adversarial machine learning. Attacks on 3D printing and machine learning are not yet covered in ATT&CK Techniques.

To map MulVAL sets of interaction rules (SIRs) to ATT&CK Techniques, we manually analyze each SIR, and according to the SIR's description and predicates, we first map it to a Tactic and then to a specific Technique. Fig. 7 presents the method of mapping a SIR to a Technique. If a SIR

TABLE 3. MITRE ATT&CK techniques listed under two or more tactics.

Technique	Tactics	Separation Recommended
Abuse Elevation Control Mechanism	Privilege Escalation, Defense Evasion	No
Access Token Manipulation	Privilege Escalation, Defense Evasion	No
BITS Jobs	Persistence, Defense Evasion	No
Boot or Logon Autostart Execution	Persistence, Privilege Escalation	Yes
Boot or Logon Initialization Scripts	Persistence, Privilege Escalation	Yes
Create or Modify System Process	Persistence, Privilege Escalation	Yes
Event-Triggered Execution	Persistence, Privilege Escalation	Yes
External Remote Services	Initial Access, Persistence	No
Group Policy Modification	Privilege Escalation, Defense Evasion	No
Hijack Execution Flow	Persistence, Privilege Escalation, Defense Evasion	No
Input Capture	Credential Access, Collection	No
Man-in-the-Middle	Credential Access, Collection	No
Modify Authentication Process	Defense Evasion, Credential Access	No
Network Sniffing	Credential Access, Discovery	Yes
Pre-OS Boot	Persistence, Defense Evasion	No
Process Injection	Privilege Escalation, Defense Evasion	No
Replication Through Removable Media	Initial Access, Lateral Movement	Yes
Scheduled Task/Job	Execution, Persistence, Privilege Escalation	Yes
Software Deployment Tools	Execution, Lateral Movement	No
Traffic Signaling	Persistence, Defense Evasion, Command and Control	No
Use Alternate Authentication Material	Defense Evasion, Lateral Movement	No
Valid Accounts	Initial Access, Persistence, Privilege Escalation, Defense Evasion	Yes
Virtualization/Sandbox Evasion	Defense Evasion, Discovery	No

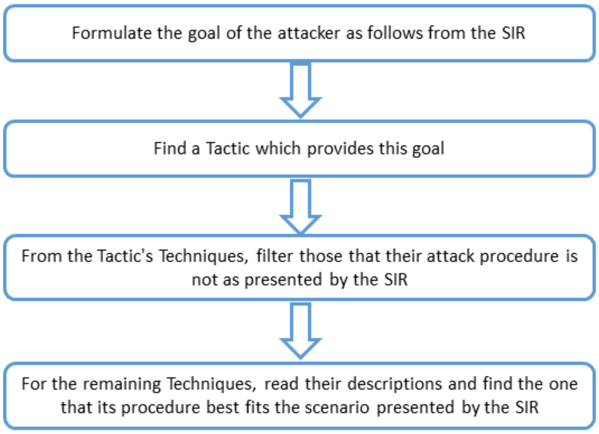

Formulate the goal of the attacker as follows from the SIR
Find a Tactic which provides this goal
From the Tactic's Techniques, filter those that their attack procedure is not as presented by the SIR
For the remaining Techniques, read their descriptions and find the one that its procedure best fits the scenario presented by the SIR

FIGURE 7. Mapping a SIR to a Technique.

TABLE 4. Expressed techniques in tactics.

Tactic [15]	No. of Techniques	No. of Expressed Tech.
Reconnaissance	10	0
Resource Development	6	0
Initial Access	9	5
Execution	10	5
Persistence	18	2
Privilege Escalation	12	4
Defense Evasion	37	3
Credential Access	14	7
Discovery	25	3
Lateral Movement	9	3
Collection	17	3
Command and Control	16	0
Exfiltration	9	3
Impact	13	4
Total	205	42

is a general rule that may partially express many techniques, we call it a building block and do not connect it to a particular Technique. In some cases, the same SIR can express different Techniques. In these cases, we connect the SIR to all of those Techniques. Since the SIRs were defined in different studies for different purposes, there are some Techniques with a few SIRs, each expressing a different procedure, and many Techniques remain uncovered. Table 4 presents the number of Techniques in each Tactic and the number of expressed Techniques in MulVAL for each Tactic.

Some Tactics are covered more, e.g., Initial Access, Execution, and Credential Access, and some Tactics are covered less or are not covered at all, e.g., Reconnaissance, Resource Development, and Command and Control. We provide the complete list of MulVAL rules and their mappings to MITRE ATT&CK Enterprise Techniques.[3] To generate attack graphs representing specific attack scenarios, one can only use the interaction rules mapped to the relevant Tactics or Techniques. For example, to assess only the risks of initial access scenarios in a network, one should use the interaction rules mapped to the Initial Access Tactic's Techniques. Mapping all of the MulVAL rules to ATT&CK Techniques also enables actionable insights: as mentioned in the previous section, Techniques' Detection and Mitigation can be used to detect and mitigate the risks found with MulVAL rules that were part of the attack graph generation.

Table 5 presents the list of Enterprise Techniques expressed with MulVAL rules, the number of SIRs mapped to them, and the popularity analysis of expressed Techniques, as described below. There are Techniques with many expressed procedures, e.g., Man-in-the-Middle, Exploitation for Privilege Escalation, and Exploitation for Client Execution. The reason for this may be the fact that these are popular attack techniques and therefore were expressed by different researchers. The table also presents the number of adversary groups mapped to each Technique, i.e., the number of groups that

[3] The list of MITRE ATT&CK Enterprise Techniques and the MulVAL rules mapped to each Technique is available at https://github.com/dtayouri/MulVAL-MITRE/blob/main/ATT%26CK%20Enterprise%20Techniques%20with%20MulVAL%20IR.xlsx

used these Techniques (and their Sub-techniques), and the number of software tools (used to conduct attacks) mapped to each Technique. The mapping is based on ATT&CK Enterprise v9.0.

