Evaluating the Security Posture of 5G Networks by Combining State Auditing and Event Monitoring

Md Nazmul Hoq[1], Jia Wei Yao[1], Suryadipta Majumdar[1], Luis Suárez[2], Lingyu Wang[1], Amine Boukhtouta[2], Makan Pourzandi[2], and Mourad Debbabi[1]

[1] Concordia University, Montreal, Canada
{mdnazmul.hoq, jiawei.yao, suryadipta.majumdar, lingyu.wang, mourad.debbabi}@concordia.ca
[2] Ericsson, Montreal, Canada
{luis.suarez, amine.boukhtouta, makan.pourzandi}@ericsson.com

Abstract. 5G network technology is being rapidly adopted in various critical infrastructures, mainly due to its unique benefits (e.g., higher throughput, lower latency, and better scalability). This wide-spread and fast adoption necessitates securing those critical services deployed over 5G technology. However, evaluating the security posture of a 5G network is challenging due to the heterogeneous and large-scale nature of 5G networks coupled with new security threats. Moreover, existing 5G security approaches fall short as their results are typically binary and difficult to be translated into the overall security posture of a 5G network. In this paper, we propose a novel solution for evaluating the security posture of 5G networks by combining the results of existing security solutions for state auditing and event monitoring. To that end, our main idea is to first build a novel *event-state model* that captures both events and states in a 5G network, and then extend this model to evaluate the overall security posture and how such security posture may evolve over time due to persistent threats. We integrate this approach with free5GC (a popular 5G open-source project) and evaluate its effectiveness.

1 Introduction

Characterized by its higher throughput, lower latency, and better scalability, 5G technology has become a popular choice for telecommunication networks. By 2025, more than two-fifths of the global population will be under the coverage of 5G networks, and 5G connections will make up about a quarter of all mobile connections [1]. Therefore, the security of 5G networks becomes essential to its wide-range of users and applications. To that end, evaluating the security posture of a 5G network deployment can provide a direct measurement of the current security status as well as the potential impact of specific future plans (e.g., deploying a security appliance). However, evaluating the security posture of a 5G network renders unique challenges, such as understanding its heterogeneous components across multiple aspects and how those components interact, and

attacker's capabilities in exploiting those unique system dependencies to cause an evolving impact on the system.

Most existing works (e.g., [2,3]) for measuring 5G security fall short to overcome those challenges, as they are limited to a particular aspect (e.g., user, network, or infrastructure) of a 5G network and do not evaluate the system's overall security posture. On the other hand, the majority of existing works (e.g., [4–6]) for non-5G environments do not consider the 5G-specific threats that may exploit a 5G network across different aspects and cause sustained impact. Moreover, existing security tools (e.g., Kubescape [7] and Falco [8]) for 5G networks are mostly designed to find a specific security breach or incident, respectively, and it would be highly challenging for security admins to interpret and correlate the results of those different tools into an overall security posture of a 5G network. We further highlight these limitations and motivate towards our work using the following example.

Motivating Example. The left side of Figure 1 shows an attack scenario (based on a well-known advanced persistent threat (APT) to telecommunications networks, LightBasin [9]) to illustrate an attacker's capability to exploit a 5G network from multiple aspects (e.g., user, network, and infrastructure) by following *Steps 1-6* and cause an evolving impact (from user aspect to infrastructure aspect). The right side of the figure first shows the limitations of existing works towards evaluating the security posture of a 5G network during the APT attack, and then illustrates our idea to overcome those limitations, as detailed in the following.

- While working with existing security solutions (e.g., auditing and monitoring tools [7,8]) and relevant security controls (e.g., [10–12]), a 5G admin observes both compliant (T) and non-compliant (F) states of a system as the attack progresses over time (t_1-t_4). However, the admin faces challenges about how to interpret those binary results from auditing individual security controls, and how to aggregate them towards evaluating the overall security posture of the 5G network.
- To address this limitation, our key idea is to combine the results of existing security tools (e.g., auditing and monitoring) to build a probabilistic model (Bayesian network) that correlates the different aspects of a 5G network and captures the evolving nature of an APT attack over time. As a result, an admin can obtain security posture values for a given goal node (G_1 and G_2) from observed breaches or events (E_1 and E_2) to have a better understanding about the security status of the system.

More specifically, we propose a 5G Security Posture Evaluator (namely, *5GSPE*) based on the results of state auditing and event monitoring. First, we build a *state model* from auditing results that capture breaches of organization-specific security controls based on 5G network states at different times, and an *event model* from monitoring results that capture the attacker's activities performed in-between breaches. Second, we combine these two models to build an *event-state model* as a Bayesian network that captures the evolving nature of 5G attacks. Finally, we leverage this probabilistic *event-state model* to evaluate the

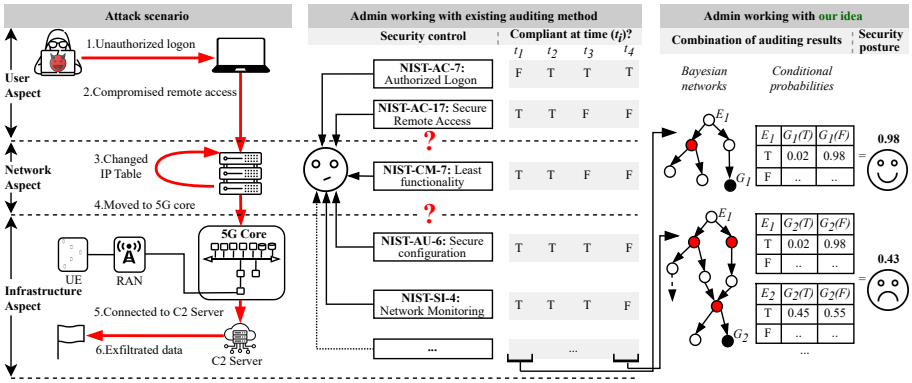


Fig. 1: A motivating example based on the LightBasin attack

security posture of a 5G network through Bayesian inference of the model. We implement *5GSPE* based on free5GC [13] (a popular project for deploying 5G core) with Kubernetes [14] (a major container orchestrator), and demonstrate its effectiveness through extensive experiments. The main contributions of this paper are as follows.

- As per our knowledge, we are the first to propose a solution for evaluating the security posture of 5G networks that can cover persistent threats involving multiple aspects in the 5G technology stack and threat evolution over time.
- To achieve this, we study security controls from multiple system aspects; we propose a novel *event-state model* by combining results of state auditing and event monitoring; we design a custom algorithm to combine those results and quantify the overall security posture of a 5G network.
- We integrate our solution with free5GC [15] and Kubernetes [14] and conduct experiments to show its effectiveness in reflecting the effects of attack progress, network scalability, and security appliances, among others on the overall security posture.

The rest of the paper is organized as follows. Section 2 provides preliminaries. Section 3 describes our methodology. Section 4 details its implementation. Section 5 presents experimental results. Section 6 reviews the literature. Section 7 concludes the work.

2 Preliminaries

This section provides the necessary preliminaries.

