Enhancing OT Threat Modelling: An Effective Rule-Based Approach for Attack Graph Generation

Enhancing OT Threat Modelling: An Effective Rule-Based Approach for Attack Graph Generation

Giulio Sunder*, Alberto Salvatore Colletto†, Sara Raimondi†, Cataldo Basile*, Alessio Viticchié†, Alessandro Aliberti*
*Politecnico di Torino, Turin, Italy. Email: name.surname@polito.it
†AlphaWaves S.r.l., Turin, Italy. Email: name.surname@awaves.it

***Abstract*—In today's data-driven world, the interconnection and automation of daily processes have become essential. As the demand for Internet connectivity grows, so does the need for robust cybersecurity measures. Operational Technology (OT), pivotal in controlling critical infrastructures such as power plants and water distribution systems, remains highly vulnerable. Many OT systems still rely on 'air gaps' for security, a measure increasingly insufficient as more systems connect to the internet for remote operation and data analysis. This article addresses the critical need for enhanced OT security solutions by introducing a novel tool focused on intelligent systems for the effective detection of cyber-attacks. The tool automates the creation of attack graphs and extracts attack paths from a JSON file describing the OT network. Leveraging the MulVAL attack graph generation engine, it provides a comprehensive visualization of potential attack vectors, enhancing the capability to identify and mitigate security threats in OT environments.**

***Index Terms*—Cybersecurity, Industrial Internet Connectivity, Operational Technology, Attack Graphs, Vulnerability Assessment**

I. Introduction

In today's fast-paced, data-ruled world, the need for interconnection and automation of daily processes has become of the utmost importance. Modern societies increasingly rely on interconnected systems to enhance efficiency, streamline operations, and provide seamless user experiences across various domains. To keep up with this rapid transition, appropriate cybersecurity measures must be taken to cope with the wider attack surface exposed due to the growing demand for internet connectivity. The proliferation of internet-enabled devices and systems has exponentially increased the number of potential entry points for malicious actors, necessitating robust and adaptive security strategies.

With these premises, Operational Technology (OT) stands out as one of the main fields that lack the most in terms of cybersecurity. Many OT systems, responsible for controlling the functionality of critical infrastructures, such as power plants, water distribution networks, and manufacturing facilities, still base most of their security on the presence of an "air gap" (physical separation from the internet). This traditional approach is increasingly inadequate in the face of modern threats and the necessity for remote management and data integration. This reliance on air gaps underscores the significant effort required to secure these systems, particularly as many are now being connected to the internet to facilitate remote operation, real-time monitoring, and advanced data analysis. Consequently, there is a pressing need to develop and implement advanced cybersecurity measures tailored specifically to the unique requirements and vulnerabilities of OT environments.

This article introduces an innovative tool that leverages intelligent systems for effective detection of cyber-attacks. Specifically, it offers an easy-to-use solution capable of automatically generating attack graphs and extracting potential attack paths within OT networks to identify vulnerable assets. This tool simplifies and streamlines threat modelling activities, making it accessible even to those with limited cybersecurity expertise. The need for such a tool arises from the challenges of performing manual security threat assessments, which can be highly time-consuming and error-prone. Manual assessments often require extensive knowledge and experience, and even then, the complexity and scale of modern OT networks can lead to oversight and inaccuracies. By automating the creation of attack graphs and the extraction of potential attack paths, this tool significantly reduces the time and effort required for comprehensive security analysis. The only effort required by the user is to provide a detailed description of the infrastructure and its network. Furthermore, the tool is designed to easily scale to accommodate larger and more complex OT systems, making it a versatile solution for various industrial applications. This capability is critical as OT networks continue to grow and evolve, integrating more devices and systems that need to be secured. Indeed, automating threat modeling activities helps reduce costs, while ensuring that security measures keep pace with the rapid advancements and increasing complexity of OT environments and related vulnerabilities.

As far as the state of the art goes, there have already been several attempts to provide solutions for automatic attack graph building. Out of all the available options, this paper will mainly focus on MulVAL, the chosen attack graph generator for this case study. Other alternatives are discussed in the related works section. However, although MulVAL is a good starting point, it presents some shortcomings that should be addressed, as i) the needs to be extended to better describe OT-related scenarios and attacks; ii) the output attack graph scales very quickly in terms of complexity, making it difficult to understand and navigate and iii) the input usually provided to the tool, even if obtained with vulnerability scanners, do not include information about network topology, which is critical to deduce graphs for sophisticated attacks.

The rest of the paper is organized as follows. Section II

979-8-3503-5477-5/24/$31.00 ©2024 IEEE

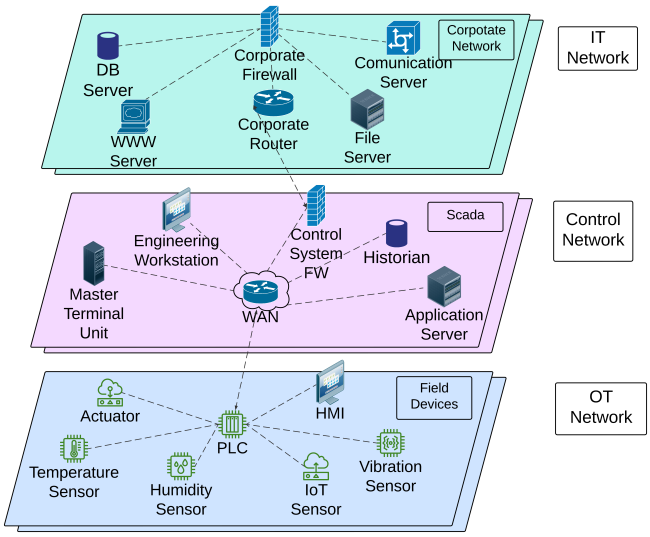

Fig. 1. Example of a standard hybrid IT-OT layered network.

provides an overview of Operational Technology (OT) and its associated security challenges, introducing key concepts and the current state of the art necessary to understand the subsequent sections. Section III details the overall architecture and information flow of the proposed tool, alongside a step-by-step description of the methodology used in its development. Section IV presents the configuration of the test networks used to evaluate the tool's effectiveness and discusses the results, highlighting some of the detected attack paths. Section V reviews and analyzes other research papers on attack graph building, discussing their strengths and weaknesses. Finally, Section VI summarizes the work, reflecting on its contributions and providing insights into potential future developments.

