HAL open science

A Grammar-Driven Approach to Model and Detect APT Attack Sequences

Antoine Rebstock, Yann Busnel, Romaric Ludinard

To cite this version:

Antoine Rebstock, Yann Busnel, Romaric Ludinard. A Grammar-Driven Approach to Model and Detect APT Attack Sequences. FPS 2025: 18th International Symposium on Foundations & Practice of Security, IMT Atlantique, Nov 2025, Brest, France. ⟨hal-05521004⟩

HAL Id: hal-05521004

https://hal.science/hal-05521004v1

Submitted on 20 Feb 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

A Grammar-Driven Approach to Model and Detect APT Attack Sequences

Antoine Rebstock[1][0009−0005−9240−8787], Yann Busnel[2][0000−0001−6908−719X], and Romaric Ludinard[1][0000−0002−4997−4813]

[1] IMT Atlantique / IRISA – firstname.lastname@imt-atlantique.fr
[2] Institut Mines-Télécom / IRISA – yann.busnel@imt.fr

Abstract. This paper introduces a grammar-driven approach to model and detect sequences of Advanced Persistent Threat (APT) attacks. APTs are characterized by their complexity and multi-step nature, making them challenging to detect with traditional systems. The proposed method employs interpretable rules to capture malicious behaviors and aligns semantically with the MITRE ATT&CK framework. This approach enables the detection of plausible attack scenarios even in the presence of incomplete or noisy data. We evaluate it using a public large CTF dataset which provides realistic and diverse attack scenarios. Experiments illustrate the effectiveness of the method in reconstructing plausible attack progressions, even with incomplete data. The main contribution of this study is an open-source implementation in Rust, ensuring reproducibility and extensibility. We also propose future enhancements to better model contextual dependencies between tactics in APT attack sequences. In a nutshell, this grammatical approach offers a robust method for detecting sophisticated threats, bridging the gap between low-level observations and high-level strategic reasoning.

1 Introduction

In recent years, cyberattacks have evolved into sophisticated, multi-step scenarios involving sequences of malicious actions designed to achieve goals such as data theft or destruction. Among these, Advanced Persistent Threats (APTs) are particularly challenging due to their stealth, persistence, and adaptability [1]. APTs are resource-intensive, which is why they are usually associated with state actors, typically pursuing long-term objectives such as espionage or sabotage. They often rely on obfuscation strategies such as mimicking legitimate behavior, spreading actions over time or exploiting 0-day vulnerabilities to evade detection.

This deceptive nature creates major challenges for detection and investigation. Traditional Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) often struggle to highlight such threats due to the massive volume of logs to correlate across extended timeframes, the subtlety of malicious activity hidden among leggit events, the limited monitoring coverage and the potential data loss caused by limited retention. Moreover, logging policies influence both the volume of collected data and the risk of missing

relevant information, thereby affecting the effectiveness of subsequent analysis. These factors impede cybersecurity analysts' efforts to reconstruct a coherent and comprehensive view of the attack.

To better detect and understand these complex threats, detection systems must evolve toward models that capture the progression and logic of multi-step malicious behaviors over time. This requires moving beyond isolated events or alerts and toward more structured representations that preserve topological, temporal and causal links between actions.

In this paper, we propose a novel grammar-based approach to model multi-step attack scenarios. By representing malicious behaviors as interpretable and composable rules over sequences of actions, our method captures dependencies between steps and enables the detection of plausible multi-step scenarios even in the presence of missing or noisy data. Our approach relies on the MITRE ATT&CK[1] framework, a well-established and recognized knowledge base for modeling hostile behaviors, ensuring semantic alignment with known adversarial techniques. While this framework is regularly updated, it is inherently limited to documented techniques and cannot account for unknown vulnerabilities or zero-day exploits. Dynamically enriching this knowledge base—particularly through generative AI or unsupervised learning—falls outside the scope of this study. Our goal is to demonstrate the feasibility of structured, interpretable detection of multi-step attacks using existing knowledge, rather than proposing a mechanism for discovering novel vulnerabilities or previously unseen behaviors. This limitation is intrinsic to any rule- or signature-based approach and highlights opportunities for future work combining grammatical modeling with emerging behavioral analysis techniques.

This paper is organized as follows. Section 2 reviews the state of the art and related literature. Section 3 first motivates the use of a grammatical model to describe multi-step attacks, highlighting the benefits in terms of expressiveness, correlation, and robustness. On the other hand, this section includes a categorization of MITRE ATT&CK Techniques and a structured hierarchy to unify heterogeneous log sources. Section 4 presents the evaluation metrics used to assess the proposed approach. Finally, Sections 5 and 6 respectively explores the future works and concludes this paper.

2 Related work

The detection and investigation of multi-step attacks, particularly Advanced Persistent Threats (APTs), have prompted the development of a diverse array of methodologies aimed at reconstructing coherent attack narratives from incomplete and noisy data. These methodologies can be broadly classified into five kinds of approaches [2]: case-based, structural-based, causal correlation, similarity-based, and mixed. Each approach presents distinct advantages and inherent limitations with respect to accuracy, scalability, and robustness.