2.1 Background on 5G network

Figure 2 shows an overview of 5G network that contains components from three major aspects: A user aspect is concerned with the administration of the 5G core network including UE, which can be both mobile and IoT devices. A network

aspect covers network equipment, such as RAN and 5G mobile core (with various virtual network functions including but not limited to AMF, UPF, SMF, and UDM) [16]. An infrastructure aspect includes the virtual resources (e.g., vSwitches, vRouters, vServers) typically hosted on cloud infrastructure. We also summarize the acronyms used throughout this paper in Table 1.

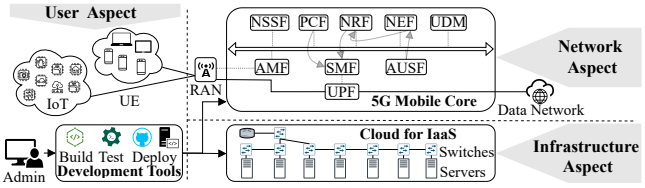


Fig. 2: An overview of the 5G network

Table 1: Acronyms used in this paper

Term	Description	Term	Description	Term	Description
3GPP	3rd Generation Partnership Project	5GC	5G Core	AMF	5G Access and Mobility Management Function
AUSF	5G Authentication Server Function	BN	Bayesian Network	CN	Core Network
EM	Event Model	ESM	Event-State Model	LTE	Long-Term Evolution
NEF	5G Network Exposure Function	NF	Network Functions	NRF	5G Network Repository Function
NSSF	5G Network Slice Selection Function	PB	Policy Breach	PCF	5G Policy Control Function
RAN	Radio Access Network	SM	State Model	SMF	5G Session Management Function
UDM	5G Unified Data Management	UE	User Equipments	UPF	5G User Plane Function

2.2 Security Posture Evaluation in 5G and Its Challenges

In the following, we define security posture in the context of 5G and outline the challenges involved in its evaluation.

Definition of Security Posture in 5G. Based on 3GPP [17], a major project to develop 5G standards, the security posture of a 5G network is defined from two main factors: (i) by checking how well security practices and guidelines are followed in a 5G deployment and its operation (as captured in our state model); and (ii) by monitoring network elements, infrastructure, and communication (as captured in our event model).

Challenges. The main challenges of evaluating security posture in 5G are:

- **Complexity of the composition of a 5G network.** Unlike pre-5G networks (e.g., LTE and 4G), which are typically deployed on an operator-managed infrastructure, 5G network components (i.e., CN, RAN, UE, etc.) are distributed across multiple aspects (user, network, and infrastructure [18]) which can be deployed and instantiated across third party clouds and systems. The interactions between network functions, virtualized environment and infrastructure owners further complicate the nature of a 5G network. Due to such added complexity, evaluating the security posture of a 5G network requires a thorough understanding of this system.
- **Capturing attack progress with existing security solutions.** Attacker capabilities (skills, tools, and resources) for exploiting a 5G network vary based on the attacker's motivation, sophistication, and objectives [19]. While different attackers can advance differently across various aspects of 5G depending on their capabilities, the current security solutions are insufficient

to understand the relationship between various breaches and to capture the attacker's progress for assessing the system's overall security posture.
- **Interpreting and aggregating the results of existing security solutions.** There are a few 5G security solutions for auditing (e.g., [20]), monitoring (e.g., [7]), and alert reporting (e.g., [8]). Different (sometimes inconsistent) security controls, formats of results, and operational mechanisms, across those solutions, may impede correlation among them to evaluate the system's overall security posture. It is also challenging to interpret the binary results to evaluate the security posture of a 5G network.

These challenges will be addressed in Section 3.

2.3 Threat Model

The *in-scope threats* of this work include the attacks whose impact on the security controls of a 5G network can be identified using existing auditing/monitoring techniques (e.g., [7,8]). We assume the 5G network and infrastructure management systems (e.g., Kubernetes) may be trusted for the integrity of the API calls, event notifications, logs, and database records. The system may have implementation flaws, misconfigurations, and vulnerabilities that can be potentially exploited by malicious attackers to violate security controls. We assume that admins are interested in the security posture with respect to given attack goals (i.e., targeting critical assets). For identifying privilege escalation nodes, we rely on expert's intervention with the help of the MITRE FiGHT [21], a 5G-specific knowledgebase of attacker strategies, which extends the generic MITRE ATT&CK [22].

The *out-of-scope threats* include the attacks that do not violate the specified security controls, that are not captured by existing security solutions, and that may remove or tamper their own logged events. We also assume loose synchronization of events across different aspects such that their order are correctly reflected in Kubernetes logs and Systems logs. Even though this work can evaluate the effects of attacks on a system, its main objective is not to detect any specific attack or exploit, but to evaluate security postures for admins.

3 Methodology

This section presents our methodology of *5GSPE*.

3.1 Overview

Figure 3 shows an overview of our methodology, which contains three major steps. First, we build a state model from auditing results of 5G network states to capture the compliance breaches by an attacker and an event model from monitoring results of 5G network events to capture an attacker's activities (as detailed in Section 3.2). Second, we fuse these two models and build an *event-state* model (which is formally defined in Appendix A) using both *horizontal*

fusing, which correlates the results of auditing and monitoring, and *vertical fusing*, which links different aspects (e.g., user and infrastructure) of a 5G network (as detailed in Section 3.3). Finally, we evaluate the security posture of a 5G network by converting our event-state model to a probabilistic model as a Bayesian network (as detailed in Section 3.4).

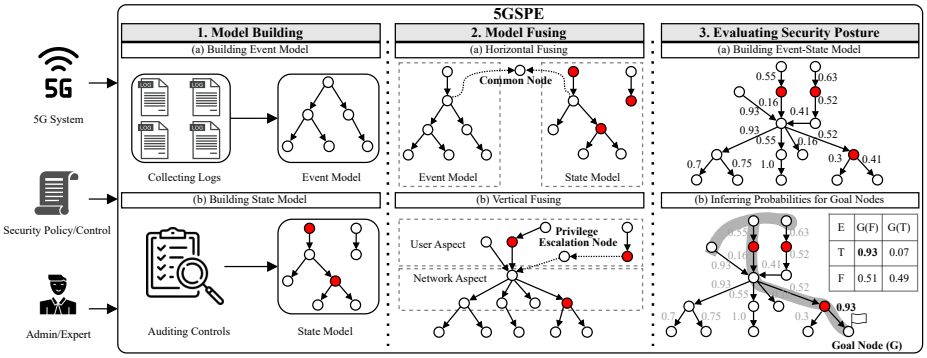


Fig. 3: Overview of our methodology

3.2 Building Event and State Models

To capture an attacker's activities and their impacts, we build the *state model* using auditing results and the *event model* using monitoring results, as follows.