II. Background

The main differences between Operation Technology (OT) and Information Technology (IT) networks can be found in their primary purposes, architectures, security requirements, and operational priorities.

According to the official guidelines expressed by NIST [1], OT systems consist of combinations of control components (e.g., electrical, mechanical, hydraulic, pneumatic) that act together to achieve an objective (e.g., manufacturing, transportation of matter or energy), as depicted in Fig. 1. OT network architectures often involve specialized hardware and real-time operating systems tailored for controlling and automating industrial processes. These systems focus on ensuring the reliability and safety of physical operations, emphasizing protection against threats that could disrupt processes, cause physical damage, or endanger human lives. For instance, an attack on an OT system controlling a power plant could lead to significant physical harm and operational disruption. In contrast, IT systems prioritize data confidentiality, integrity, and availability, aiming to protect against cyber threats such as data breaches, unauthorized access, and data loss. While IT systems can often tolerate certain delays and downtime for maintenance or upgrades, OT systems require high availability and real-time response to ensure continuous and efficient operation of physical processes. Downtime in OT environments can lead to significant operational and safety risks, including financial loss, environmental damage, and threats to human safety.

Today's operational technology has significantly evolved due to the integration of IT capabilities into existing physical systems, often replacing or enhancing traditional physical control mechanisms [2]. Indeed, embedded digital controls have supplanted analogue mechanical controls in rotating machines and engines. While this evolution increases the connectivity and importance of these systems, it also needs greater adaptability, resilience, safety, and security. IT capabilities in physical systems result in significant security implications. Consequently, engineering models and analyses are advancing to address these emergent properties, including interdependencies related to safety, security, privacy, and environmental impact.

Traditional manual security assessments are often time-consuming, error-prone, and require extensive expertise. On the other hand, automated solutions can streamline this process, providing consistent and accurate results while scaling to accommodate larger and more complex OT networks. Attack graphs provide a visual representation of potential attack paths within a network, highlights the ways an attacker could compromise system assets and a hierarchical representation of the overall system security. By automating the creation of attack graphs and extracting potential attack paths, this work aims to simplify and enhance the security assessment process for OT environments.

A. Main OT network components

Operational technology architectures involve several critical components with distinct roles in ensuring systems' effective operation and management. Among these, the most important are i) Supervisory Control And Data Acquisition (SCADA), ii) Programmable Logic Controllers (PLC) and iii) Engineering Workstations [3].

Supervisory Control and Data Acquisition represent the central element of OT networks. They consist of software applications that gather and analyze real-time data from various industrial processes. SCADA systems provide operators with a comprehensive view of the operations, allowing them to monitor, control, and optimize processes from a centralized location. Hence, SCADA systems enable efficient decision-making by offering insights into system performance and potential issues.

Programmable Logic Controllers are robust industrial-grade computers designed for controlling machinery and processes. Unlike general-purpose machines, PLCs are built to withstand harsh industrial environments and offer real-time responsiveness. They execute pre-programmed instructions to manage tasks such as machinery operation, process control, and safety interlocks. PLCs are crucial in ensuring industrial processes' smooth and reliable operation.

Finally, the Engineering Workstations (EWS) serve as the interface for configuring and maintaining the various components within the OT network. Engineers use these workstations to develop, test, and deploy control strategies, create and modify system configurations, and troubleshoot issues. EWS provide the necessary tools for programming PLCs, setting up SCADA systems, and ensuring all components function seamlessly.

B. Need for Cybersecurity solutions

The evolution of OT systems, along with the introduction of IT capabilities, has clearly highlighted the need for adopting cybersecurity patterns and methodologies in industrial contexts [4]. Moreover, the complexity of OT systems, with their specialized hardware and custom software, require advanced automatic cybersecurity tools to prevent sophisticated threats and ensure continuous and safe operation. Automatic inference tools capable of elaborating potential attacks that may threaten a given OT infrastructure would simplify threat modelling and risk management activities.

As underlined by [5], attack graphs are one of the most effective methods for modelling complex threats and precisely identifying system weak points. Attack graphs sequentially represent vulnerabilities and actions an attacker could take to compromise an organization's assets. They have been analyzed in several works to date [6]–[9], as they allow for modelling security threats in a standardized way, which is a key requirement for automating the security assessment process. Extensive research has been carried out in this field, and various approaches to the problem have been proposed over the years as past surveys reported [10], [11]. Based on these surveys, the best approach for attack graph generation in terms of precision and effectiveness is represented by rule-based reasoning engines. Moreover, MulVAL is the best rule-based inference tool for extensibility, complexity, documentation availability, and cost.