[1] `https://attack.mitre.org/matrices/enterprise/`

Case-based approaches rely on matching incoming traces to predefined attack templates or known scenarios. A prominent example is STATL [3], which models known attack sequences as state transitions and detects intrusions by aligning system events with these predefined patterns. While such methods are effective against replay attacks, they lack flexibility in the face of novel or evolving threat behaviors. This rigidity has motivated a shift toward more dynamic and generalizable approaches.

Structural-based methods, such as the one proposed by Lanoe *et al.* [4], interpret observed events in the context of a model representing the underlying network or system. By validating whether observed attack steps align with feasible paths in the system's topology, these approaches enable predictive analysis of potential future steps. However, they tend to be tightly coupled to environment-specific configurations, limiting portability and generalization across diverse infrastructures.

Causal correlation approaches aim to infer meaningful dependencies between events in an attack sequence. However, they exhibit limited robustness in detecting novel or previously unseen behaviors. Ren *et al.* [5] illustrate a statistical inference sub-kind by employing Bayesian networks, dynamically inferring probabilistic relationships between observed events to guide intrusion response. Other model matching approaches, such as those proposed by Milajerdi *et al.* [6] and Alsaheel *et al.* [7], map system log activities to kill chain steps using expert-defined rules derived from MITRE ATT&CK and enhance this with deep sequence learning. These methods reconstruct plausible and semantically rich attack narratives even from sparse or noisy data. Wilkens *et al.* [8] propose using cyber kill chain-based state machines to model an attacker's progression, building alert graphs that reflect the causal evolution of attacks and improve situational understanding. Alternatively, Zhang *et al.* [9] and Al-Mamory and Zhang [10] introduce grammar-based approaches that model attack sequences and represent known attack scenarios as structured sequences of alerts. While these grammar models are more expressive and compact than traditional attack graphs, they rely solely on alert data and do not provide a generic modeling of multi-step attacks.

Similarity-based methods, on the other hand, operate by measuring how closely new traces resemble known patterns without explicitly modeling causal relationships. For instance, HERCULE [11] leverages attribute correlation, assuming that events linked to the same attack share common attributes such as IP addresses or user IDs, thus identifying potential malicious links through similarity in event features. Log2Vec [12] adopts a scenario clustering approach by converting log entries into heterogeneous graphs representing entities and relations (users, objects, actions, hosts, time), then learns embeddings of these graphs to cluster and detect anomalous behavior patterns. DeepLog [13] focuses on anomaly detection through sequential similarity by training LSTM models on benign log sequences to predict expected events, flagging deviations as anomalies. MAAC [14] extends this perspective by vectorizing the semantic content of alert fields to cluster similar alerts into super-alerts, labeling each according to its corresponding attack step (*e.g.*, privilege escalation). These labeled alerts

are linked to form multi-step attack graph, enabling automatic extraction of the most suspicious paths. These methods predominantly employ unsupervised learning to model normal behavior, detecting deviations as potential threats. While scalable and adaptable, their lack of explicit causal reasoning often leads to false positives and limits interpretability regarding attack progression.

Despite these advances, several limitations persist. First, as highlighted by Navarro *et al.* [2], the field still lacks a precise and operational definition of what constitutes a valid link between steps of an attack. Most methods continue to rely on manually coded pre-conditions and post-conditions, making the notion of causality a human-defined construct rather than a formally grounded one. Additionally, the overwhelming majority of approaches are built around alerts generated by IDS, even though many relevant attack steps can be found in other types of system, application, or network logs. The heavy reliance on expert-crafted knowledge further limits the scalability and adaptability of these systems in the face of novel or evolving threats. Moreover, as Ingale *et al.* [15] point out, most existing methodologies struggle to cope with rare or missing data, a common issue in real-world environments where attack traces may be uncommon or incomplete. This deficiency hampers the correct inference of attack sequences, as each step's interpretation often depends on the successful detection of the previous one. Consequently, there is a pressing need for models that remain robust under uncommon data and that can maintain predictive capabilities even when some events are missing.

3 Towards a grammatical model of attacks

Detecting multi-step attacks requires understanding how individual malicious actions unfold in a structured, goal-oriented manner. Attackers typically perform sequences of actions, each contributing to intermediate objectives that ultimately serve a broader goal, such as data theft or sabotage. However, this process is rarely linear. Attackers often explore multiple paths in parallel, testing different approaches before finding or settling on an effective course of action. As a result, the attack progression can be viewed as a tree-like structure, where one path from root to leaf represents a full, successful attack scenario.

As described in [16], this progression can be conceptualized as an operational flow, a sequential representation of attacker behavior, where each step reflects a specific MITRE ATT&CK Technique contributing to the overall objective. Through the concept of propagation area, they also argue that each attacker's action is strongly influenced by his position within the network topology and his level of situational awareness regarding the environment.

To faithfully capture the semantic significance of attacker actions and their interdependencies, a more expressive modeling framework than provenance graph is needed. While graphs can encode relationships between events, they often fail to capture their goals, conditional prerequisites, and context-dependent constraints that shape an attacker's behavior. For instance, a lateral movement action is not just a step connected to a previous one: it typically requires the

attacker to possess valid credentials, know the target host, and have a foothold in the current host, all of which depend on prior actions and observations. A grammar-based model makes it possible to represent these constraints explicitly and to define recursive, conditional, or alternative paths, reflecting the strategic flexibility of Advanced Persistent Threats. This approach thus provides a richer abstraction than provenance graphs, supporting a more nuanced interpretation of attack scenarios.