As can be seen in the table, the average number of adversary groups using each Enterprise Technique is 13, and the average number of software tools using each Enterprise Technique is 31. We can see that there are some expressed Techniques where the number of mapped groups and software tools is much higher than the average, e.g., Command and Scripting Interpreter, File and Directory Discovery, Process Injection, and Phishing. The number of groups and software tools using each Technique can be used to prioritize the Techniques to express. The number of Group-Technique mappings (i.e., the number of adversary groups using each Technique, including Sub-techniques) for all Enterprise Techniques is 2,390; for expressed Techniques, it is 811, which represents 34% of Group-Technique mappings. This percentage is much higher than the percentage of the expressed Techniques, which is 20%. This indicates that expressed Techniques are the more popular techniques used by adversaries. The table also presents the papers with SIRs expressing each Technique, the number of times these papers have been cited, and the average number of citations per paper.

Fig. 8 (in the appendix) presents the Enterprise Techniques expressed by MulVAL rules as a matrix.

As an example of an expressed Technique, the Endpoint Denial of Service (DoS) Technique expressed by SIRs is presented in Listing 5:

```
dos(Principal, Host) :-
    localAccess(Principal, Host, User),
    localService(Host, Prog, User),
    vulHost(Host, VulID, Prog, localExploit, dos),
    malicious(Principal).
dos(Principal, DstHost) :-
    malicious(Prin),
    l2Access(Prin, SrcHost, DstHost, Prot, BusID, bus)
systemDown(Host) :-
    execCode(Host, _Perm2),
    vulExists(Host, _, SW, localExploit, Overuse),
    misuseAction(Overuse).
```

Listing. 5. Endpoint DoS Technique expressed with SIRs.

V. RELATED WORK

Several previous studies performed surveys of different attack generation tools. Yi et al. [53] surveyed and analyzed attack graph generation and visualization technology and compared several academic and commercial attack graph generation tools. Barik et al. [97] presented a consolidated view of major attack graph generation and analysis techniques. In an extensive survey of relevant papers, Haque et al. [98] summarized the different approaches to attack modeling, i.e., attack graphs and attack trees. Hong et al. [16] discussed the current state of graphical security models in terms of four phases: generation, representation, evaluation, and modification. Garg et al. [99] conducted a literature review focusing on the generation and analysis of attack graphs. He et al. [100] surveyed unknown vulnerability risk assessment based on directed graph models and classified their security metrics. By analyzing more than 180 attack graphs and attack trees, Lallie et al. [41] presented empirical research aimed at identifying how attack graphs and attack trees present cyber attacks in terms of their visual syntax. None of the papers mentioned above surveyed MulVAL attack graph generation extensions.

Many studies mapped attack entities, such as malware, CVE, and CTI, to MITRE ATT&CK, as the de facto standard for cyber threat modeling. Oosthoek et al. [90] mapped Windows malware families to the ATT&ACK framework. Legoy [101] evaluated different multi-label text classification models to retrieve TTPs from textual sources based on the ATT&CK framework and developed a tool for extracting ATT&CK Tactics and Techniques from cyber threat reports to a structured format. Aghaei et al. [102] suggested using machine learning, deep learning, and natural language processing to map CVE to CAPEC and ATT&CK automatically and found the appropriate mitigation for each CVE. By mapping the MITRE ATT&CK Matrix to the NIST cyber security framework, Kwon et al. [103] offered approaches and practical solutions to cyber threats. Purba et al. [104] defined a cyber-phrase embedding model to map CTI texts to the ATT&CK ontology. They created an ontology based on MITRE ATT&CK by integrating 2,236 attack patterns associated with ATT&CK Tactics and Techniques. Lee et al. [105] analyzed 10 selected cyber attacks in which fileless techniques were utilized and mapped the attacks to ATT&CK Techniques. However, none of these works mapped MulVAL interaction rules to MITRE ATT&CK Techniques.

To the best of our knowledge, we are the first to survey all MulVAL extensions and map all of the MulVAL interaction rules to MITRE ATT&CK Techniques.

VI. SUMMARY

AGs, in general, and MulVAL, in particular, are essential tools for network risk assessment and cyber security improvement. For providing a comprehensive risk assessment of an organization's network, attack graphs should be able to present as many attack scenarios as possible. The main security goal of this paper is to assess the coverage of attack scenarios supported by a popular logical AG generation framework.

A. INSIGHTS

Our main insights are: 1) In academic research, MulVAL is the most commonly used attack graph generation framework. 2) MulVAL interaction rules can be mapped to ATT&CK Tactics and Techniques. 3) Today, MulVAL rules cover less than a quarter of the ATT&CK Techniques; therefore, its risk assessment capability still needs to be improved. 4) There is a need for AG generation tools with complete and up-to-date coverage of attack scenarios.

B. MAIN CONTRIBUTIONS

Since MulVAL was introduced in 2005, interaction rules have been added to represent additional attack scenarios.

TABLE 5. Popularity analysis of expressed techniques.