C. MulVAL: rule-based inference engine

MulVAL is an open-source framework that supports the security assessment of a system through attack graph generation [8]. It is based on a rule inference engine, which uses model checking to conduct multi-host and multi-stage vulnerability analysis on a network. Attack graph generation is performed by MulVAL starting from a complete description of the infrastructure, which includes the following:

- *Hosts description*, information regarding hosts connected to the target network, such as hostnames, services, and protocols.
- *Network description*, information about network topology and hosts' reachability.
- *Theoretical vulnerabilities*: information about potentially existing issues, such as CVEs associated with hardware and software versions,
- *Attacker information*, attack entry points, needed privilege levels, and attack goals to test for.

The rule inference engine takes a set of *primitive facts* describing the target network and deduces a set of *derived facts* by applying *interaction rules*.

Although MulVAL has been selected as the best option for attack graph generation, it still has some limitations that could be improved. The main limitations considered in the scope of this work are listed below.

(***L1*** – *Low input interoperability*) MulVAL necessitates a full network description to effectively compute attack graphs, with extensive information about hosts, protocols, services, vulnerabilities and topology. However, the custom input clause description language required by MulVAL is complex and hard to produce by both automated network scanning tools and users. Therefore, the definition of a standardized and complete descriptive model is necessary. Indeed, a standardized descriptive model that can bring information in a form that could be easily generated (even manually) and consequently parsed into MulVAL primitives could significantly increase compatibility with automated tools and promote integration into pipelined processes.

(***L2*** – *Low network modeling capabilities*) The default MulVAL interaction rule set cannot capture complex network interactions and sophisticated network scenarios like ones from OT; this often results in imprecise or incomplete attack graphs.

(***L3*** – *Attack graph complexity*) The number of nodes of the generated graphs rapidly increases with the size of the input network. Hence, the attack graph size grows correspondingly, even though the amount of threat modelling information does not increase accordingly. Indeed, it often produces redundant attack paths, meaningless connections and overly complex graphs that result in poorly usable security assessments.

D. Enhancing MulVAL for OT

Our proposed methodology focuses on enhancing MulVAL attack graph generation. By addressing MulVAL's current limitations and improving its capabilities, we aim to produce attack graphs that comprehensively detail realistic attack paths for cybersecurity assessments in OT contexts. This enhancement will streamline analysis and reduce the complexity of generated outputs.

First, we have designed a formal model able to describe any aspect of OT networks and capture information regarding hosts, network topology, and reachability, thus improving MulVAL context knowledge awareness and interoperability. Hence, our approach integrates detailed network and attack goal descriptions into MulVAL. This ensures precise and complete attack graphs that simplify interpretation and aid in effective cybersecurity strategies. Nonetheless, it addresses limitation ***L1***. The network description has been provided manually for this research and the relative prototype. However, the preparation of the model instance can be made more user-friendly with graphical user interfaces and templates or automatic network discovery tools, which is irrelevant to this research.

Second, we addressed limitation ***L2*** by improving the attackers' description. We enriched the MulVAL rule set with

attack goal templates that describe a more extensive set of attacks that better fit the OT scenario.

Finally, we improved the effectiveness of the results to mitigate the risk of an excessive graph complexity that would hide errors or inconsistencies, thus facing limitation ***L3***. Our methodology supports automated analysis of attack paths, highlighting the most important attack information and mitigating manual errors. By improving MulVAL's analysis, we provide a robust tool to secure OT environments against evolving threats, aligning with the complex demands of modern OT systems.

III. Methodology

The proposed tool can be described as a pipelined architecture system consisting of three modules: an input parser, an attack graph generator, and an attack paths extractor. It takes as input the description of the target network, and produces as output an attack graph and a list of attack paths detailing the sequence of steps an attacker can take to achieve a specific goal within the system. The modules are detailed as follows

- The parser module is responsible for parsing the network description (as described in Section III-A) and translating it into a set of MulVAL clauses (explained in III-B)
- The second module is the MulVAL engine itself, which takes the clauses generated in the previous step, along with a set of extended interaction rules (described in III-B), and an attack goal template file containing queries listing the attack goals to be tested. It generates the attack graph corresponding to the specified goals.
- The attack path extractor traverses and prunes the generated attack graph from the previous step. It extracts the attack paths, detailing the sequence of attacker actions and relevant vulnerabilities.

A. Defining a network description model

For solving limitation ***L1***, we defined a formal description model to comprehensively represent any network aspects of interest for security, the *D-Model*. The model introduces a hierarchical tree representation that is easy to generate both automatically, through a combination of network scanning tools, and manually. The defined model is provided with a parser that can translate a network description into MulVAL clauses, allowing MulVAL to be easily integrated into an automated and pipelined environment. The model accepts a list of subnet descriptions, each reporting all relevant network details (e.g., subnet name, IP address, netmask). Each subnet node can report the array of network protocols in use within the subnet and an array of hosts belonging to the subnet. Each host node can report an array of services and an array of hosts that are reachable from that node. The model's root level can accept a list representing the open end-to-end communications sessions established between the hosts, and a list of all the potential vulnerabilities related to any item in network.

The model has been manually instantiated in JSON for our validation prototypes. However, the model has been designed to import data from automatic scanning tools.

B. Extending MulVAL interaction rules

As described in the background section, MulVAL relies on a set of interaction rules to deduce attack graphs. However, the default rules provided by MulVAL are inadequate for accurately representing OT systems (limitation ***L2***); they only link vulnerabilities to hosts, thus neglecting communication protocols, host interconnections and running services. Therefore, the extension of the existing interaction rule set is crucial, especially for OT network environments. For this work, we considered a set of rules that encompasses common network attacks and various communication types, such as wireless and wired communication, defined in a past work [12].