Building State Model. A major angle of the attacker's activities would involve the execution of malicious operations that could result in non-compliant system states. Therefore, to capture this aspect of the attacker's activities, we build the state model. First, we obtain auditing results (e.g., compliant/non-compliant) over a period of time by applying existing security auditing tools (e.g., KubeScape [7]) and security controls (e.g., NIST [10]) covering different aspects of a 5G network. Then, we measure the frequency of both compliant and non-compliant results for each security control to calculate the probability of a non-compliant result. Next, we determine the pre-condition (cause of the breach) and the post-condition (impact of the breach) for each non-compliant result using the description of the control, and MITRE FiGHT [21] and ATT&CK framework [22]. Afterward, for each non-compliant control, we add its pre-condition and post-condition as parent and child, respectively, allowing us to correlate attackers non-compliant state related activities with 5G related activities. Finally, we combine all non-compliant controls, and their pre-conditions and post-conditions to build a *state model* using model fusing (described in Section 3.3).

Example 1. We utilize the same attack scenario as in our motivating example to illustrate how attackers compromise system states. Figure 4 explained how to build the state model. Initially, an attacker performs an unauthorized logon that violates the `AC-7: Authorized LogOn` control from NIST in the user aspect and enables the attacker to escalate to the network aspect. Afterwards, s/he violates two additional security controls (i.e., `SI-4` and `CM-7`) through further malicious

activities. To capture the effects of these malicious activities, a state model is built as follows. (i) We collect auditing results from three tools, KubeScape [7], Falco [8], and a custom tool based on formal verification [23]. (ii) We count the number of compliance (3) and non-compliance (2) for `AC-7` and calculate the probability of non-compliance (0.40). (iii) We check `AC-7` description ("*consecutive invalid logon attempts should be limited*") and its corresponding MITRE tactics `Credential Access (MITRE TA0006)` and with the help of an expert, we find MITRE sub-technique `Password Spraying` as a pre-condition (cause of the breach), and `Bypass User Account Control` as a post-condition (a potential result of the breach). Afterwards, we connect the pre-condition as a parent and the post-condition as a child of the breach node of `AC-7` control. Following these same steps for other controls, we build the entire state model.

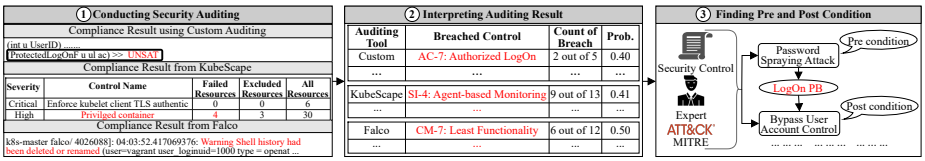


Fig. 4: Example of building a state model

Building Event Model. Another angle of the attacker's activities would involve performing activities supported by 5G network. Therefore, to capture such activities, we build the event model. First, as a source of events, we collect the monitoring results (e.g., event logs) from multiple sources of a 5G network to cover its different aspects: system logs for users, auditing logs for infrastructure, and 5G core logs for networking. Then, to extract the event-related information (e.g., timestamps and details of relevant events [24]), we filter out any extraneous logs and trim the log data. Subsequently, we combine all the logs from different sources based on their timestamp, resulting in a single processed log. We also identify similar events (e.g., performing the same operations) that are named differently in different logs, and uniquely categorize them as a single event type in our model. Then, to capture attacker activities (represented as a BN), we generate event sequences where each sequence ends as soon as any event is repeated to avoid cycles in our model (mainly because a BN is acyclic). Finally, we construct our *event model* (as further illustrated using the following example).

Example 2. After gaining unauthorized access as an admin, an attacker may: `access to server` at the Kubernetes [14] level to alter the IP table to avoid detection, and `query to SMF` at the 5G level to get subscriber information from UDM. Figure 5 shows the steps to build the event model. (1) We collect and parse raw logs from Kubernetes and free5GC. (2) We eliminate the entries with the message `set report call: false`, and the `Log Level`, `Component`, and `Module` columns, as they do not provide event-related information. (3) We combine both logs, and sort them based on the *Timestamp*. (4) We rename both `Created container UPF` and `Started container UPF` messages as `connect to UPF` event, and identify the repetition of `Query to SMF` events for sequence

generation without a cycle. (5) We obtain three event sequences. (6) We construct an event model, where nodes `access server` and `Query to SMF` and their transition probability (0.7) are learned from the sequences. Thus, the event model shows the activities of the attacker.

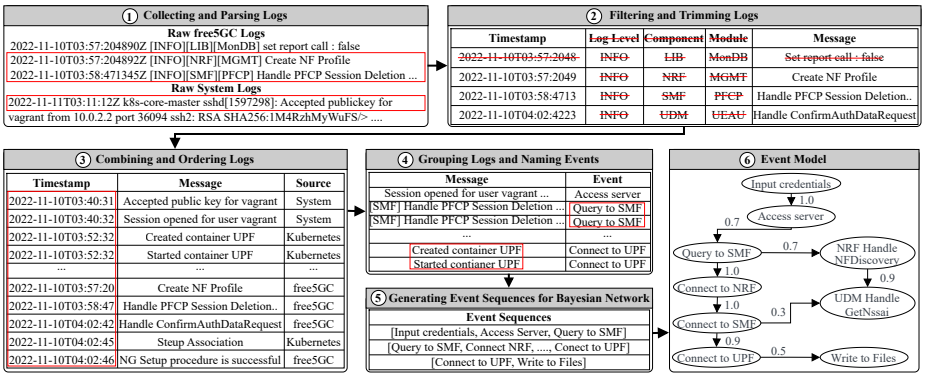


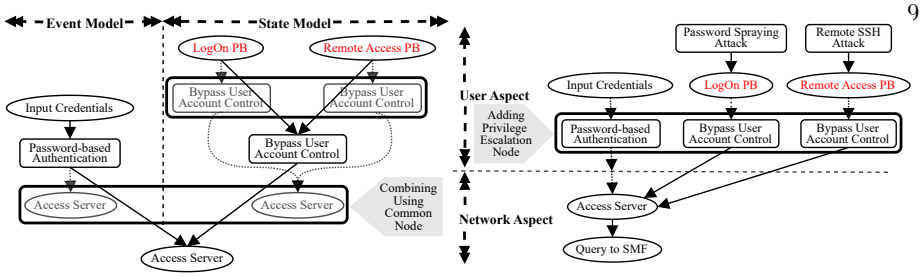
Fig. 5: Example of building an event model

3.3 Model Fusing

To capture the overall impacts of the attacker's activities on the system, both models are fused in this step by performing two major steps as follows.

Horizontal Fusing. This step is to show the combined effects of the attacker's activities causing non-compliance system states from the state model (SM) and in-between breach activities supported by the 5G network from the event model (EM). First, we identify all the common nodes between *EM* and *SM* caused by the impact of a privilege escalation resulting from a non-compliant system state. Second, for each common node, we record and merge their parents and children from both models with their respective transition probabilities. This merged list represents the steps an attacker might perform, whether by mimicking legitimate users or causing a breach, and it captures the transition between these two. Third, we create a new model by adding each common node to their parents and children. Finally, the remaining nodes (which have not been added yet) from both EM and SM are added to this new model.