By framing attack detection as a parsing problem over grammatically valid sequences of Techniques, we aim to capture the intentional structure of multi-step attacks and assist analysts in identifying meaningful and coherent scenarios, even in noisy or incomplete logging environments.

3.1 Semantic enrichment using MITRE ATT&CK Techniques

One major challenge in applying this grammatical model lies in the heterogeneity of event representations across logging systems. A single attacker objective can manifest differently depending on:

- the observation layer involved (*e.g.*, system-level, network-level, application-level) ;
- the execution environment (*e.g.*, GNU/Linux, Windows) ;
- the logging source (*e.g.*, auditd, syslog, Windows Event Logs, or IDS tools such as Suricata or Zeek).

As a result, a given attack step may surface in multiple syntactic forms, originate from distinct infrastructure components, or vary in granularity and completeness of semantic content, which affects interpretability. This diversity poses significant challenges for event correlation and for projecting raw observations onto high-level representations of attack steps.

To address this challenge, we propose a semantic normalization layer that abstracts events into MITRE ATT&CK Techniques, regardless of the events' originating layer, execution environment, or logging source. Concretely, we consider a MITRE ATT&CK Technique to be characterizable by a sequence of events, which we model using state machines to capture their temporal and logical structure. This approach supports flexible representations of a single Technique by capturing the diverse execution paths an attacker may follow, allowing certain events to be designated as optional when they are less critical, and enabling the detection of Techniques that manifest only through coherent sequences of related events rather than through isolated events.

For instance, Figure 1 illustrates a state machine specifically designed to model process behavior with the objective of detecting Privilege Escalation. The model relies on the analysis of process identifiers (PIDs) and their parent-child relationships (PPIDs), alongside monitoring changes in user privilege levels as indicated by the effective user identifier (EUID). In this context, a Privilege Escalation is characterized as an event in which a process initially running without

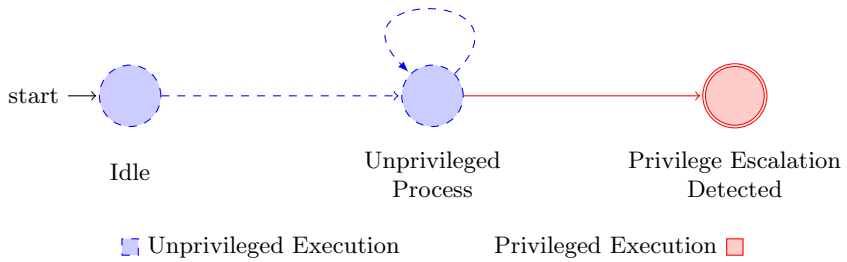

Fig. 1: Exploitation for Privilege Escalation State Machine

elevated privileges (*i.e.*, with a EUID different from 0) initiates or executes another process that operates with elevated privileges (EUID = 0, corresponding to the root user).

By abstracting away low-level heterogeneity and retaining only the events necessary to infer the presence of MITRE ATT&CK Techniques, this projection enables the aggregation of partial and noisy logs into coherent entities. These unified representations serve as the foundation for our grammatical framework, allowing automated systems to reason about attacks at a more strategic level, grounded in a shared, standardized vocabulary of attacker behavior.

3.2 Functional categorization of Tactics

Having established the semantic normalization of techniques, which abstracts low-level events into coherent behavioral patterns, let focus on the strategic dimension of attacker behavior as delineated by MITRE ATT&CK Tactics. While Techniques describe how specific malicious actions are performed, Tactics convey why they are executed by representing the adversary's high-level objectives during various phases of an intrusion.

To enhance the modeling of attacker progression through an information system, we therefore propose a functional categorization of these Tactics based on their operational role within the kill chain. This classification aims to clarify the broader intent behind observed Techniques and facilitates reasoning about attack goals and trajectories in a more structured and actionable manner. We define the following categories:

GetInfo Tactics aimed at gathering information about the network, users, or system configuration ;

MoveUser Tactics involving movement by switching between user accounts ;

MoveHost Tactics involving movement across hosts ;

Objective Final-step Tactics such as data exfiltration or impact ;

External Tactics occurring outside the monitored information system perimeter ;

Tactics	GetInfo	MoveUser	MoveHost	Objective	External	NoProgress
Reconnaissance	✔					
Credential Access	✔					
Discovery	✔					
Collection	✔					
Privilege Escalation		✔				
Initial Access			✔			
Lateral Movement			✔			
Exfiltration				✔		
Impact				✔		
Resource Development					✔	
Execution						✔
Persistence						✔
Defense Evasion						✔
Command & Control						✔

Table 1: Distribution of MITRE Tactics by functional categories.

NoProgress Tactics that, while malicious, do not independently enable the attacker to progress either topologically or informationally (*e.g.*, persistence, defense evasion).

Table 1 illustrates how standard MITRE ATT&CK Tactics are distributed among these six functional categories.