This extension enhances MulVAL's capability to represent complex scenarios, particularly in industrial contexts. Moreover, these extended rules enable the representation of vulnerabilities associated with protocols at both the link and application layer. This enhancement allows for modelling diverse attack techniques outlined in the MITRE ATT&CK framework [13], including Man-In-The-Middle (MITM) attacks, network traffic sniffing, Denial-of-Service (DoS) attacks, access control bypass.

Here are examples of primitive and derived facts that have been incorporated:

```
primitive(located(Host,Zone,Type)).
primitive(existingProtocol(Zone,Protocol)).
primitive(dataFlow(Src,Dst,FlowName)).
derived(l2Connection(Dev1,Dev2,LinkId,Prot,Type)).
derived(mitmLink(Principal,Src,Dst,SpoofHost)).
```

C. From D-Model to MulVAL clauses

MulVAL requires a set of clauses described using the Datalog language, a subset of Prolog. Therefore, a parser module is required to map system information from the D-Model into these MulVAL primitives. For the purpose of this work, the D-Model has been instantiated in JSON. The current parser module is implemented in Python and consists of six distinct functions, each responsible for parsing a subset of the information into its corresponding primitive format, as described in Section III-B.

A brief overview of the functionality of these parsing functions is provided next:

- `parse_subnets(sys_desc)` – this function is responsible for deducing subnet location from the D-Model and producing a set of the following primitive
  ```
  located(Host,Subnet,Type).
  ```
- `parse_hacl(sys_desc)` – it maps the access control list of each host to primitives of the following form
  ```
  hacl(SrcHost,DstHost,Prot,Port).
  ```
- `parse_services(sys_desc)` – it extracts all the services, from any host, and maps each of them to one of these primitives, whether the service is exposed to the network or only reachable from the machine
  ```
  networkService(Host,Service,Prot,Port,Priv).
  localService(Host,Service,Priv).
  ```
- `parse_l2_protocols(sys_desc)` – it parses the layer 2 protocols and produces
  ```
  existingProtocol(Subnet,Prot).
  ```

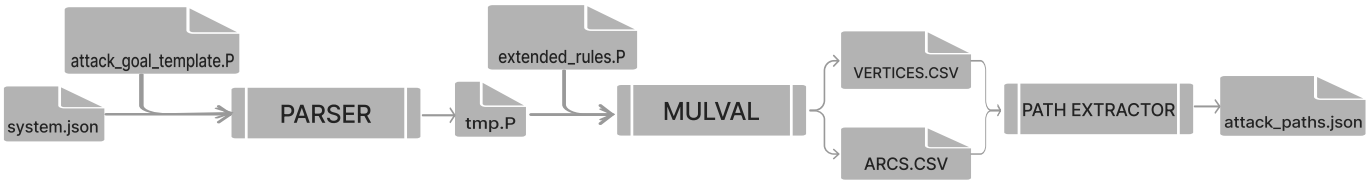

Fig. 2. General workflow of the tool

- `parse_data_flows(sys_desc)` – it parse the information about data communications across the network and maps each of them to the following couple of primitives

```
dataFlow(SrcHost,DstHost,FlowName).
flowBind(FlowName,Prot,Port).
```

- `parse_vulnerabilities(sys_desc)` – this function is in charge of producing the primitives for vulnerabilities description. The following are the types of primitive that could be generated, respectively, whether the vulnerability refers to a link layer protocol, an end-to-end protocol, a data flow or a service running on a host (`vulExists` and `vulProperty`).

```
vulLinkProtocol(Subnet,VulID,Prot,Range,Cons).
vulE2EProtocol(Src,Dst,VulID,Prot,Port,Range,Cons).
vulData(FlowName,VulID,VulName,Cons).
vulExists(Host,VulID,Service).
vulProperty(VulID,Range,Cons).
```

All generated clauses are appended to a file that includes the hypothetical attacker's position in the network and the list of attack goals to be tested by MulVAL. As interaction rules are expected to be expanded in the future to accurately represent new scenarios currently not covered, the parser may also require expansion to accommodate potential new primitive clauses added to MulVAL.

D. Attack graph analysis and pruning

One of the main downfalls of using MulVAL is the complexity of the generated attack graphs, which scale rapidly with the size of the considered network. Most of the nodes in the attack graph are transition state nodes generated by MulVAL from combining the provided interaction rules and do not actually carry information about attacks. Therefore, as part of this work, a module for wisely pruning the attack graph has been introduced in order to reduce complexity and keep only the information regarding attack steps and relative exploited vulnerabilities, thus mitigating limitation ***L2***. To achieve this goal, all the interaction rules corresponding to an attacker exploit have been labelled with an "ATTACKER_ACTION" tag (as shown in Listing 1), so that it would be easy to filter the nodes of the graph possessing it. The implementation of this module is based on a depth-first search algorithm applied to the attack graph. The search starts from each attack goal node and recursively works its way up through the parent nodes. If, at any point, a labelled node is reached, and it has not already been processed before, its information is stored in a new node, which is then inserted in the resulting pruned graph. In the same way, nodes corresponding to vulnerabilities are stored in the resulting graph, with the difference that these nodes' Datalog clauses start with the "vul" sub-string, therefore this condition is used as a filtering criterion. An illustrative version of the proposed algorithm can be found in Listing 2.

```
interaction_rule(
  (dos(Principal,DstHost):-
    networkService(DstHost,Prog,Prot,Port,User),
    hacl(SrcHost,DstHost,Prot,Port),
    vulExists(DstHost,VulID,Prog,remoteExp,dos),
    netAccess(Principal,SrcHost,DstHost,Prot,Port),
    malicious(Principal)),
  rule_desc('ATTACKER_ACTION Network based DoS', 1.0)).
```