Example 3. In Figure 6a, we fuse the event model (on left) and state model (on right) using their common node, `Access Server`; which combines two different activities, `Password-based Authentication` representing attacker mimicking a legitimate user, and `Bypass User Account Control` representing a post-condition of a breach. Prior to this step, we also use the same procedure within the state model in combining two breach-related activities `LogOn Policy Breach (PB)` and `Remote Access PB`, leading to a common privilege escalation `Bypass User Account Control`. We use dashed lines to indicate the model's instance before fusing and solid lines to indicate its instance after fusing.



(a) Example of horizontal fusing (b) Example of vertical fusing

Fig. 6: Examples of fusing event model with state model into event-state model

Vertical Fusing. This step is to show the impacts of a breach among different aspects (e.g., user, network) of a 5G network. First, we identify the privilege escalation (post-condition) that occurs as a result of a breach (explained in Section 3.2). Then, using the MITRE FiGHT [21] and ATT&CK [22] framework, we identify the impact of this privilege escalation. This impact could be a legitimate event from the EM or another breach of the system state from the SM. Moreover, an expert verifies the decision using his/her understanding of attack and mitigation. Finally, we connect the impacted node with the privilege escalation node to show the attacker's progress in different aspects.

Example 4. As shown in Figure 6b, vertical fusing connects the user aspect and network aspect of the system where an attacker uses two breaches `LogOn PB` and `Remote Access PB` to gain the same privilege of `Bypass User Account Control`. Using expert assistance and MITRE framework, we determine that `Bypass User Account Control` can be used for both `Credential Access (MITRE TA0006)` and `Lateral Movement (MITRE TA0008)` which impact is the `Access Server` in EM. Therefore, we connect `Bypass User Account Control` from the user aspect of the state model (on top), to `Access Server` from the network aspect of the event model (on bottom) by adding an edge between these two nodes. This shows the attack progress from the user aspect to the network aspect and allows attackers to execute the `Query to SMF` operation.

3.4 Evaluating Security Posture

Evaluating security posture is performed using the following two steps.

Building Event-State Model. First, we use the fused model from the previous step to build the structure of Bayesian Network (BN). Then we use the historical log data from both monitoring events and auditing states of a 5G network to learn its parameters. Each incoming edge to a node in the resultant BN model indicates the probability of an activity and each node is represented as follows: (i) white nodes represent normal events, (ii) red nodes represent a breach state, and (iii) rectangular nodes represent privilege escalation, as depicted in Figure 7.

Inferring Probabilities for Goal Nodes. This step makes inferences from the resultant event-state model based on the observed conditions (value of the nodes: occurred or not) of the system and the user-defined goal node. A *goal node* is an event or state of a system that are chosen by the admins based on

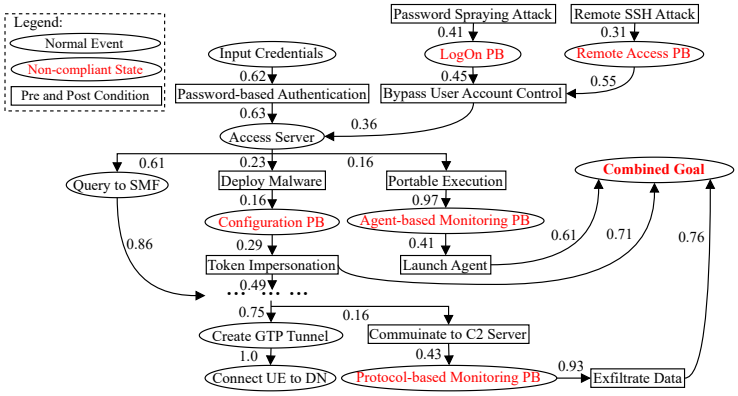


Fig. 7: Evaluating security posture based on the event-state model

the security requirements of the organization. Given the goal node and observed conditions, we use Bayesian inference to calculate the conditional probability of the goal node. This probability measures an attacker's ability to reach the goal node, so from an administrator's perspective, security posture is the value that complements it. Choosing multiple goal nodes is also possible in the real world. To that end, we consider a dummy node that can be linked to all those goal nodes. Then, we re-build the model so that it adds the dummy node and re-calculates the conditional probability for each node based on this change. Using the same approach, we can infer the value for this combined node which represents the combined security posture for all those goal nodes.

Example 5. Figure 7 shows an excerpt of our event-state model where we evaluate the security posture for the user-defined goal: `Exfiltrate Data`. Admin set the value of `Access Server`, `Deploy Malware`, and `Configuration PB` as occurred as observed conditions. Then Bayesian inference method on the event-state model yields a conditional probability of 0.07 and taking the complement, we get the security posture value of 0.93 for `Exfiltrate Data`. Moreover, to facilitate multiple goal nodes, a new node is created and labeled as `Combined Goal` which connects three different goal nodes (`Token Impersonation`, `Launch Agent`, and `Exfiltrate Data`). After rebuilding the model using the same observed conditions, the new security posture value is 0.76 because there are now more paths to the combined goal, thereby decreasing the security posture value.

4 Implementation

This section discusses the implementation and integration details of 5GSPE.

4.1 5G Testbed Implementation

For our 5G testbed, we use Towards5GS-helm [15] to automate the deployment of Free5GC [13] (version 3.2.1) on top of Kubernetes [14] (version 1.22). The simulation of RAN is ensured in conjunction with UERANSIM[3] for testing 5G

[3] https://github.com/aligungr/UERANSIM

core functionalities. We also use existing auditing tools, KubeScape [7], Falco [8], and a custom tool based on Sugar [23], to monitor and audit the security of a 5G network. We use KubeScape (v2.0.183) [7], to scan for potential vulnerabilities and misconfigurations in a Kubernetes cluster. However, since KubeScape does not focus on runtime detection, we use Falco (v0.33.1) [8] for detecting potential run-time security breaches in Kubernetes. As, Falco does not audit any activities in 5G network functions, we use custom detection rules, and a first-order logic-based tool called Sugar [23] to detect breaches in 5G network functions. Table 2 shows a few example rules for those tools. **Challenges:** We face a memory limit issue in Kubescape, and reconfigure the `kube-apiserver.yaml` file and set `--audit-webhook-mode` to `blocking` so that the Kubernetes API server does not send a response for each event. We also use `journactl` to dynamically retrieve all Falco alerts to address the issue of varying time precision among tools.