Among these categories, GetInfo, MoveUser, MoveHost, and Objective appear to be the most relevant for capturing both the informational and topological progression of an attacker. In particular, GetInfo plays a critical enabling role by supplying the attacker with the knowledge required to carry out subsequent actions – whether in MoveUser, MoveHost, Objective, or further GetInfo activities. This functional categorization should allow us to explicitly model data dependencies between Techniques, reflecting how an attacker's growing knowledge of the environment conditions their capacity to act upon it. For example, lateral movement presupposes control over the starting position, as well as knowledge of both a reachable host and valid credentials information acquired during prior discovery steps.

3.3 From functional categories to logs: a multi-level representation

To improve the understanding of the structure and interdependencies of attack steps, we propose a top-down hierarchical model, as depicted in Figure 2. This model, grounded in the functional categorization introduced earlier, establishes a connection between the semantic abstraction of attacker actions – their functional roles – and the concrete data utilized for their detection.

```
Functional category
├─ Tactic (MITRE ATT&CK) #1
│  ├─ Technique (MITRE ATT&CK) #1
│  │  ├─ Sub-technique (MITRE ATT&CK) #1
│  │  │  ├─ Events
│  │  │  │  ├─ Required data type (system, network)
│  │  │  │  │  ├─ Data source(s) (log type)
│  │  │  │  │  │  ├─ Data (log entries)
│  │  │  │  │  │  └─ ...
│  │  │  │  │  └─ ...
│  │  │  │  └─ ...
│  │  │  └─ ...
│  │  └─ Sub-technique (MITRE ATT&CK) #2
│  │     └─ ...
│  └─ Technique (MITRE ATT&CK) #2
└─ ...
```

Fig. 2: Top-down hierarchy to relate functional categories in low level logs.

This structured view serves multiple purposes:

- It enables the semantic unification of heterogeneous events by anchoring them in a common functional and tactical context.
- It supports structured inference, allowing one or more log entries to be associated with a Technique via a reasoning process that considers both the content and context of the event.
- It provides a foundation for topological anchoring – *i.e.*, associating each attacker action with a specific location in the system (*e.g.*, host, user), which is crucial for modeling propagation across an infrastructure.

This structured view enables semantic normalization, context-aware inference, and topological anchoring. Together, these dimensions contribute to a richer and more precise grammatical model of attacker behavior – one that goes beyond linear sequences to incorporate the underlying dependencies that drive attacker progression. We explore these inter-technique dependencies in the next section.

3.4 Progression model and grammar

Multi-step attacks can be naturally interpreted as sequences of transitions between MITRE ATT&CK Techniques, where each step reflects a concrete adversarial action serving a precise tactical objective (which Berady *et al.* [16] refers to *operational flow*). These transitions are not arbitrary; they follow patterns that reflect the attacker's evolving goals, system knowledge, and topological position within the infrastructure. To capture this structured progression, we introduce an abstract grammar that models how techniques can validly follow one another.

This grammar serves as a template for plausible attack scenarios, enabling the system to detect sequences that exhibit coherent strategic intent rather than isolated suspicious events. By modeling attack progression as a grammatical construct, we can formalize not only the expected order and dependencies between actions, but also provide a basis for automated parsing of complex behavior across time and space. This structured view is particularly well-suited to detecting stealthy, long-term threats such as APTs.

To formalize the structure of multi-step attacks, we define a set of production rules within a sequential production model. These rules represent typical APT behavior as a sequence of transitions between MITRE ATT&CK Techniques. The grammar presented below offers a high-level abstraction illustrating how such attacks evolve:

$$
\begin{aligned}
\langle \mathsf{ATTACK} \rangle &\rightarrow \langle \mathsf{A} \rangle \ \langle \mathsf{B} \rangle \ \langle \mathsf{C} \rangle \\
\langle \mathsf{A} \rangle &\rightarrow \text{Reconnaissance+ InitialAccess} \\
\langle \mathsf{B} \rangle &\rightarrow \text{(Discovery} \mid \text{CredentialAccess)+} \\
\langle \mathsf{C} \rangle &\rightarrow \{\{\langle \text{OBJ} \rangle \mid \text{PrivilegeEscalation} \mid \text{LateralMovement}\}+ \ \langle \text{B} \rangle\}+ \\
\langle \mathsf{OBJ} \rangle &\rightarrow \text{Exfiltration} \mid \text{Impact}
\end{aligned}
$$

This model captures three main phases commonly observed in APT attack scenarios:

1. **Initial Access Phase** – $\langle \mathsf{A} \rangle$: The attacker begins with a Reconnaissance Technique to gather information about the target, followed by an Initial Access (*e.g.*, exploiting a vulnerability or using previously obtained credentials). For example, access might be achieved by initiating an SSH connection to a target system using valid login information acquired earlier through external means.
2. **Exploration Phase** – $\langle \mathsf{B} \rangle$: Once access is gained, the attacker proceeds with discovery techniques (*e.g.*, enumerating files, users, or network hosts) and possibly credential access techniques to escalate privileges or prepare for lateral movement.
3. **Objective-Oriented Phase** – $\langle \mathsf{C} \rangle$: The final phase includes the actual objectives, such as exfiltration or impact techniques, possibly interleaved with several privilege escalations or lateral movements to reach more sensitive assets. The recursive structure of $\langle \mathsf{C} \rangle$ allows for iterative progression, reflecting the adaptive and persistent nature of APTs.