Listing 1: Example of labelled interaction rule.

```
def prune_graph(attack_graph):
    pruned_graph = new graph
    for node in attack_graph:
        if node is attack_goal:
            search_paths_dfs(node, pruned_graph)
    return pruned_graph

def search_paths_dfs(node, pruned_graph):
    if node is exploit and node not in pruned_graph:
        pruned_graph.add(node)
    if node is vulnerability and node not in pruned_graph:
        pruned_graph.add(node)
    if node is leaf:
        return
    for parent in node.parents:
        search_paths_dfs(node, pruned_graph)
```

Listing 2: Illustrative version of the pruning algorithm.

E. Extracting attack paths from pruned graph

Once the attack graph is refined to include information only about attacks, vulnerabilities, and their dependencies, the next step involves extracting and categorizing all potential attack paths from the graph. To accomplish this, a second depth-first search procedure is conducted on the filtered graph. Starting from the attack goal nodes, the algorithm recursively explores the graph until it reaches leaf nodes. Vulnerability nodes encountered during this traversal are grouped with their corresponding exploit nodes. When a leaf node is reached, the identified path is saved as an ordered array of objects. Each object pertains to an exploit node and its associated vulnerabilities, where only the identifier of each vulnerability is stored. This enables cross-reference with the input system description discussed in Section III-A to retrieve detailed information as needed.

IV. Experimental Remarks

This section discusses the experimental evaluation of the proposed enhanced MulVAL tool for generating attack graphs in OT networks by presenting the experimental setup and discussing the evidence drawn.

A. Experiment setup

The reference network we considered for the experimental evaluation of the proposed solution is reported in Figure 3. The reference network is inspired by a use case we deployed in a real-world OT industrial plant. The network can be divided into PLC, SCADA, and Enterprise subnet.

The *PLC subnet* connects a SCADA supervisor running on a Hewlett-Packard PC with a series of Allen-Bradley PLCs. These devices communicate using the OPC-UA protocol via Ethernet, utilizing the OPC-UA server service. Each PLC directly connects to its respective HMI, providing a graphical representation of the processes the PLCs manage. Additionally, all PLCs communicate directly with the historian PC to send real-time data. The historian PC is used for real-time data analysis and hosts three main services: an MQTT broker, InfluxDB and Grafana. The MQTT broker service is responsible for receiving updates from the PLCs and notifying the other two services. InfluxDB is a high-performance time-series database optimized for storing and retrieving time-stamped data. Grafana, on the other hand, is an open-source platform that provides powerful data visualization capabilities. Together, they enable robust data analysis and visualization solutions, allowing for effective monitoring and decision-making based on real-time data.

The *SCADA subnet* manages communication between the SCADA system and all external connections through designated ports using the Ignition portal. This server software is the central hub for complete system integration on the plant floor. The same subnet also hosts an EWS personnel use to interact with the SCADA.

The *Enterprise subnet* is composed of a series of PCs associated with the company that maintains the site and can reach the SCADA subnet via VPN. It is located outside the firewall that protects the SCADA subnet. In this case study, we decided to include only one PC inside the network configuration of the Enterprise subnet because it is enough to model the enterprise network from a security behaviour point of view. Indeed, multiple instances of the same machine do not bring additional information but just redundant replications of the same element.

We considered two scenarios for the evaluation. The first scenario assumes the attacker's starting position to be within the SCADA subnet, allowing direct communication with the SCADA system and communication with the PLCs via the switch between the SCADA and the PLCs. The second scenario locates the attacker within the Enterprise subnet, thus requiring them to first bypass the firewall to exploit vulnerabilities in the SCADA system.

As introduced in Section III, the MulVAL module must have as input a list of potential vulnerabilities to be associated with the various nodes of the network under examination. Vulnerabilities collected include the CVE database and the list of issues related to common host misconfigurations we forged to enrich the issues' knowledge base.

Given this setup, we expect a set of possible attacks that involve crucial vulnerabilities as validation results. First, we expect attacks related to OPC-UA server vulnerabilities. Designated as CVE-2022-25304, this vulnerability allows remote attackers to exploit the OPC-UA server, resulting in a Denial-of-Service (DoS) condition, leading to production downtime and potential financial losses.

CVE-1999-0667 is related to the ARP protocol. This vulnerability presents a significant security risk as attackers could masquerade as legitimate network entities, potentially leading to unauthorized access, data interception or Man-in-the-Middle (MITM) attacks.

A vulnerability affecting the Ignition Portal service on the SCADA PC (CVE-2022-36126) enables remote attackers to exploit unauthorized access to administrative privileges on the Ignition web portal service, potentially leading to remote code execution and privilege escalation.

Additionally, vulnerabilities in low-level protocols such as Ethernet have been considered within the PLC subnet. Weaknesses in the implementation of the Ethernet protocol may enable adversaries to eavesdrop on network communications. Eavesdropping attacks seriously threaten data confidentiality, as sensitive information exchanged between devices on the PLC subnet could be intercepted and exploited by malicious actors. Furthermore, vulnerabilities exist in end-to-end (E2E) communication protocols utilized within the system, particularly the MQTT protocol. Also, these vulnerabilities allow for unauthorized eavesdropping due to an incorrect configuration of the protocol parameters.

Lastly, vulnerabilities in data flow security have been detected within the system, specifically in unencrypted data transmission from PLCs. These vulnerabilities expose sensitive data to potential interception via sniffing attacks, compromising the confidentiality and integrity of industrial processes.