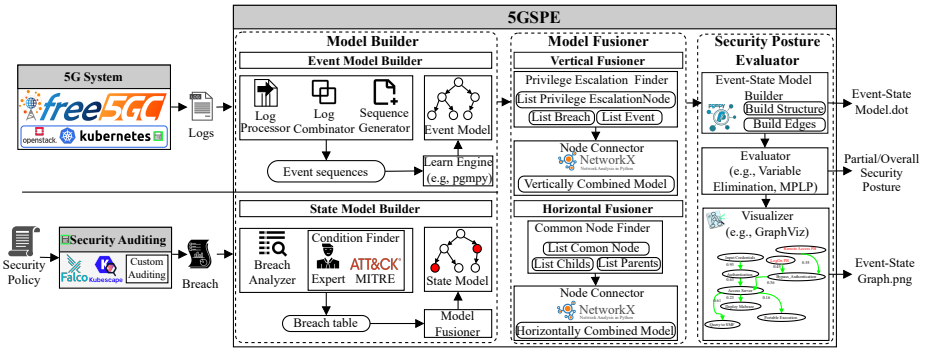


Fig. 8: System architecture of 5GSPE

4.2 5GSPE Implementation and Integration to free5GC

Figure 8 depicts the system architecture of 5GSPE. The *model builder* module is responsible for building both the event and state model. 5GSPE first automatically collects logs from different components, e.g., Kubernetes, Falco, KubeScape, etc., and pre-processes them to identify event sequences from the logs. These event sequences are converted to the input (in .text) to `PGMPY` [25] (v0.1.19), a Python library used for building models and learning with a Bayesian network. Next, to build the state model, we first deploy a breach analyzer to analyze the security breach and construct a breach table. The model fuser module is implemented based on a customized algorithm (Algorithm 1 in Appendix B). We use networkX [26] to implement various graph operations, such as add_node, find_cycle. Lastly, the security posture evaluator module uses different classes (e.g., Max-Product Linear Programming (MPLP) and variable elimination) from `PGMPY` to evaluate the security posture. We also use graphviz [27] to visualize the models. **Challenges:** Identifying relevant security controls from thousands of existing controls is a laborious and time-consuming process. To summarize and extract keywords from the controls, we use NLP tools (e.g., BERT [28], NER [29]). We also use them to determine the pre- and post-conditions of a security control breach from the MITRE framework. Graphviz's default layouts

limit the user's ability to analyze, and arrange nodes and edges aesthetically; therefore, we employ pydotplus [30] coupled with Graphviz to solve this issue.

Table 2: Example of detection rules for security tools

Security Tool	Detection Rules	Auditing Source
Falco	Detect any attempt to attach/exec into a Pod	Kubernetes audit log
Falco	Detect any inbound connection from a source outside of an allowed set of IPs, networks, or domain names	Systemcalls
KubeScape	Check for delete or deletecollection RBAC permissions on workloads	RoleBindings
Custom scripts	Detect any unexpected HTTP requests to UDM	free5GC UDM logs

5 Experiments

This section describes the experimental setup and results.

5.1 Experimental Setup

Our testbed is deployed on an OpenStack [31] environment with one master node, and two worker nodes on a server running with Ubuntu 20.04. In addition to our testbed data, we generate synthetic data in a simulated environment that creates 25,000 event sequences including 25 unique event types including breaches of specific security controls from NIST [10] (`AC-7: Logon Policy Breach (PB)`, `AC-17: Remote access PB`, `AC-8: System use notification PB`, `CM-7: Configuration PB`, `SI-4 (2): Agent-based monitoring PB`, `SI-4 (7): Rule-based monitoring PB`). To generate event sequences synthetically, we follow the dependency among event types captured from our testbed and assign different event types to attackers in a random but realistic manner. We first define the *attacker capability*, which is the number of events (both regular and attack events) in a sequence that an attacker is capable of completing. Then, we generate attackers with different capabilities using an exponential distribution (where the majority of attackers possess an initial capability and comparatively fewer attackers with higher capabilities) and assign generated events accordingly. We also divide each event sequence into three attack stages based on the MITRE ATT&CK framework, with each stage showing the attacker's attack progression (in Table 3). We conduct each experiment 1,000 times and calculate the average value.

Table 3: Statistics of our dataset

Attack Stage	MITRE Tactics	No. of Seq.
1	Reconnaissance to Initial Access stages	9,290
2	Execution to Credential Access stages	5,938
3	Discovery to Exfiltration stages	10,556
	Total	25,784

Attacker Level	Sequence Length	No. of Seq.
1	1 to 2	17,242
2	3 to 4	5,518
3	5 to 6	1,737
4	more than 6	1,287

5.2 Experimental Results

In the following, we present our experimental results.

Effects of Attack Progress. The purpose of our first set of experiments is to accurately determine the overall security posture as the attack progresses. Experiment parameters include the different attack stages and goal nodes. To better interpret the security posture values derived from our solution, we also measure the attacker success ratio (ASR), which is the proportion of successful attackers

reaching to the goal node during an attack stage. Figures 9a, 9b, 9c, and 9d show the relationship between the attack stages and the overall security posture value for different goal nodes: `combined goal` (considering all the following goal nodes), `exfiltrate data`, `launch agent`, and `token impersonation`, respectively. We observe that the security posture values and the attacker success ratio are roughly inversely proportional. This is expected because a higher security posture value indicates a better-secured system, and vice versa. On the other hand, as an attack advances from one stage to the next, the value of the whole security posture declines as shown in Figure 9. Moreover, the average security posture value reduces more when the goal node is closer to the attacker (e.g., for token impersonation as the goal node, the average posture value decreases by 29% (0.93 to 0.66 in Figure 9d) whereas, for exfiltrate data as target node, it decreases by just 1% (0.99 to 0.98 in Figure 9b) from attack Stage 1 to attack Stage 2. Therefore, we can conclude that, in addition to the goal node, the attack stage also plays a critical role in affecting the overall security posture.

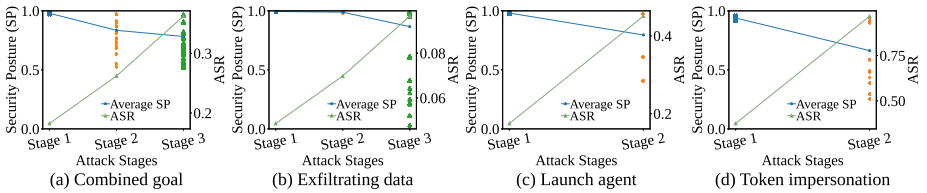


Fig. 9: Effects of attack progress for different goal; ASR: Attacker Success Ratio

Effects of Scaling Network Size (of 5G Core network). This second set of experiments examines how network size affects security posture. We generate different 5G core networks by varying the number of NFs (e.g., UPF and AMF) between 1 and 20 [32]. Then, for each network, we generate random event sequences (as described in Section 5.1). Finally, we calculate the average security posture value, model size (total number of nodes and edges), time to build the model and time to evaluate it for different network sizes. Figure 10a depicts the linear increase in model size due to network scaling which remains reasonably small compared to the size of other related graphs (e.g., attack graph [33], which can exceed 10,000 nodes). Here, `NF1` represents a user plane function, and `NF2` represents a control plane function. Figure 10b shows that both model building time and prediction time grow linearly with the scaling of the network. However, up to 15 NFs, the time needed to build the model is less than two seconds, showing that our model is efficient up to a certain level of network scaling. We also notice a significant increase in the build time for 20 NFs, which may be brought on by the interconnection of several NEFs with other NFs, leading to an increase in the number of edges. Note that if the network size remains constant, we do not need to rebuild the model with each increase in data, and updating the model with new data takes very little time. Figure 10c shows that the security posture value does not change much with network size because network size does not necessarily increase security control breaches. The red line in the figure illustrates a magnified version of `NF2_EXfiltrate` using the right y-axis with a

different scale. It is evident that the difference in security posture is minuscule (i.e., 0.02 while increasing the number of NFs from 1 to 20).