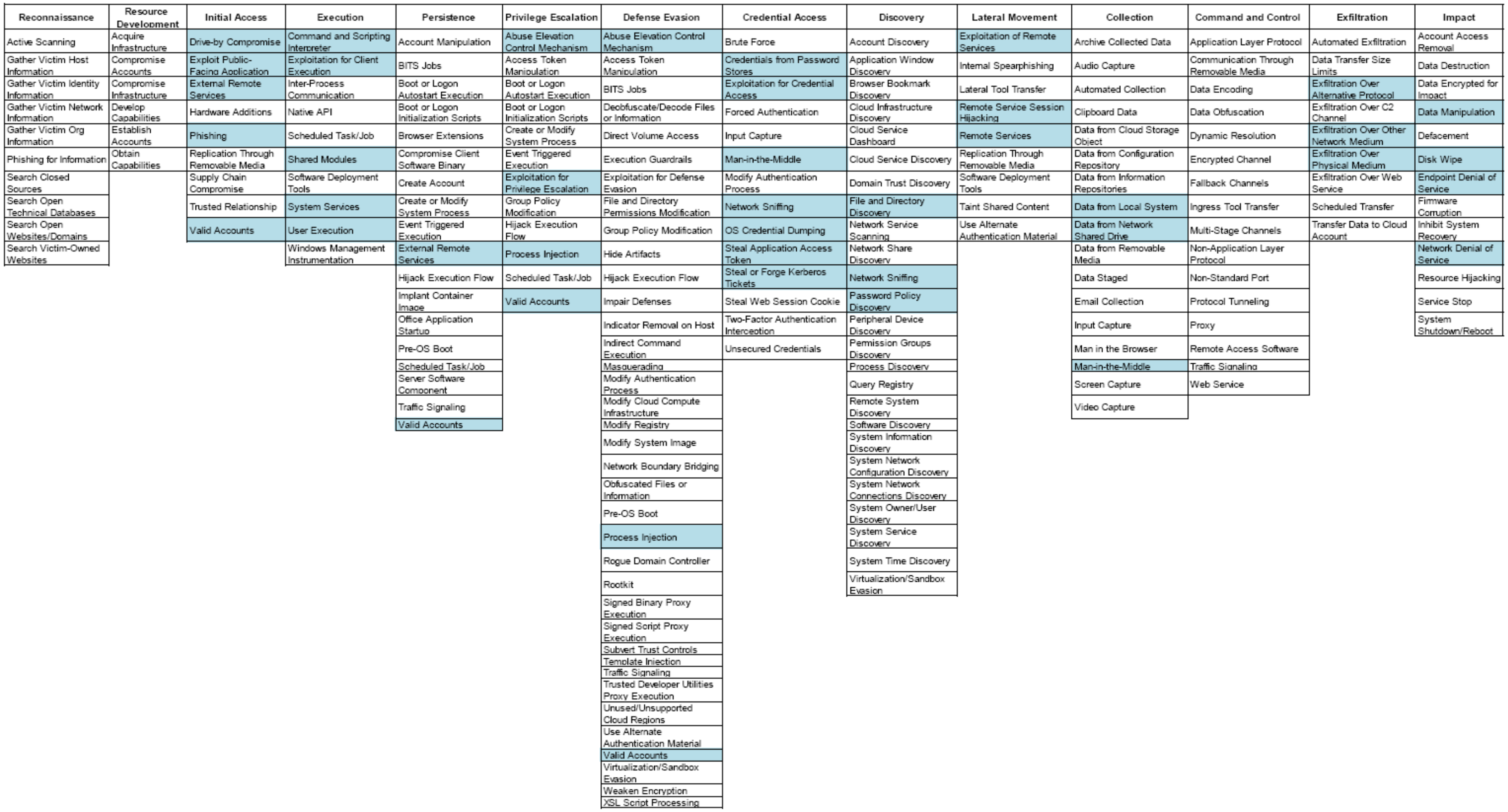
Techniques [15]	No. of Expressing SIRs	No. of Using Groups	No. of Using SW Tools	No. of Papers	Papers with Expressing SIRs	No. of Citations	Average No. of Citations
External Remote Services	1	17	4	1	[62]	50	50
Phishing	1	98	31	1	[31]	57	57
Command and Scripting Interpreter	1	185	351	1	[64]	47	47
Shared Modules	1	0	11	1	[47]	6	6
System Services	1	11	35	1	[82]	3	3
Abuse Elevation Control Mechanism	1	9	30	1	[31]	57	57
File and Directory Discovery	1	34	170	1	[74]	0	0
Exploitation of Remote Services	1	5	9	1	[69]	9	9
Remote Service Session Hijacking	1	0	1	1	[82]	3	3
Exfiltration Over Physical Medium	1	2	5	1	[74]	0	0
Disk Wipe	1	5	6	1	[81]	4	4
Drive-by Compromise	2	21	6	2	[62] [84]	51	26
Process Injection	2	24	112	2	[82] [69]	12	6
Credentials from Password Stores	2	31	73	1	[84]	1	1
Exploitation for Credential Access	2	0	0	1	[31]	57	57
Steal Application Access Token	2	1	0	1	[75]	8	8
Exploit Public-Facing Application	3	14	3	3	[31] [82] [84]	61	20
Valid Accounts	3	47	13	2	[31] [84]	58	29
Remote Services	3	52	42	3	[82] [75] [69]	20	7
Exfiltration Over Alternative Protocol	3	9	17	1	[83]	0	0
Data Manipulation	3	4	4	2	[82] [83]	3	2
Endpoint Denial-of-Service	3	1	2	2	[47] [83]	6	3
User Execution	4	86	49	3	[31] [47] [84]	64	21
OS Credential Dumping	5	69	58	3	[31] [83] [84]	58	19
Password Policy Discovery	5	3	5	1	[47]	6	6
Steal or Forge Kerberos Tickets	6	4	7	4	[31] [75] [83] [84]	66	17
Exfiltration over Other Network Medium	6	0	1	1	[84]	1	1
Data from Network Shared Drive	7	6	4	3	[31] [47] [58]	65	22
Data from Local System	9	28	60	4	[31] [82] [84] [56]	64	16
Network Denial of Service	9	1	1	3	[47] [83] [84]	7	2
Exploitation for Client Execution	11	28	11	5	[31] [47] [83] [84] [57]	102	20
Network Sniffing	11	6	9	3	[47] [75] [84]	15	5
Exploitation for Privilege Escalation	13	11	10	8	[31] [47] [62] [64] [69] [83] [58] [81]	175	22
Man-in-the-Middle	17	3	5	3	[47] [75] [84]	15	5
Total		811	1,145				