In the second scenario, we also expect a possible remote exploit vulnerability related to a firewall located between the SCADA subnet and the Enterprise subnet. The consequences of this vulnerability are particularly concerning, as it could lead to access control bypass, allowing unauthorized entities to circumvent security measures and gain unauthorized access to critical system resources.

B. Discussion

As mentioned, the network configuration shown in Figure 3 has been tested for two different initial attacker locations: *attacker in SCADA subnet* and *attacker in Enterprise subnet*.

In the first case, the attacker supposedly has local access to the EWS placed in the SCADA subnet. This permits them only to communicate with the SCADA PC through the exposed Ignition web portal service. Hence, the attacker must be able to compromise the SCADA PC to reach the PLC subnet hosting all the PLCs and HMIs. This can be seen both in the extracted paths and in the reduced graph depicted in Figure 4, as all attacks must start with successfully exploiting the remote code execution vulnerability linked to the Ignition web portal service (CVE-2022-36126). Once this step has been achieved, the attacker has complete access to the SCADA PC. Therefore, he can leverage this to target devices in the PLC subnet, since

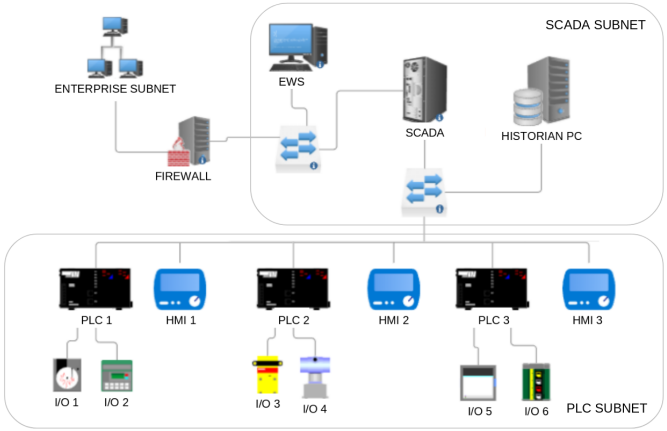

Fig. 3. Network configuration

they can now reach it through the SCADA PC. This opens the door to DoS (CVE-2022-25304), MITM (CVE-1999-0667) and eavesdropping attacks on PLCs, HMIs and historian PC.

In the second case, the attacker's supposed location is within the enterprise network. This adds an extra layer of security due to the presence of a firewall, which limits access to the SCADA subnet from the outside. For the sake of providing a proof of concept, a firewall misconfiguration has been introduced. If successfully exploited, it can lead to access control bypass, allowing the attacker to freely reach hosts within the SCADA subnet. Once the attacker has successfully bypassed the firewall restrictions, the attack scenario becomes equivalent to the first case; where the attacker was already inside the SCADA subnet.

For compactness, only the output graph corresponding to the second scenario, where the attacker is situated in the Enterprise network, is shown in Figure 4. The output graph for the first scenario would be identical to the one presented, excluding nodes 1, 8, and 9, which relate to firewall bypass. It is important to note that in MulVAL, primitive facts are represented as boxes, interaction rules as ovals, and derived facts (obtained from a combination of primitive and derived facts with interaction rules) as diamonds. After executing the pruning algorithm, the resulting graph contains information solely about exploit sequentiality and associated vulnerabilities. Indeed, the resulting graph does not report any interaction rule, as it reports redundant information that resumes relationships between host, vulnerability and exploits, which is already in the graph. Therefore, in Figure 4, diamond-shaped nodes represent exploits, while box-shaped nodes correspond to vulnerabilities. Hence, the simplest attack path representation is an alternated sequence of vulnerabilities and exploits.

The graph shows that the initial exploit from which all paths originate is the access control bypass node (node 8). This exploit allows the attacker to exploit a firewall misconfiguration (vulnerability node 1) to bypass firewall restrictions and gain access to the SCADA PC via HTTPS on port 443. Subsequently, the attacker proceeds to the code execution node (node 10) in the graph, which results in remote code execution and privilege escalation on the SCADA PC through a vulnerability in the Ignition web portal (node 2). With complete access to the SCADA PC, as depicted in the graph, the attacker can execute various attacks such as Man-in-the-Middle (nodes 13, 14), Denial-of-service (node 12), and traffic sniffing (node 11).

Many aspects of the improvements made by our solution are important to discuss. D-Model and the Parser module allow the user provide MulVAL input in a unified simple data structure. It reduces the intricacy of generating MulVAL clauses to extensively describe any aspect of the network to achieve the equivalent information granularity. It also improves MulVAL interoperability, thus, mitigating limitation ***L1***.

The introduced extended interaction rule set improves MulVAL's network descriptive capabilities. In fact, the output produced by MulVAL, using the default rule set, misses many vulnerabilities and exploits that we expect to exist and effectively impact the network. Considering the first network scenario, using the default interaction rule set defined by MulVAL, the attack graph produced cannot represent MitM (Man-in-the-Middle) and accessDataFlow exploits, because they are not described as rules in the default MulVAL rule set. The situation is even worse in the second scenario: MulVAL does not produce any attack graph because all the exploits strongly depend to accessControlBypass exploit that is not listed. Then, it vastly increases MulVAL network modelling capabilities, thus mitigating limitation ***L2***.

Finally, generating a more compact and human-readable attack graph was impossible without the Path Extractor module. In both network scenarios, our model removes nearly 80% of nodes and nearly 90% of all edges as they are helpful for graph generation but they do not bring significant, thus redundant, information. Thus, it validates the improvements brought by the proposed tool in mitigating limitation ***L3***.