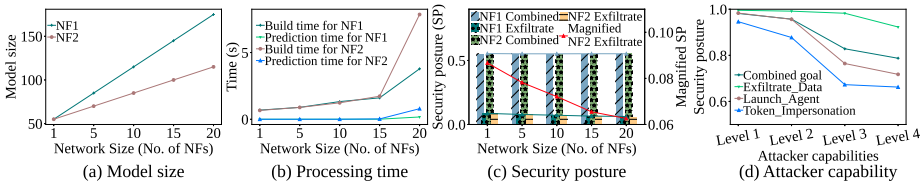


Fig. 10: Effects of scaling network size and Effects of attacker capabilities

Effects of Attacker Capabilities. The third set of experiments illustrates how the capabilities of an attacker affect the values of the overall security posture. Figure 10d demonstrates that, as the attacker's capability grows, the security posture value declines. The average security posture value for an attacker with capability *Level 1* (the lowest, defined in Table 3) is 0.98, whereas, for an attacker with capability *Level 4* (the highest), it is 0.78 for the combined goal node (a dummy node, as described in Section 3.4). Additionally, as most attackers with initial capability level (e.g., 1) can reach the goal node closer to the attacker (e.g., `Token Impersonation`) more easily, the security posture value is lower for that goal node than for the far-end goal node (e.g., *Exfiltrate data*). However, since the combined goal node additionally considers all other goal nodes, its security posture value falls between those of the other goal nodes.

Effects of Reducing Policy Breach by Implementing Security Solutions. The final set of experiments investigates how our solution can find the overall effect on security posture after implementing security solutions in a system. To do so, first, we calculate the security posture value using the attack example in Figure 1, which has six policy breaches, without reducing its breaches. Then, we reduce individual security policy breaches (as a potential impact of implemented security solutions) and recompute the security posture value. Figure 11a indicates that as the number of security policy breaches is reduced, the size of the model reduces, with the greatest reduction occurring when there are more than two policy breaches. Both build time and prediction time also decrease, but we do not observe a significant change as the model is not large enough and, therefore, the compute time is not much affected (in Figure 11b). Figure 11c illustrates that the security posture value is minimum when no policy breach is addressed (without reduction), and it increases as security policy breaches are solved. The red line illustrates a magnified version of `EXfiltrate Data` using the right y-axis with a different scale which shows an increase in security posture value. Figure 11d corroborates this trend with data from two distinct attacks deployed on our testbed: LightBasin (blue line) and a custom APT attack (orange line), both of which breach three security policies described in Table 4. In both attacks, with all three policy breaches, the security posture is minimal, and it increases as the number of policy breaches reduces. The average security posture value increases for the custom APT attack from 0.62 with all three breaches (EIP, TSCF, and ACR) to 0.99 with no breaches.

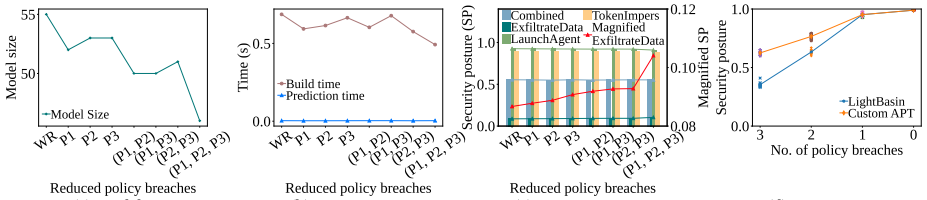


Fig. 11: Effects of reducing security policy breaches where WR: Without Reduction, P1: Reducing System Use Notification Policy Breach (PB), P2: Reducing Remote Access PB, P3: Reducing LogOn PB

Table 4: Implemented attacks description

Attack	Attack Steps	Purpose	Breached Security Policy
Light Basin	Unexpected Inbound Connections (UIC)	To gain access to the system	AC-3 (Access Enforcement)
	Creating NodePort Services (CNS)	To expose containerized 5G Network Functions(NFs) to the Internet	CM-6 (Configuration Settings)
	Unexpected HTTP Requests to UDM (UHRU)	To exfiltrate subscribers' data from the UDM	SC-7 (Boundary Protection)
Custom APT	Exec Into Pods (EIP)	To execute malicious codes in pods	AC-3 (Access Enforcement)
	Tampering Shell Configuration Files (TSCF)	To weaken the system security	SI-7 (Software, Firmware, and Information Integrity)
	Attaching Cluster-admin Role (ACR)	To impersonate network admin.	AC-6 (Least Privilege)

6 Related Works

This section reviews the related literature.

5G Security and its Requirements. Existing works (e.g., [2,34]) in 5G use an attack graph-based approach to propose a 5G-specific metric using cross-layer information shared among 5G network stakeholders and focusing more on its user aspect. Whereas, other works (e.g., [35]) measure the security of commercial 5G deployment while considering the user aspects (e.g., subscriber protection, user plane protection). However, all of them are limited to a particular aspect (e.g., user, network, or infrastructure) of a 5G network. There are a few works (e.g., [36]) that quantify other aspects of 5G, but they do not evaluate its security posture. Ericsson researchers recommend continuous monitoring, logging, least privilege principles, data encryption, threat detection and response, etc. to evaluate security posture and identify areas for improvement [37]. 5G network vulnerabilities include hardware, firmware, and software weaknesses, as well as issues with signaling and control plane protocols, containers, and Kubernetes. This is a call to a comprehensive, multi-layered approach required to assess network security and address all service aspects and internal components [38].

Security Posture of Non-5G network. Several works (e.g., [4, 5]) involve attack-graph analysis, where the likelihood and potential impact of a specific threat is evaluated using probabilistic models. Specifically, Frigault et al. [4] incorporate temporal factors using dynamic Bayesian networks. Whereas Wang et al. [5] focus on causal relationships between vulnerabilities encoded in attack graphs. Other works (e.g., [39, 40]) evaluate security metrics either from the vulnerabilities or from the behavior of network devices. However, none of them address 5G-specific challenges to capture the persistent threats with evolving impacts on multiple aspects of a 5G network. Additionally, there are several risk

assessment frameworks, such as the NIST Cybersecurity Framework [10] and the ISO 27001 [24], for organizations to assess risks. However, they usually provide a more generic assessment, whereas our objective is to evaluate the overall security posture for a particular 5G network deployment.

Standalone Security Solutions. There are a few built-in tools (e.g., Kubescape [7] and Falco [8]) that support auditing and monitoring for 5G environments on Kubernetes. Moreover, different auditing techniques are for specific applications, such as auditing in the cloud [41] and IoT [42]. Other solutions in 5G (e.g., [43–47]) focus on various security objectives (e.g., attack and interference detection, log management, etc.) However, none of those works focus on security posture evaluation. Also, the European Telecommunications Standards Institute (ETSI) proposes a security framework [48] to manage virtualized networks securely, while 3GPP proposes a data analytic architecture for 5G networks [49, 50]. These efforts demonstrate the significance of considering security as a crucial factor for the stability and availability of services deployed using 5G networks. However, none provides an overall security posture of a 5G network.