In this paper, we surveyed the 938 academic publications mentioning MulVAL and identified 38 papers extending MulVAL. To improve the usefulness of MulVAL, we provide the list of all MulVAL interaction rules, which can work together to enable broader risk assessment. To evaluate the extent to which MulVAL rules can represent different attacks, we mapped all of the MulVAL rules to MITRE ATT&CK Techniques and summarized the attack coverage capabilities provided by the MulVAL rules.

Mapping between the most commonly used attack graph generation tools, such as MulVAL, and the MITRE ATT&CK threat model will enable security administrators to handle more realistic attack scenarios. A clear understanding of an existing network's strength against different types of TTPs is critical, and the simulation of MITRE-based attack scenarios enables such understanding. For example, this can help security administrators decide which defensive measures to implement.

C. MAIN CHALLENGES

The main challenge we faced while conducting this survey was the lack of standard terminology across the published MulVAL extensions. For example, the meaning of **User**



Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Active Scanning	Acquire Infrastructure	**Drive-by Compromise**	**Command and Scripting Interpreter**	Account Manipulation	**Abuse Elevation Control Mechanism**	**Abuse Elevation Control Mechanism**	Brute Force	Account Discovery	**Exploitation of Remote Services**	Archive Collected Data	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Gather Victim Host Information	Compromise Accounts	**Exploit Public-Facing Application**	**Exploitation for Client Execution**	BITS Jobs	Access Token Manipulation	Access Token Manipulation	**Credentials from Password Stores**	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information	Compromise Infrastructure	**External Remote Services**	Inter-Process Communication	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	BITS Jobs	**Exploitation for Credential Access**	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding	**Exfiltration Over Alternative Protocol**	Data Encrypted for Impact
Gather Victim Network Information	Develop Capabilities	Hardware Additions	Native API	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Infrastructure Discovery	**Remote Service Session Hijacking**	Clipboard Data	Data Obfuscation	Exfiltration Over C2 Channel	**Data Manipulation**
Gather Victim Org Information	Establish Accounts	**Phishing**	Scheduled Task/Job	Browser Extensions	Create or Modify System Process	Direct Volume Access	Input Capture	Cloud Service Dashboard	**Remote Services**	Data from Cloud Storage Object	Dynamic Resolution	**Exfiltration Over Other Network Medium**	Defacement
Phishing for Information	Obtain Capabilities	Replication Through Removable Media	**Shared Modules**	Compromise Client Software Binary	Event Triggered Execution	Execution Guardrails	**Man-in-the-Middle**	Cloud Service Discovery	Replication Through Removable Media	Data from Configuration Repository	Encrypted Channel	**Exfiltration Over Physical Medium**	**Disk Wipe**
Search Closed Sources		Supply Chain Compromise	Software Deployment Tools	Create Account	**Exploitation for Privilege Escalation**	Exploitation for Defense Evasion	Modify Authentication Process	Domain Trust Discovery	Software Deployment Tools	Data from Information Repositories	Fallback Channels	Exfiltration Over Web Service	**Endpoint Denial of Service**
Search Open Technical Databases		Trusted Relationship	**System Services**	Create or Modify System Process	Group Policy Modification	File and Directory Permissions Modification	**Network Sniffing**	**File and Directory Discovery**	Taint Shared Content	**Data from Local System**	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains		**Valid Accounts**	**User Execution**	Event Triggered Execution	Hijack Execution Flow	Group Policy Modification	**OS Credential Dumping**	Network Service Scanning	Use Alternate Authentication Material	**Data from Network Shared Drive**	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			Windows Management Instrumentation	**External Remote Services**	**Process Injection**	Hide Artifacts	**Steal Application Access Token**	Network Share Discovery		Data from Removable Media	Non-Application Layer Protocol		**Network Denial of Service**
				Hijack Execution Flow	Scheduled Task/Job	Hijack Execution Flow	**Steal or Forge Kerberos Tickets**	**Network Sniffing**		Data Staged	Non-Standard Port		Resource Hijacking
				Implant Container Image	**Valid Accounts**	Impair Defenses	Steal Web Session Cookie	**Password Policy Discovery**		Email Collection	Protocol Tunneling		Service Stop
				Office Application Startup		Indicator Removal on Host	Two-Factor Authentication Interception	Peripheral Device Discovery		Input Capture	Proxy		System Shutdown/Reboot
				Pre-OS Boot		Indirect Command Execution	Unsecured Credentials	Permission Groups Discovery		Man in the Browser	Remote Access Software		
				Scheduled Task/Job		Masquerading		Process Discovery		**Man-in-the-Middle**	Traffic Signaling		
				Server Software Component		Modify Authentication Process		Query Registry		Screen Capture	Web Service		
				Traffic Signaling		Modify Cloud Compute Infrastructure		Remote System Discovery		Video Capture			
				Valid Accounts		Modify Registry		Software Discovery					
						Modify System Image		System Information Discovery					
						Network Boundary Bridging		System Network Configuration Discovery					
						Obfuscated Files or Information		System Network Connections Discovery					
						Pre-OS Boot		System Owner/User Discovery					
						Process Injection		System Service Discovery					
						Rogue Domain Controller		System Time Discovery					
						Rootkit		Virtualization/Sandbox Evasion					
						Signed Binary Proxy Execution							
						Signed Script Proxy Execution							
						Subvert Trust Controls							
						Template Injection							
						Traffic Signaling							
						Trusted Developer Utilities Proxy Execution							
						Unused/Unsupported Cloud Regions							
						Use Alternate Authentication Material							
						Valid Accounts							
						Virtualization/Sandbox Evasion							
						Weaken Encryption							
						XSL Script Processing							

FIGURE 8. MITRE ATT&CK Enterprise Matrix - Techniques expressed by MulVAL interaction rules (highlighted).

differs among the published papers – it may relate to the configured host account or the logical user principal. We also found that since different researchers generated the MulVAL extensions, there are some duplicate rules. In addition, the MulVAL-related articles do not relate the proposed interaction rules to MITRE TTPs. Mapping the rules to the most appropriate ATT&CK Techniques posed an additional challenge.