This grammatical structure allows us to constrain detection and analysis to valid, strategically coherent sequences, rather than arbitrary event chains. It also provides a foundation for parsing real-world event streams into high-level attack scenarios, enabling both detection and forensic reconstruction.

As a toy example, based on the following raw logs, thanks to our grammar, the rule `LateralMovement PrivilegeEscalation` would capture the common progression in which an attacker moves laterally across systems before escalating privileges locally.

```
% Lateral Movement
type=USER_LOGIN msg=audit(1715985978.277:44113): [...] id=1001
[...] addr=10.35.229.10 [...] res=success

% Privilege Escalation
type=SYSCALL msg=audit(1715986193.010:44385): [...] syscall=59
success=yes [...] pid=10113 [...] uid=1001 [...]
euid=1001 [...] comm="sh" exe="/usr/bin/dash" [...]
type=PATH msg=audit(1715986193.010:44385): [...] name="/bin/sh"
[...] mode=0100755 [...]

type=SYSCALL msg=audit(1715986193.010:44386): [...] syscall=59
success=yes [...] ppid=10113 pid=10114 [...] uid=1001 [...]
euid=0 [...] comm="file"
exe="/tmp/.yolo/CVE-2023-0386/ovlcap/upper/file" [...]
type=PATH msg=audit(1715986193.010:44386): [...]
name="./ovlcap/upper/file" [...] mode=0104777 [...]
```

In this case, the sequence was identified through the detection of a successful remote login (Remote Services Technique) followed by the execution of a process with elevated privileges by a parent process that initially lacked such privileges (Exploitation for Privilege Escalation Technique).

4 Experimentation results

In this study, we advance the field of grammar-based APT detection with an open-source Rust implementation[2], ensuring full reproducibility and supporting future research and development. Our extensible framework allows for the integration and refinement of MITRE ATT&CK Techniques, enhancing the modeling of attacker behaviors. A key innovation is our fully modifiable grammatical rule system, which adapts to evolving threat landscapes. Leveraging Rust's performance and safety features, our solution bridges the gap between theoretical models and practical applications. This section presents our experimental validation, demonstrating the effectiveness and adaptability of our approach in detecting APT sequences.

4.1 Implementation design

To address the requirements of performance, safety, and expressiveness in APT detection through a grammar-based model, we selected the Rust programming

[2] `https://gitlab.imt-atlantique.fr/sotern-public/apt-grammar/`

language for its technical and methodological advantages. Rust ensures memory safety and high execution performance, both of which are critical for real-time processing of large-scale, heterogeneous log data, such as in semantic enrichment pipelines and event parsers. Its expressive type system and ownership model support the development of modular and statically verifiable parsing frameworks, enabling the safe encoding of grammars to model complex, stateful attack behaviors like those described in the MITRE ATT&CK framework. Additionally, Rust's mature ecosystem – featuring libraries for parser combinators (*e.g.*, `nom`[3]), finite automata (*e.g.*, `rust_fsm`[4]), and graph modeling – facilitates the definition of high-level abstractions for grammatical rule formalization and attack sequence representation. Importantly, Rust's design philosophy aligns with modern security engineering practices, prioritizing safety and performance by default – an essential consideration in the privileged, adversarial context of APT detection systems, where robustness and attack surface minimization are imperative. To bridge the gap between low-level system supervision and high-level adversarial behavior detection, we leverage a parser combinator approach. Parser combinators are recursive descent parsers; thus, they fit perfectly with our grammar design. In addition, any modification of the grammar, for instance due to the discovery of a new vulnerability, is facilitated by the recursive nature of our approach and thus eases modular piecewise construction. Indeed, due to the nature of the parsed raw logs, which represents actions of attackers, it allows designing a low-level parser. Results of these parsers are then aggregated by higher-level parsers, allowing us to rebuild a global attack scenario. Importantly, only logs that match the grammar rules are retained, ensuring that the system focuses exclusively on relevant events for attack reconstruction.

All experiments were conducted on a workstation equipped with a 12th Gen Intel® Core™ i7-1250U CPU (10 cores, 12 threads), 16 GB RAM, and an integrated Intel Iris Xe Graphics GPU. The system ran Ubuntu 24.04.2 LTS with kernel version 6.11.0-29-generic. The software environment included Rust (cargo 1.88.0). Code execution was performed on an SSD-based system to ensure fast I/O.

4.2 Evaluation

Our evaluation leverages the CasinoLimit publicly available dataset [17], a controlled and MITRE ATT&CK-annotated scenario, to demonstrate the feasibility of our approach in a realistic yet manageable setting. This deliberate choice emphasizes the method's strengths–such as robustness to missing data and interpretability–while exposing its limitations, particularly regarding probe placement and log coverage. For instance, steps involving credentials acquired through unmonitored channels (*e.g.*, a webcam) cannot be detected without ad-hoc instrumentation. The evaluation does not claim to cover all realistic scenarios or assess large-scale scalability. Instead, it serves as a proof of concept,

[3] `https://docs.rs/nom/latest/nom/`
[4] `https://docs.rs/rust-fsm/latest/rust_fsm/`

validating our model's ability to reconstruct plausible attack sequences despite inevitable gaps in collected data. Future work will focus on broader evaluations across heterogeneous environments and varied logging policies to further assess generalizability.