Although we validate only against one network scenario, we assume that the proposed solution is valid and improves the state of the art in general, multi-layer OT network scenarios, given its granular descriptive power and general algorithmic approach, which is not tied to any specific network constraint and fo not include any pre-defined rule. Indeed, the proposed tool makes MulVAL reason only on provided inputs that are generated, time by time, ad-hoc for each new network scenario of interest.

V. Related Works

With the rise of cyber attacks, cybersecurity risk assessment has become crucial. Since 1998, attack graphs have been a significant focus in cybersecurity research [6]. Numerous studies have reviewed and compared tools for attack graph generation and visualization [14] and summarized different attack modelling approaches, including attack graphs and attack trees [10], [15], [16]. Hong *et al.* analyzed graphical security models across four phases [17]. Recent work examined unknown vulnerability risk assessments using directed graph models [18], and empirical research analyzed over 180 attack

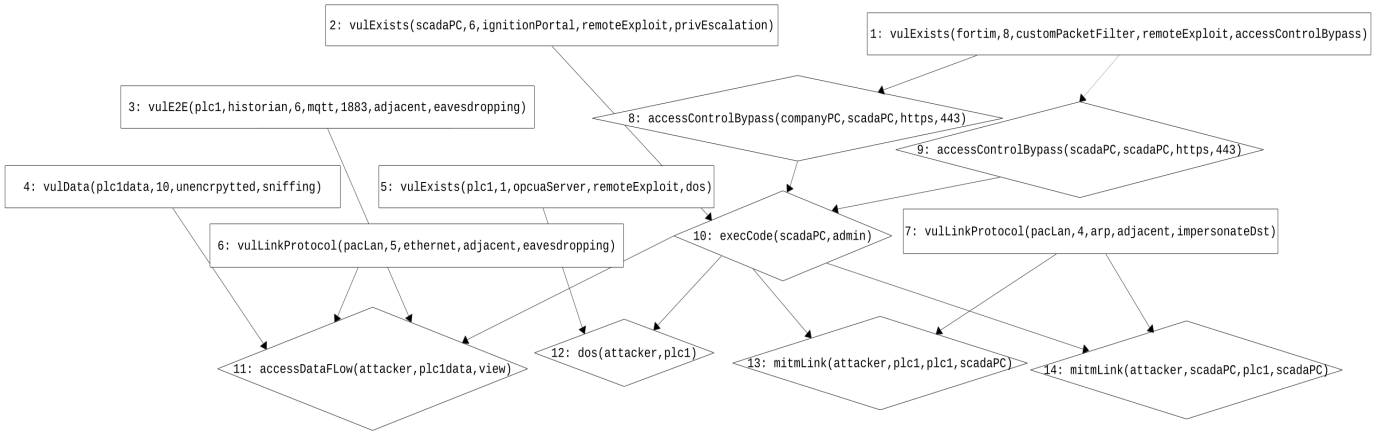

Fig. 4. Attack graph pruned from MulVal output

graphs and trees [19]. However, none of these studies detailed analyses of MulVAL attack graph generation extensions.

Several works mapped malware, CVEs, and CTI to the MITRE ATT&CK framework, a standard for cyber threat modelling [13]. Researchers mapped Windows malware families to ATT&CK and developed tools for structuring cyber threat reports [20], [21]. Machine learning techniques have been used to map CVEs to CAPEC and ATT&CK, identifying appropriate mitigations [22]–[25]. Kwon *et al.* aligned the MITRE ATT&CK Matrix with the NIST cybersecurity framework [26], and others developed a cyber-phrase embedding model for CTI texts [27]. Despite these efforts, no studies have mapped MulVAL interaction rules to MITRE ATT&CK Techniques.

These approaches are mostly confined to IT network scenarios and are not easily applicable to OT environments. Unlike standardized IT networks, Industrial IoT (IIoT) environments supporting OT often use proprietary systems tailored for specific industries [28]. Communication protocols also differ significantly, with OT networks using protocols like LTE and low-power wide-area communications [29]. Security practices vary, with IT emphasizing regular updates and OT often using outdated systems [3], [30].

Our research proposes an innovative solution based on MulVAL to address these limitations in OT networks. By leveraging MulVAL's rule-based inference engine, we generate detailed attack graphs that automate attack analysis, reduce errors, and provide comprehensive insights into network topology and related security issues.

VI. Conclusion and Future Works

This work represents an advancement in the capabilities of MulVAL, an established attack graphing tool, by introducing a novel tool focused on intelligent systems for effectively detecting cyber-attacks and several key enhancements. These include implementing a standardized JSON structured model for input, which facilitates a clear and consistent representation of network configurations and vulnerabilities. This structured approach not only improves the tool's usability but also enhances its compatibility across different environments and scenarios within OT networks. Moreover, the work focuses on refining OT scenario modelling by introducing extended interaction rules. These rules enable a more detailed and accurate representation of attack paths, accommodating specific nuances and complexities inherent in industrial control systems. By incorporating these rules, the tool can effectively simulate a broader range of attack tactics and techniques, aligning with the diverse threat landscape outlined in frameworks such as MITRE ATT&CK. While the current implementation serves as an initial prototype rather than a comprehensive solution, future iterations aim to broaden its functionality. Potential enhancements include expanding the repertoire of modelled attack scenarios by continuously integrating new interaction rules. Furthermore, there is a planned focus on enhancing the tool's practical utility by enabling the verification of identified vulnerabilities through automated testing of extracted attack paths. This capability validates the presence of vulnerabilities and provides actionable insights for security teams, facilitating targeted mitigation efforts and enhancing overall resilience. Additionally, the automation of network scanning and data-gathering processes is envisioned to streamline input preparation. By automating these tasks and adhering to the standardized input format, the tool can effectively reduce manual effort and operational overhead, optimising security assessments' efficiency in OT environments. Lastly, integrating metrics for risk and impact analysis represents a significant enhancement. These metrics will enable stakeholders to quantify and prioritize security risks, offering informed decision-making support for resource allocation and risk management strategies. By incorporating these enhancements, the tool aims to establish itself as a robust framework for automatic security assessment in OT environments.