D. FUTURE WORK

In future work, we intend to normalize the MulVAL rules, removing interaction rules that were defined more than once with different names, different parameter names, or a different order of parameters. In addition, we plan to propose a methodology for expressing arbitrary ATT&CK Techniques using MulVAL interaction rules. A grand challenge would be modeling the entire known attack scenario, e.g., all the ATT&CK Techniques, to interaction rules. This will enhance MulVAL's ability to provide realistic network risk assessment. The following milestones on the MulVAL development road-map may be MITRE ATT&CK Mobile and ICS Techniques. Finally, this MulVAL extension development would highly benefit from automation in the interaction rule generation process.

APPENDIX

Expressed ATT&CK Techniques as a Matrix. See figure 8.

REFERENCES

[1] E. Johns, "Cyber security breaches survey 2020," Dept. Digit., Culture, Media Sport, UK Government, Tech. Rep., 2020. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/893399/Cyber_Security_Breaches_Survey_2020_Statistical_Release_180620.pdf

[2] S. Furnell, H. Heyburn, A. Whitehead, and J. N. Shah, "Understanding the full cost of cyber security breaches," *Comput. Fraud Secur.*, vol. 2020, no. 12, pp. 6–12, Jan. 2020.

[3] D. J. Landoll and D. Landoll, The Security Risk Assessment Handbook: A Complete Guide for Performing Security Risk Assessments. Boca Raton, FL, USA: CRC Press, 2005.

[4] V. Katta, P. Karpati, A. L. Opdahl, C. Raspotnig, and G. Sindre, "Comparing two techniques for intrusion visualization," in *Proc. IFIP Work. Conf. Pract. Enterprise Model.* Cham, Switzerland: Springer, 2010, pp. 1–15.

[5] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Lead. Issues Inf. Warfare Secur. Res.*, vol. 1, no. 1, p. 80, 2011.

[6] D. F. Haasl, N. H. Roberts, W. E. Vesely, and F. F. Goldberg, "Fault tree handbook," Nuclear Regulatory Commission, Washington, DC, USA, Tech. Rep., NUREG-0492, 1981.

[7] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proc. Workshop New Secur. Paradigms*, Jan. 1998, pp. 71–79.

[8] A. Sabur, A. Chowdhary, D. Huang, and A. Alshamrani, "Toward scalable graph-based security analysis for cloud networks," *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108795.

[9] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Oct. 2006, pp. 336–345.

[10] B. Schneier, "Attack trees," *Dr. Dobb's J.*, vol. 24, no. 12, pp. 21–29, 1999.

[11] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proc. IEEE Symp. Secur. Privacy*, May 2002, pp. 273–284.

[12] S. Noel, S. Jajodia, B. O'Berry, and M. Jacobs, "Efficient minimum-cost network hardening via exploit dependency graphs," in *Proc. 19th Annu. Comput. Secur. Appl. Conf.*, Dec. 2003, pp. 86–95.

[13] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in *Proc. 22nd Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Dec. 2006, pp. 121–130.

[14] X. Ou, W. F. Boyer, and S. Zhang, "MulVAL: A logic-based enterprise network security analyzer," in *Proc. 14th USENIX Secur. Symp.*, 2013, pp. 1–16.

[15] MITRE. (2022). *Mitre Att&ck Web Site*. [Online]. Available: https://attack.mitre.org/

[16] J. B. Hong, D. S. Kim, C.-J. Chung, and D. Huang, "A survey on the usability and practical applications of graphical security models," *Comput. Sci. Rev.*, vol. 26, pp. 1–16, Nov. 2017.

[17] L. Wang, J. Sushil, S. Anoop, P. Cheng, and S. Noel, "*k*-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 1, pp. 30–44, Feb. 2013.

[18] S. Noel and S. Jajodia, "Optimal IDS sensor placement and alert prioritization using attack graphs," *J. Netw. Syst. Manage.*, vol. 16, no. 3, pp. 259–275, Sep. 2008.

[19] S. Roschke, F. Cheng, and C. Meinel, "High-quality attack graph-based IDS correlation," *Log. J. IGPL*, vol. 21, no. 4, pp. 571–591, Aug. 2013.

[20] C. Liu, A. Singhal, and D. Wijesekera, "Using attack graphs in forensic examinations," in *Proc. 7th Int. Conf. Availability, Rel. Secur.*, Aug. 2012, pp. 596–603.

[21] L. Wang, S. Noel, and S. Jajodia, "Minimum-cost network hardening using attack graphs," *Comput. Commun.*, vol. 29, no. 18, pp. 3812–3824, 2006.

[22] R. Horne, S. Mauw, and A. Tiu, "Semantics for specialising attack trees based on linear logic," *Fundamenta Informaticae*, vol. 153, nos. 1–2, pp. 57–86, Jun. 2017.

[23] R. Kumar and M. Stoelinga, "Quantitative security and safety analysis with attack-fault trees," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2017, pp. 25–32.