The CasinoLimit dataset offers realistic and large-scale data, particularly due to its high number of participants, making it well-suited for testing multi-step attack detection methods. It contains system and network activity logs collected over a 12-hour period from 99 human participants, each engaging independently in their own isolated instance of a complex red team challenge. The participants must move through the information system to reach a predefined goal, which naturally generates sequences of actions that reflect the progression of the multi-step attack and closely reflect real-world adversarial behavior. For evaluation purposes, we have chosen to rely on the 9 attackers who have successfully completed the attack. Additionally, the dataset is labeled using the Techniques of the MITRE ATT&CK framework, enabling fine-grained and semantically meaningful evaluation of technique- and tactic-level detection models.

This dataset addresses several limitations observed in traditional benchmarks for intrusion detection. Prior studies have highlighted that commonly used datasets are outdated, lack full coverage of the attack lifecycle [1], and fail to reflect the complexity and diversity of modern adversarial behavior [18]. Furthermore, many of these datasets were originally designed for simple IDS and are poorly suited for the analysis of sophisticated, multi-step attacks [19]. By contrast, the CasinoLimit dataset offers up-to-date, realistic, and richly annotated data tailored to these complex detection scenarios. It captures a wide range of attacker behaviors and technical skill levels, providing a heterogeneous set of attack patterns. This makes it particularly valuable for evaluating models focused on detecting APT sequences. In addition, due to the predefined nature of the attack scenario and the intentionally vulnerable information system, all attackers are constrained to follow the same steps to succeed, allowing to assess our approach and identify its limits.

To begin with, the attacker gains access to the start host (`h1`). His aim is to delete data from a database on the intranet host (`h4`) - *i.e.*, to perform an application-level operation targeting specific records concerning him. To do this, he must pass through the meetingcam host (`h2`) to retrieve information enabling him to access the bastion host (`h3`), then repeat this process from the bastion host (`h3`) to the intranet host (`h4`).

Since our approach is modular and extensible, allowing the integration of new data sources, we have chosen to focus on the processing of audit logs to demonstrate its versatility. As mentioned previously, logs corresponding to the steps involved in accessing `h4` and deleting information from the database are application dependent and are therefore not taken into consideration due to the lack of appropriate logging rules. For the purposes of evaluation, we have considered an attack sequence to be detected if we were able to detect it up to the right Privilege Escalation on `h3`, taking into account the tolerated number k of missing steps.

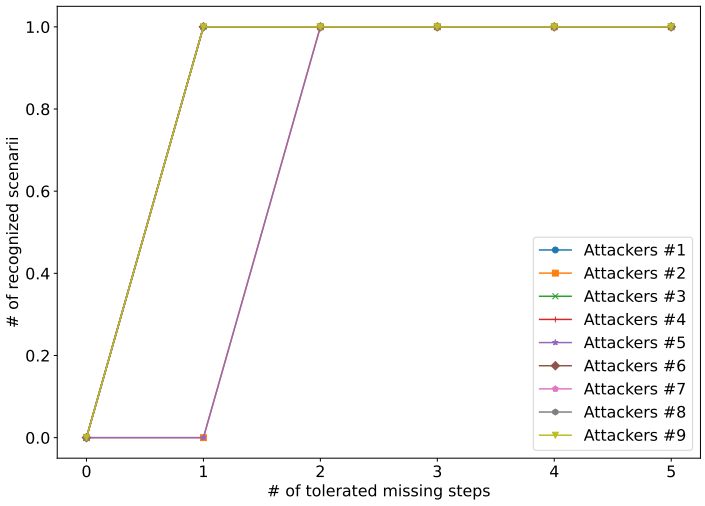

Fig. 3: Evaluations of APT grammar as a function of k – Number of recognized scenarii

Figures 3–5 summarize the detection results under varying tolerance thresholds k for missing steps, *i.e.* missing logs due to supervision policy. In particular, Figure 3 shows that our approach successfully detects attack sequences across all scenarios when a reasonable tolerance is allowed, confirming its robustness to incomplete log data. This flexibility is crucial in practical settings, where perfect logging coverage is rarely guaranteed. However, the detection fails correspond to cases in which the audit logging policy does not capture key events required to identify specific steps in the attack sequence. Indeed, Figure 4 and Figure 5 highlight the dependency of detection capabilities on the granularity and completeness of available log data: the number of plausible scenarios, as well as the time (in seconds) required to detect those scenarios, increases with the number of missing steps. This reinforces the importance of comprehensive and well-aligned logging strategies in real-world deployment scenarios.

Figure 6 illustrates the durations for extracting Techniques across various batches corresponding to each attacker logs, with minimum and maximum values depending on k. It is noteworthy that the parameter k has minimal impact on these durations. Instead, the size of the log plays a significant role in determining extraction times. However, it should be observed that these log sizes are not directly proportional to the durations due to the combinatorial nature of the extraction process. Despite these challenges, our solution demonstrates high performance given the task's complexity and the real-time constraints of detection systems.