In summary, 5GSPE differentiates from other works by addressing 5G-specific challenges (e.g., covering its multiple aspects, capturing evolving impact of persistent threats) in evaluating the overall security posture of a 5G network.

7 Conclusion

In this paper, we proposed 5GSPE, a new approach for evaluating the security posture of 5G networks while considering the 5G-specific security threats (e.g., persistent threats with evolving impacts from multiple aspects of a 5G network). Specifically, we first built an event model using monitoring results and a state model using auditing results. We then combined these two models as a Bayesian network by leveraging their common nodes along with privilege escalation nodes. Finally, we evaluated the security posture of a 5G network through Bayesian inference of the model. We implemented and integrated our solution for free5GC and Kubernetes, and demonstrate its effectiveness through extensive experiments.

There are a few limitations of this work. Our work presently focuses on measuring the current security posture of the system and does not predict future security posture. In the future, we intend to extend our work to support predictive capabilities based on historical data analysis. We currently rely on the knowledge of security experts to identify the pre- and post-conditions of a security breach. Our future work will focus on reducing this reliance. Also, we plan to apply our work to other systems (e.g., clouds, IoT).

Acknowledgments. The authors thank the anonymous reviewers for their valuable comments. This work is mainly supported by Ericsson Canada and the first author was partially supported by the Natural Sciences and Engineering Research Council of Canada under the Discovery Grants RGPIN-2021-04106.

References

1. GSMA, "The Mobile Economy 2022." https://www.gsma.com/mobileeconomy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf, Dec. 2022. (accessed: 24 May 2023).
2. L. Zhao, M. S. Oshman, M. Zhang, F. F. Moghaddam, S. Chander, and M. Pourzandi, "Towards 5G-ready security metrics," in *ICC 2021-IEEE International Conference on Communications*, pp. 1–6, IEEE, 2021.
3. E. Yocam, A. Gawanmeh, A. Alomari, and W. Mansoor, "5G mobile networks: reviewing security control correctness for mischievous activity," *SN Applied Sciences*, vol. 4, no. 11, pp. 1–17, 2022.
4. M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic Bayesian network," in *Proceedings of the 4th ACM workshop on Quality of protection*, pp. 23–30, 2008.
5. L. Wang, A. Singhal, and S. Jajodia, "Measuring the overall security of network configurations using attack graphs," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 98–112, Springer, 2007.
6. M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 1071–1086, 2016.
7. "Github:Kubescape." https://github.com/kubescape/kubescape, Dec. 2022.
8. Falco, "Falco." https://github.com/falcosecurity/falco, Dec. 2022.
9. "LightBasin: A roaming threat to telecommunications companies." https://www.crowdstrike.com/blog/an-analysis-of-lightbasin-telecommunications-attacks/, Dec. 2022.
10. NIST, "SP 800-53 Rev. 5, Security and Privacy Controls for Information Systems and Organizations." https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final, Dec. 2022. (accessed: 14 May 2023).
11. CSA, "CSA Cloud Controls Matrix (CCM)." https://cloudsecurityalliance.org/research/cloud-controls-matrix/. (accessed: 14 May 2023).
12. "5G Security Controls Matrix–ENISA." https://www.enisa.europa.eu/publications/5g-security-controls-matrix/.
13. "free5GC." https://www.free5gc.org/. (accessed: 14 May 2023).
14. K8s, "Kubernetes." https://kubernetes.io/. (accessed: 14 May 2023).
15. "Towards5GS-helm." https://github.com/Orange-OpenSource/towards5gs-helm, Dec. 2022. (accessed: 16 May 2023).
16. Ericsson, "5G Core (5GC) network: Get to the core of 5G." https://www.ericsson.com/en/core-network/5g-core, Dec. 2022. (accessed: 24 May 2023).
17. 3GPP, " TR 33.894 Study on zero-trust security principles in mobile networks," 2022.
18. M. Yang, Y. Li, L. Hu, B. Li, D. Jin, S. Chen, and Z. Yan, "Cross-layer software-defined 5G network," *Mobile Networks and Applications*, vol. 20, pp. 400–409, 2015.
19. "Taxonomy of attacker capabilities." https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks, May 2023. (accessed: 16 May 2023).
20. I. Nadir, Z. Ahmad, H. Mahmood, G. A. Shah, F. Shahzad, M. Umair, H. Khan, and U. Gulzar, "a1-an auditing framework for vulnerability analysis of iot system," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 39–47, IEEE, 2019.

21. Mitre, "MITRE FiGHT." https://fight.mitre.org/. (accessed: 24 June 2023).
22. Mitre, "MITRE ATT&CK." https://attack.mitre.org/, Dec. 2022.
23. N. Tamura, "Sugar: a SAT-based constraint solver.." https://cspsat.gitlab.io/sugar/, Dec. 2022. (accessed: 14 May 2023).
24. "ISO/IEC 27002:2022 Information security, cybersecurity and privacy protection—Information security controls." https://www.iso.org/standard/75652.html, Dec. 2022. (accessed: 14 May 2023).
25. pgmpy, "pgmpy 0.1.19 documentation." https://pgmpy.org/, Dec. 2022.
26. "NetworkX." https://networkx.org/, Jan. 2023. (accessed: 16 May 2023).
27. graphviz, "Graphviz." https://graphviz.org/, May 2023.
28. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.
29. "bert-base-NER." https://huggingface.co/dslim/bert-base-NER, May 2023.
30. "PyDotPlus Homepage." https://pydotplus.readthedocs.io/, May 2023.
31. "Open Source Cloud Computing Infrastructure - OpenStack." https://www.openstack.org/, May 2023. (accessed: 16 May 2023).
32. 3GPPspace, "Inside TS 23.501: AMF Load Balancing," Jan. 2021.
33. X. Ou, S. Govindavajhala, A. W. Appel, *et al.*, "Mulval: A logic-based network security analyzer.," in *USENIX security symposium*, vol. 8, pp. 113–128, Baltimore, MD, 2005.
34. M. Shafayat Oshman, *Assessing Security in the Multi-Stakeholder Premise of 5G: A Survey and an Adapted Security Metrics Approach*. PhD thesis, Carleton University, 2022.
35. S. Nie, Y. Zhang, T. Wan, H. Duan, and S. Li, "Measuring the deployment of 5g security enhancement," in *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2022.
36. S. Bartoletti, G. Bernini, I. Palamà, M. De Angelis, L. M. Monteforte, T. E. Kennouche, K. Tsagkaris, G. Bianchi, and N. B. Melazzi, "Uncertainty quantification of 5g positioning as a location data analytics function," in *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pp. 255–260, IEEE, 2022.
37. Ericsson, "5G security for public and hybrid cloud deployments." https://www.ericsson.com/en/reports-and-papers/further-insights/5g-security-for-hybrid-cloud, Dec. 2022.
38. Spirent, "Keeping Pace with the Requirements of 5G Security." https://www.spirent.com/assets/white-paper-keeping-pace-with-the-requirements-of-5g-security, July 2022. Publisher: Spirent.
39. M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Computing Surveys (CSUR)*, vol. 49, 2016.
40. M. Xie and R. A. May, "Network security framework based scoring metric generation and sharing," Sept. 29 2020. US Patent 10,791,146.
41. S. Majumdar, G. S. Chawla, A. Alimohammadifar, T. Madi, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi, "ProSAS: Proactive security auditing system for clouds," *IEEE Transactions on Dependable and Secure Computing*, 2021.
42. K. Huang, X. Zhang, Y. Mu, F. Rezaeibagha, X. Wang, J. Li, Q. Xia, and J. Qin, "EVA: Efficient versatile auditing scheme for IoT-based datamarket in jointcloud," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 882–892, 2019.
43. G. Apruzzese, R. Vladimirov, A. Tastemirova, and P. Laskov, "Wild networks: Exposure of 5g network infrastructures to adversarial examples," *IEEE Transactions on Network and Service Management*, 2022.