[24] B. Fila and W. Wideł, "Exploiting attack–defense trees to find an optimal set of countermeasures," in *Proc. IEEE 33rd Comput. Secur. Found. Symp. (CSF)*, Jun. 2020, pp. 395–410.

[25] H. Nishihara, Y. Kawanishi, D. Souma, and H. Yoshida, "On validating attack trees with attack effects," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2020, pp. 309–324.

[26] P. H. Meland, D. G. Spampinato, E. Hagen, E. T. Baadshaug, K.-M. Krister, and K. S. Velle, "SeaMonster: Providing tool support for security modeling," in *Proc. Norsk informasjonssikkerhetskonferanse*, 2008, pp. 1–10.

[27] L. P. Swiler, C. Phillips, and T. Gaylor, "A graph-based network-vulnerability analysis system," Sandia Nat. Labs., Albuquerque, NM, USA, Tech. Rep., SAND-97-3010/1, 1998.

[28] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, Nov. 2002, pp. 217–224.

[29] S. Jajodia, S. Noel, and B. O'berry, "Topological analysis of network attack vulnerability," in *Proc. 2nd ACM Symp. Inf., Comput. Commun. Secur.*, Mar. 2007, pp. 247–266.

[30] *The CyVision Pedigree*, Cyvision, San Jose, CA, USA, 2016.

[31] X. Ou and A. W. Appel, *A Logic-Programming Approach to Network Security Analysis*. Princeton, NJ, USA: Princeton Univ. Princeton, 2005.

[32] Y. Liu and H. Man, "Network vulnerability assessment using Bayesian networks," *Proc. SPIE*, vol. 5812, pp. 61–71, Mar. 2005.

[33] M. L. Artz, "Netspa: A network security planning architecture," Ph.D. thesis, Massachusetts Inst. Technol., Cambridge, MA, USA, 2002.

[34] *Firemon Risk Analyzer*, FireMon, Overland Park, KS, USA.

[35] H. Holm, T. Sommestad, M. Ekstedt, and L. NordströM, "CySeMoL: A tool for cyber security analysis of enterprises," in *Proc. 22nd Int. Conf. Exhib. Electr. Distribution (CIRED)*, 2013, pp. 1–4.

[36] H. Holm, M. Ekstedt, T. Sommestad, and M. Korman, "A manual for the cyber security modeling language," Roy. Inst. Technol. (KTH), Stockholm, Sweden, Tech. Rep., 2013.

[37] S. Noel, E. Harley, K. H. Tam, M. Limiero, and M. Share, "CyGraph: Graph-based analytics and visualization for cybersecurity," *Handbook of Statistics*, vol. 35. Oxford, U.K.: Elsevier, 2016, ch. 4, pp. 117–167.

[38] A. Nadeem, S. Verwer, S. Moskal, and S. J. Yang, "Alert-driven attack graph generation using S-PDFA," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 731–746, Apr. 2022.

[39] Z. Li, J. Zeng, Y. Chen, and Z. Liang, "AttacKG: Constructing technique knowledge graph from cyber threat intelligence reports," 2021, *arXiv:2111.07093*.

[40] T. Li, Y. Jiang, C. Lin, M. S. Obaidat, Y. Shen, and J. Ma, "DeepAG: Attack graph construction and threats prediction with bi-directional deep learning," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 740–757, Feb. 2023.

[41] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100219.

[42] *NVD: National Vulnerability Database*, NIST, Gaithersburg, MD, USA, 2022.

[43] *Vulnerability Database*, Vuldb, Zürich, Switzerland, 1970.

[44] *Whitesource Vulnerability Database*, Whitesource Software, New York, NY, USA, 2023.

[45] *The Nessus Project*, Renaud Deraison, New York, NY, USA, 2022.

[46] J. Baker, M. Hansbury, and D. Haynes, *The OVAL Language Specification*. Bedford, MA, USA: MITRE, 2011.

[47] O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, and A. Shabtai, "Extending attack graphs to represent cyber-attacks in communication protocols and modern IT networks," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1936–1954, May 2022.

[48] P. Mensah, "Generation and dynamic update of attack graphs in cloud providers infrastructures," Ph.D. thesis, CentraleSupélec, France, 2019.

[49] M. Albanese, N. Cooke, G. Coty, D. Hall, C. Healey, S. Jajodia, P. Liu, M. D. McNeese, P. Ning, D. Reeves, and V. S. Subrahmanian, "Computer-aided human centric cyber situation awareness," in *Theory and Models for Cyber Situation Awareness*. Germany: Springer, 2017, pp. 3–25.

[50] D. Malzahn, Z. Birnbaum, and C. Wright-Hamor, "Automated vulnerability testing via executable attack graphs," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Secur.)*, Jun. 2020, pp. 1–10.

[51] T. Wang, Q. Lv, B. Hu, and D. Sun, "CVSS-based multi-factor dynamic risk assessment model for network system," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 289–294.

[52] P. Rao, K. Sagonas, T. Swift, D. S. Warren, and J. Freire, "XSB: A system for efficiently computing well-founded semantics," in *Proc. Int. Conf. Log. Program. Nonmonotonic Reasoning*. Cham, Switzerland: Springer, 1997, pp. 430–440.

[53] S. Yi, Y. Peng, Q. Xiong, T. Wang, Z. Dai, H. Gao, J. Xu, J. Wang, and L. Xu, "Overview on attack graph generation and visualization technology," in *Proc. Int. Conf. Anti-Counterfeiting, Secur. Identificat. (ASID)*, Oct. 2013, pp. 1–6.

[54] E. Bacic, M. Froh, and G. Henderson, "MulVAL extensions for dynamic asset protection," Cinnabar Netw. Inc. Ottawa, Ottawa, ON, USA, 2006.