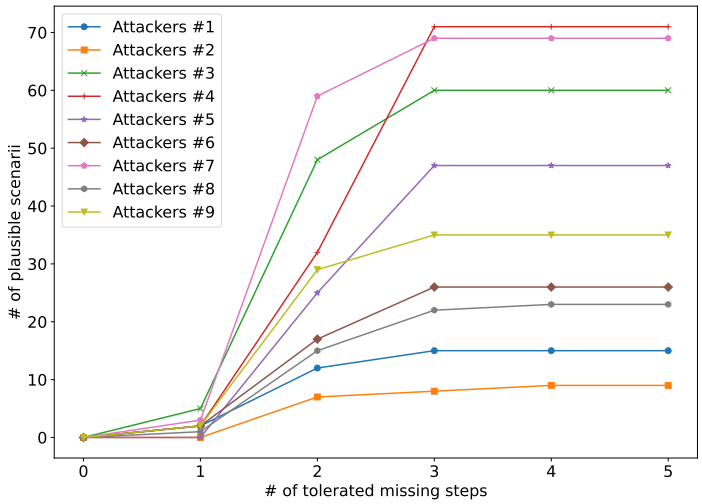

Fig. 4: Evaluations of APT grammar as a function of k – Number of plausible scenarii

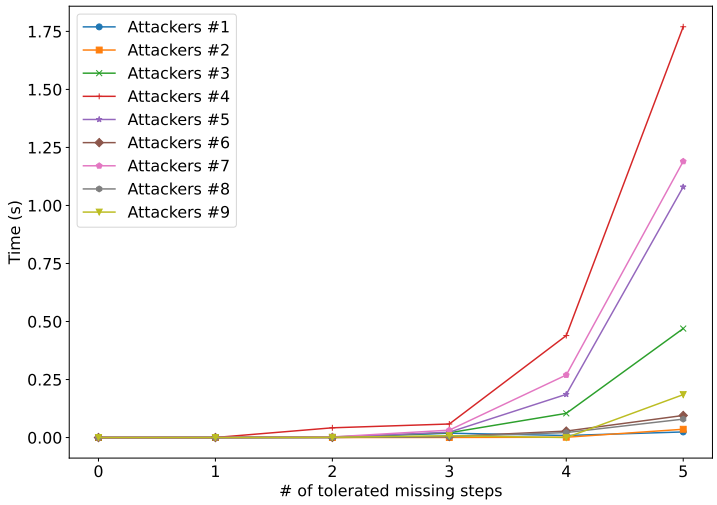

Fig. 5: Evaluations of APT grammar as a function of k – Time to detect plausible scenarii

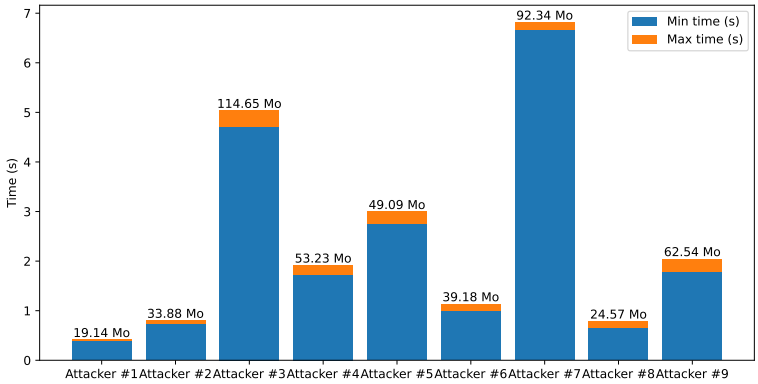

Fig. 6: Time to extract Technics as a function of attacker identifier.

5 Future Works

A promising direction for future work lies in deepening the modeling of contextual dependencies between tactics in APT attack sequences. While our current grammatical framework accounts for temporal progression and semantic plausibility, it does not yet explicitly enforce or leverage the full range of dependencies that exist between Tactics. Future enhancements could address several types of dependencies more formally:

Informational dependency a Tactic requires prior knowledge acquired from another Tactic. For instance, performing lateral movement via "Remote Services" (MoveHost) such as SSH requires that the attacker has identified the existence and accessibility of a target host (GetInfo). This typically involves prior network scanning or host discovery (*e.g.*, via "Network Service Discovery" or "Remote System Discovery") to determine which systems are reachable and which services are exposed.

Topological dependency a Tactic is constrained by the attacker's current position within the infrastructure. For instance, a privilege escalation (MoveUser) can only occur on a host where the attacker already has a foothold. Similarly, lateral movement (MoveHost) requires network access to the destination host.

Scalability In the current approach, each log entry is exhaustively correlated with the complete set of techniques defined in the MITRE ATT&CK framework. To mitigate the resulting computational overhead, we propose leveraging both information-flow and topological dependencies to enable more efficient reasoning over the data. This strategy is expected to substantially reduce processing time while maintaining analytical completeness, thereby

improving the method's applicability to enterprise-scale logging systems or high-volume real-time environments.

To achieve these aims, future extensions of our grammatical model could incorporate explicit dependency tracking, either through enriched transition annotations or through external contextual state models.