44. F. Boeira, M. Asplund, and M. Barcellos, "Provable non-frameability for 5g lawful interception," in *ACM WiSec*, 2023.
45. M. Orsós, M. Kecskés, E. Kail, and A. Bánáti, "Log collection and siem for 5g soc," in *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 000147–000152, IEEE, 2022.
46. A. Brighente, J. Mohammadi, P. Baracca, S. Mandelli, and S. Tomasin, "Interference prediction for low-complexity link adaptation in beyond 5g ultra-reliable low-latency communications," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8403–8415, 2022.
47. C. J. Mitchell, "The impact of quantum computing on real-world security: A 5g case study," *Computers & Security*, vol. 93, p. 101825, 2020.
48. ETSI, "Network Functions Virtualisation (NFV) Release 4 Security; Security Management Specification," Jan. 2021.
49. 3GPP, "TS 23.288 architecture enhancements for 5G System (5GS) to support network data analytics services v17.4.0 (2022-03)," 2022.
50. A. R. Prasad, S. Arumugam, S. B., and A. Zugenmaier, "3gpp 5g security," *J. ICT Stand.*, vol. 6, no. 1-2, pp. 137–158, 2018.
51. J. R. Hamlet and C. C. Lamb, "Dependency graph analysis and moving target defense selection," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pp. 105–116, 2016.

Appendix

A Definition of Event-State Model

To evaluate security postures in 5G, we build a model, namely, *event-state model* (a combination of the event model and state model). The *state model* captures the results of auditing system states, and the *event model* captures the results of monitoring system events. We define these models more formally as follows.

Event Model (EM). Given a list of event types *event-types* and the log of historical events *hist*, the event model is defined as a Bayesian network $EM = (G_e, E_e)$, where G_e is a directed acyclic graph (DAG) in which each node represents an event type in *event-types*, and each directed edge between two nodes indicates the first node would immediately precede the other in some event sequences in *hist* whose probability is part of the list of parameters E_e.

State Model (SM). Given a list of breaches of different security control from different security standards, B and the pre-, and post-conditions of individual breach, P, and the auditing logs of the system over time *auHist*, the state model is defined as a dependency graph [51] $SM = (G_s, E_s)$, where G_s is a set of DAGs in which each node corresponds to the breach of security control and their pre-or post-condition from B and P, and each directed edge between two nodes indicates the transition probability derived from *auHist* and it is part of the list of parameters E_s.

Event-State Model (ESM). Event-State Model is a Bayesian network $ESM = (G_c, E_c)$, where $G_c = \{G_e \cup G_s\}$ (i.e., all the nodes in both the event and state models) and $E_c = \{E_e \cup E_s \cup E_p\}$, where E_p is the set of edges that connects the privilege escalation vertices (responsible for lateral movement caused by a

breach) to the resulted vertices (either from *EM* or *SM*) and the edge values are coming from *hist* and *auHist* as a probability which is part of the list of parameters E_c.

B Algorithm for Building Event-state Model

Algorithm 1 is used to construct an event-state model from the event model, and the state model. We define two distinct functions: `vertical_fusing` (Lines 3 to 16), and `horizontal_fusing` (Lines 17 to 28) to combine the model vertically and horizontally. For vertical fusing, in Line 6, we first list all the privilege escalation nodes manually by taking help from an expert. Then, for each privilege escalation node, we attach the breach node and the event node to the privilege escalation node in Lines 9–13. In Lines 29–36, we define one utility function named `findCommonNode` to list all the common nodes between two models for horizontal fusing. Line 18 of the `horizontal_fusing` function utilizes this utility function. Line 23 and 24 adds the parent and child subgraphs from both the event model and the state model to the common node.

Algorithm 1: Building event-state Model

```
   Input:
            stateModel ← state model;
            eventModel ← event model;
   Output:
            combinedModel ← CombinedModel
 1 combinedModel ← vertical_fusing(stateModel, eventModel);
 2 combinedModel ← horizontal_fusing(stateModel, eventModel);
 3 Function vertical_fusing(stateModel, eventModel):
 4 |   combinedModel ← [];
 5 |   foreach breachState in stateModel do
 6 |   |   privilegeEscalationNode ← breachState.previlegeEscalation;
 7 |   |   normalEvent ← eventModel.breachState;
 8 |   |   if privilegeEscalationNode ∉ combinedModel then
 9 |   |   |   combinedModel.add_node(privilegeEscalationNode) ;
10 |   |   |   combinedModel.add_node(breachState) ;
11 |   |   |   combinedModel.add_node(normalEvent) ;
12 |   |   |   combinedModel.add_edge(node1, newNode) ;
13 |   |   |   combinedModel.add_edge(newNode, node2)
14 |   |   end
15 |   end
16 |   return combinedModel;
17 Function horizontal_combnination(stateModel, eventModel):
18 |   commonNodeList ← findCommonNode(stateModel, eventModel);
19 |   foreach cnode in commonNodeList do
20 |   |   if cnode ∉ combinedModel then
21 |   |   |   if isCreateCylce(cnode, combinedModel) == False then
22 |   |   |   |   combinedModel ← add_node(cnode);
23 |   |   |   |   combinedModel ← add_subgraph(cnode.parent);
24 |   |   |   |   combinedModel ← add_subgraph(cnode.child);
25 |   |   |   end
26 |   |   end
27 |   end
28 |   return combinedModel;
29 Function findCommonNode(stateModel, eventModel):
30 |   listOfCommonNodes ← ∅;
31 |   foreach node ∈ stateModel do
32 |   |   if node ∈ eventModel then
33 |   |   |   listOfCommonNodes ← listOfCommonNodes ∪ {node};
34 |   |   else
35 |   end
36 |   return listOfCommonNodes;
37 return combinedModel;
```