[55] S. Bhatt, W. Horne, J. Pato, R. Rajagopalan, and P. Rao, "Model-based validation of enterprise access policies," *Nato Secur. Through Sci. Ser. D-Inf. Commun. Secur.*, vol. 2, p. 107, Jan. 2006.

[56] S. Govindavajhala, "Status of the MulVAL project," Dept. Comput. Sci., Princeton Univ., Princeton, NJ, USA, Tech. Rep., TR-743-06, 2006.

[57] S. Govindavajhala and A. W. Appel, "Windows access control demystified," Secure Internet Program. Lab., Princeton Univ., Princeton, NJ, USA, Tech. Rep., 2006.

[58] S. Govindavajhala, "A formal approach to practical network security management," Princeton Univ., Princeton, NJ, USA, Tech. Rep., 2006.

[59] S. Govindavajhala and A. W. Appel, "Automatic configuration vulnerability analysis," Dept. Comput. Sci., Tech. Rep., TR-773-07, 2007.

[60] J. Homer, X. Ou, and M. A. McQueen, "From attack graphs to automated configuration management—An iterative approach," Kansas State Univ. Manhattan, KS, USA, Tech. Rep., 2008.

[61] D. Saha, "Extending logical attack graphs for efficient vulnerability analysis," in *Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS)*, 2008, pp. 63–74.

[62] X. Ou and A. Singhal, *Quantitative Security Risk Assessment of Enterprise Networks*. Germany: Springer, 2011.

[63] H. Almohri, "High assurance models for secure systems," Ph.D. thesis, Virginia Tech, Blacksburg, VA, USA, 2013.

[64] H. M. J. Almohri, L. T. Watson, D. Yao, and X. Ou, "Security optimization of dynamic networks with probabilistic graph modeling and linear programming," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 4, pp. 474–487, Jul. 2016.

[65] C. Liu, A. Singhal, and D. Wijesekera, "A model towards using evidence from security events for network attack analysis," in *Proc. WOSIS*, 2014, pp. 83–95.

[66] C. Liu, A. Singhal, and D. Wijesekera, "Relating admissibility standards for digital evidence to attack scenario reconstruction," *J. Digit. Forensics, Secur. Law*, vol. 9, no. 2, p. 181, 2014.

[67] C. Liu, A. Singhal, and D. Wijesekera, "A logic-based network forensic model for evidence analysis," in *Proc. IFIP Int. Conf. Digit. Forensics*. Cham, Switzerland: Springer, 2015, pp. 129–145.

[68] C. Liu, "A logic-based network forensic model for evidence analysis," Ph.D. thesis, George Mason Univ., Fairfax, VA, USA, 2015.

[69] X. Sun, J. Dai, A. Singhal, and P. Liu, "Inferring the stealthy bridges between enterprise network islands in cloud using cross-layer Bayesian networks," in *Proc. Int. Conf. Secur. Privacy Commun. Netw.* Cham, Switzerland: Springer, 2014, pp. 3–23.

[70] X. Sun, J. Dai, A. Singhal, and P. Liu, "Probabilistic inference of the stealthy bridges between enterprise networks in cloud," *ICST Trans. Secur. Saf.*, vol. 4, no. 13, Jan. 2018, Art. no. 153526.

[71] J. Sembiring, M. Ramadhan, Y. S. Gondokaryono, and A. A. Arman, "Network security risk analysis using improved MulVAL Bayesian attack graphs," *Int. J. Electr. Eng. Informat.*, vol. 7, no. 4, pp. 735–753, Dec. 2015.

[72] S. Jilcott, "Securing the supply chain for commodity IT devices by automated scenario generation," in *Proc. IEEE Int. Symp. Technol. for Homeland Secur. (HST)*, Apr. 2015, pp. 1–6.

[73] X. Dong, S. Jauhar, W. G. Temple, B. Chen, Z. Kalbarczyk, W. H. Sanders, N. O. Tippenhauer, and D. M. Nicol, "Establishing common input scenarios for security assessment," Adv. Digit. Sci. Center, Singapore, Tech. Rep., 2023.

[74] X. Dong, S. Jauhar, W. G. Temple, B. Chen, Z. Kalbarczyk, W. H. Sanders, N. O. Tippenhauer, and D. Nicol, "The right tool for the job: A case for common input scenarios for security assessment," in *Proc. Int. Workshop Graph. Models Secur.* Cham, Switzerland: Springer, 2016, pp. 39–61.

[75] J. C. Acosta, E. Padilla, and J. Homer, "Augmenting attack graphs to represent data link and network layer vulnerabilities," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 1010–1015.

[76] J. T. W. Jing, L. W. Yong, D. M. Divakaran, and V. L. L. Thing, "Augmenting MulVAL with automated extraction of vulnerabilities descriptions," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2017, pp. 476–481.

[77] X. Sun, A. Singhal, and P. Liu, "Towards actionable mission impact assessment in the context of cloud computing," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*. Cham, Switzerland: Springer, 2017, pp. 259–274.

[78] B. M. Anderson, "Determining vulnerability using attack graphs: An expansion of the current fair model," M.S. thesis, Eastern Washington Univ., Cheney, WA, USA, 2018.

[79] C. Cao, L.-P. Yuan, A. Singhal, P. Liu, X. Sun, and S. Zhu, "Assessing attack impact on business processes by interconnecting attack graphs and entity dependency graphs," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*. Cham, Switzerland: Springer, 2018, pp. 330–348.

[80] P. Appana, X. Sun, and Y. Cheng, "What to do first: Ranking the mission impact graph for effective mission assurance," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2019, pp. 567–571.