By combining grammatical structure with contextual awareness, our approach could support plausibility-based reasoning: it could highlight which transitions are semantically coherent and topologically grounded, without excluding alternative paths when information is partial or ambiguous. This flexibility could help reduce false positives and enhance the interpretability of incomplete sequences.

6 Conclusion

In this work, we introduced a grammatical approach to modeling APT attacks, grounded in the MITRE ATT&CK framework and designed to capture both the semantic structure and contextual dependencies of adversarial behavior. By representing these attack sequences as interpretable constructs within structured grammar, we propose a method to detect coherent attack progressions even in noisy or incomplete log environments. Our approach addresses key challenges in detecting these threats, including the need for higher-level abstraction beyond events inferred from raw logs. By semantically enriching events into MITRE ATT&CK Techniques and functionally categorizing Tactics, we build a multi-level representation that supports contextual reasoning, topological anchoring and structured inference. This enables attacker behavior to be identified through a formalism that reflects real-world attack strategies. Experimental results on the CasinoLimit dataset demonstrate the feasibility of our method in realistic contexts. Despite the presence of incomplete data and variability between scenarios, our grammar-based model succeeded in reconstructing plausible attack progressions and proved robust to missing steps. These results suggest that grammar-based modeling can effectively bridge the gap between low-level observations and high-level strategic reasoning.

References

[1] B. Stojanović, K. Hofer-Schmitz and U. Kleb. APT datasets and attack modeling for automated detection methods: A review. *Computers & Security* 92 (2020).

[2] J. Navarro, A. Deruyver and P. Parrend. A systematic survey on multi-step attack detection. *Computers & Security* 76 (2018).

[3] S. T. Eckmann, G. Vigna and R. A. Kemmerer. STATL: An attack language for state-based intrusion detection. *Journal of computer security* 10.1-2 (2002).

[4] D. Lanoe, M. Hurfin and E. Totel. A scalable and efficient correlation engine to detect multi-step attacks in distributed systems. In: *2018 IEEE 37th Symposium on Reliable Distributed Systems*. SRDS. 2018.

[5] H. Ren, N. Stakhanova and A. A. Ghorbani. An online adaptive approach to alert correlation. In: *7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. DIMVA. 2010.

[6] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar and V. Venkatakrishnan. Holmes: real-time apt detection through correlation of suspicious information flows. In: *IEEE Symposium on Security and Privacy*. S&P. 2019.

[7] A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. B. Celik, X. Zhang and D. Xu. ATLAS: A sequence-based learning approach for attack investigation. In: *30th USENIX security symposium*. USENIX Security. 2021.

[8] F. Wilkens, F. Ortmann, S. Haas, M. Vallentin and M. Fischer. Multi-stage attack detection via kill chain state machines. In: *Proceedings of the 3rd Workshop on Cyber-Security Arms Race*. 2021.

[9] Y. Zhang, X. Fan, Y. Wang and Z. Xue. Attack grammar: A new approach to modeling and analyzing network attack sequences. In: *2008 Annual Computer Security Applications Conference*. ACSAC. 2008.

[10] S. O. Al-Mamory and H. Zhang. Ids alerts correlation using grammar-based approach. *Journal in computer virology* 5 (2009).

[11] K. Pei, Z. Gu, B. Saltaformaggio, S. Ma, F. Wang, Z. Zhang, L. Si, X. Zhang and D. Xu. Hercule: Attack story reconstruction via community discovery on correlated log graph. In: *Proceedings of the 32nd Annual Conference on Computer Security Applications*. ACSAC. 2016.

[12] F. Liu, Y. Wen, D. Zhang, X. Jiang, X. Xing and D. Meng. Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In: *Proceedings of the 2019 ACM SIGSAC conference on Computer and Communications Security*. CCS. 2019.

[13] M. Du, F. Li, G. Zheng and V. Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In: *Proceedings of the 2017 ACM SIGSAC conference on Computer and Communications Security*. CCS. 2017.

[14] X. Wang, X. Gong, L. Yu and J. Liu. MAAC: Novel alert correlation method to detect multi-step attack. In: *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications*. TrustCom. 2021.

[15] S. Ingale, M. Paraye and D. Ambawade. A survey on methodologies for multi-step attack prediction. In: *Fourth International Conference on Inventive Systems and Control*. ICISC. 2020.

[16] A. Berady, M. Jaume, V. V. T. Tong and G. Guette. PWNJUTSU: A dataset and a semantics-driven approach to retrace attack campaigns. *IEEE Transactions on Network and Service Management* 19.4 (2022).

[17] S. Kilian, V. Viet Triem Tong, J.-F. Lalande, F. Majorczyk, A. Sanchez, N. Talon, P.-V. Besson, H. Orsini, P. Lledo and P.-F. Gimenez. Casino-

Limit: An Offensive Dataset Labeled with MITRE ATT&CK Techniques. In: *Proceedings of the 28th International Symposium on Research in Attacks, Intrusions and Defenses*. RAID '25. 2025.

[18] P. Goldschmidt and D. Chudá. Network intrusion datasets: a survey, limitations, and recommendations. *Computers & Security* (2025).

[19] M. Landauer, F. Skopik and M. Wurzenberger. Introducing a new alert data set for multi-step attack analysis. In: *Proceedings of the 17th Cyber Security Experimentation and Test Workshop*. 2024.