References

[1] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, and S. Lightman, "Guide to operational technology (ot) security," *National Institute of Standards and Technology: Gaithersburg, MD, USA*, 2022.

[2] D. G. Pivoto, L. F. de Almeida, R. da Rosa Righi, J. J. Rodrigues, A. B. Lugli, and A. M. Alberti, "Cyber-physical systems architectures

for industrial internet of things applications in industry 4.0: A literature review," *Journal of manufacturing systems*, vol. 58, pp. 176–192, 2021.

[3] E. D. Knapp, *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Elsevier, 2024.

[4] M. S. Sonkor and B. García de Soto, "Operational technology on construction sites: A review from the cybersecurity perspective," *Journal of Construction Engineering and Management*, vol. 147, no. 12, p. 04021172, 2021.

[5] M. Tatam, B. Shanmugam, S. Azam, and K. Kannoorpatti, "A review of threat modelling approaches for apt-style attacks," *Heliyon*, vol. 7, no. 1, 2021.

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

[7] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing, "Automated generation and analysis of attack graphs," in *Proceedings 2002 IEEE Symposium on Security and Privacy*, 2002, pp. 273–284.

[8] X. Ou, S. Govindavajhala, A. W. Appel *et al.*, "Mulval: A logic-based network security analyzer." in *USENIX security symposium*, vol. 8. Baltimore, MD, 2005, pp. 113–128.

[9] L. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, vol. 2, 2001, pp. 307–321 vol.2.

[10] M. S. Barik, A. Sengupta, and C. Mazumdar, "Attack graph generation and analysis techniques," *Defence science journal*, vol. 66, no. 6, p. 559, 2016.

[11] D. Tayouri, N. Baum, A. Shabtai, and R. Puzis, "A survey of mulval extensions and their attack scenarios coverage," *IEEE Access*, vol. 11, pp. 27 974–27 991, 2023.

[12] 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 Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1936–1954, 2022.

[13] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," in *Technical report*. The MITRE Corporation, 2018.

[14] 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 *2013 International Conference on Anti-Counterfeiting, Security and Identification (ASID)*. IEEE, 2013, pp. 1–6.

[15] S. Haque, M. Keffeler, and T. Atkison, "An evolutionary approach of attack graphs and attack trees: A survey of attack modeling," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2017, pp. 224–229.

[16] U. Garg, G. Sikka, and L. K. Awasthi, "A systematic review of attack graph generation and analysis techniques," *Computer and Cyber Security*, pp. 115–146, 2018.

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

[18] W. He, H. Li, and J. Li, "Unknown vulnerability risk assessment based on directed graph models: a survey," *IEEE Access*, vol. 7, pp. 168 201–168 225, 2019.

[19] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Computer Science Review*, vol. 35, p. 100219, 2020.

[20] K. Oosthoek and C. Doerr, "Sok: Att&ck techniques and trends in windows malware," in *Security and Privacy in Communication Networks: 15th EAI International Conference, SecureComm 2019, Orlando, FL, USA, October 23-25, 2019, Proceedings, Part I 15*. Springer, 2019, pp. 406–425.

[21] V. S. M. Legoy, "Retrieving att&ck tactics and techniques in cyber threat reports," Master's thesis, University of Twente, 2019.

[22] E. Aghaei and E. Al-Shaer, "Threatzoom: neural network for automated vulnerability mitigation," in *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*, 2019, pp. 1–3.

[23] E. Aghaei, W. Shadid, and E. Al-Shaer, "Threatzoom: Cve2cwe using hierarchical neural network," *arXiv preprint arXiv:2009.11501*, 2020.

[24] P. Vishnu, P. Vinod, and S. Y. Yerima, "A deep learning approach for classifying vulnerability descriptions using self attention based neural network," *Journal of Network and Systems Management*, vol. 30, no. 1, p. 9, 2022.

[25] J. H. An, Z. Wang, and I. Joe, "A cnn-based automatic vulnerability detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, no. 1, p. 41, 2023.

[26] 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 *2020 Resilience Week (RWS)*. IEEE, 2020, pp. 106–112.

[27] M. D. Purba, B. Chu, and E. Al-Shaer, "From word embedding to cyber-phrase embedding: Comparison of processing cybersecurity texts," in *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2020, pp. 1–6.

[28] P. K. Garimella, "It-ot integration challenges in utilities," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. IEEE, 2018, pp. 199–204.

[29] R. Paes, D. C. Mazur, B. K. Venne, and J. Ostrzenski, "A guide to securing industrial control networks: Integrating it and ot systems," *IEEE Industry Applications Magazine*, vol. 26, no. 2, pp. 47–53, 2019.

[30] C. Alcaraz, "Secure interconnection of it-ot networks in industry 4.0," *Critical Infrastructure Security and Resilience: Theories, Methods, Tools and Technologies*, pp. 201–217, 2019.