[81] N. Khakpour, C. Skandylas, G. S. Nariman, and D. Weyns, "Towards secure architecture-based adaptations," in *Proc. IEEE/ACM 14th Int. Symp. Softw. Eng. Adapt. Self-Managing Syst. (SEAMS)*, May 2019, pp. 114–125.

[82] M. Inokuchi, Y. Ohta, S. Kinoshita, T. Yagyu, O. Stan, R. Bitton, Y. Elovici, and A. Shabtai, "Design procedure of knowledge base for practical attack graph generation," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Jul. 2019, pp. 594–601.

[83] L. Zhou, "Security risk analysis based on data criticality," Linnaeus Univ., Växjö, Sweden, Tech. Rep., 2020.

[84] M. McCormack, S. Chandrasekaran, G. Liu, T. Yu, S. DeVincent Wolf, and V. Sekar, "Security analysis of networked 3D printers," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 118–125.

[85] O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, and A. Shabtai, "Heuristic approach for countermeasure selection using attack graphs," in *Proc. IEEE 34th Comput. Secur. Found. Symp. (CSF)*. Washington, DC, USA: IEEE Comput. Soc., Jun. 2021, pp. 63–78.

[86] H. Binyamini, R. Bitton, M. Inokuchi, T. Yagyu, Y. Elovici, and A. Shabtai, "A framework for modeling cyber attack techniques from security vulnerability descriptions," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2574–2583.

[87] R. Bitton, N. Maman, I. Singh, S. Momiyama, Y. Elovici, and A. Shabtai, "Evaluating the cybersecurity risk of real world, machine learning production systems," 2021, *arXiv:2107.01806*.

[88] *MITRE Corporate Overview*, MITRE, McLean, VA, USA, 2022.

[89] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," MITRE, USA, Tech. Rep., 2018.

[90] K. Oosthoek and C. Doerr, "SoK: ATT&CK techniques and trends in windows malware," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Cham, Switzerland: Springer, 2019, pp. 406–425.

[91] P. Maynard and K. McLaughlin, "Big fish, little fish, critical infrastructure: An analysis of Phineas Fisher and the 'Hacktivist' threat to critical infrastructure," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Jun. 2020, pp. 1–7.

[92] N. Munaiah, A. Rahman, J. Pelletier, L. Williams, and A. Meneely, "Characterizing attacker behavior in a cybersecurity penetration testing competition," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Sep. 2019, pp. 1–6.

[93] K. Nickels, "Using ATT&CK to advance cyber threat intelligence," MITRE, USA, Tech. Rep., 2018.

[94] N. Shevchenko, "Threat modeling: 12 available methods," Carnegie Mellon Univ., USA, Tech. Rep., 2018. [Online]. Available: https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/

[95] M. S. Khan, S. Siddiqui, and K. Ferens, "A cognitive and concurrent cyber kill chain model," in *Computer and Network Security Essentials*. Cham, Switzerland: Springer, 2018, pp. 585–602.

[96] K. Nickels, "How to be a savvy ATT&CK consumer," MITRE, USA, Tech. Rep., 2019.

[97] M. S. Barik, A. Sengupta, and C. Mazumdar, "Attack graph generation and analysis techniques," *Defence Sci. J.*, vol. 66, no. 6, p. 559, Oct. 2016.

[98] S. Haque, M. Keffeler, and T. Atkison, "An evolutionary approach of attack graphs and attack trees: A survey of attack modeling," in *Proc. Int. Conf. Secur. Manag. (SAM)*, 2017, pp. 224–229.

[99] U. Garg, G. Sikka, and L. Aawsthi, "A systematic review of attack graph generation and analysis techniques," in *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives*. Boca Raton, FL, USA: CRC Press, 2018, pp. 115–146.

[100] W. He, H. Li, and J. Li, "Unknown vulnerability risk assessment based on directed graph models: A survey," *IEEE Access*, vol. 7, pp. 168201–168225, 2019.

[101] V. S. M. Legoy, "Retrieving ATT&CK tactics and techniques in cyber threat reports," M.S. thesis, Dept. Elect. Eng., Math., Comput. Sci., Univ. Twente, Enschede, The Netherlands, 2019.

[102] E. Aghaei and E. Al-Shaer, "ThreatZoom: Neural network for automated vulnerability mitigation," in *Proc. 6th Annu. Symp. Hot Topics Sci. Secur.*, Apr. 2019, pp. 1–3.

[103] R. Kwon, T. Ashley, J. Castleberry, P. Mckenzie, and S. N. G. Gourisetti, "Cyber threat dictionary using MITRE ATT&CK matrix and NIST cybersecurity framework mapping," in *Proc. Resilience Week (RWS)*, Oct. 2020, pp. 106–112.

[104] M. D. Purba, B. Chu, and E. Al-Shaer, "From word embedding to cyberphrase embedding: Comparison of processing cybersecurity texts," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Nov. 2020, pp. 1–6.

[105] G. Lee, S. Shim, B. Cho, T. Kim, and K. Kim, "Fileless cyberattacks: Analysis and classification," *ETRI J.*, vol. 43, no. 2, pp. 332–343, Apr. 2021.

DAVID TAYOURI received the B.Sc. and M.Sc. degrees in computer science. He is currently pursuing the Ph.D. degree with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev. His research interests include network security, the security of software, containers, intelligent transportation systems, and risk assessment with attack graphs.

NICK BAUM received the B.Sc. degree in computer science. He is currently pursuing the master's degree with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev.

ASAF SHABTAI is currently a Professor with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev. His research interests include computer and network security, machine learning, security of the IoT, smart mobile devices, and operational technology (OT) systems.

RAMI PUZIS is currently a Professor with the Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev. His research interests include network analysis with applications to security, social networks, communication, and biology.

• • •