Journal of Universal Computer Science, vol. 30, no. 9 (2024), 1184-1204
submitted: 27/9/2023, accepted: 30/5/2024, appeared: 14/9/2024 CC BY-ND 4.0

A Flexible Multilevel System for Mitre ATT&CK Model-driven Alerts and Events Correlation in Cyberattacks Detection

Javier Muñoz-Calle
(Dpt. of Telematics Engineering, University of Seville, Seville, Spain
https://orcid.org/0000-0001-8146-8438, fjmc@us.es)

Rafael Estepa Alonso
(Dpt. of Telematics Engineering, University of Seville, Seville, Spain
https://orcid.org/0000-0001-8505-1920, rafaestepa@us.es)

Antonio Estepa Alonso
(Dpt. of Telematics Engineering, University of Seville, Seville, Spain
https://orcid.org/0000-0003-1841-3973, aestepa@us.es)

Jesús E. Díaz-Verdejo
(Dpt. of Signal Theory, Telematics and Communications, University of Granada, Granada, Spain
https://orcid.org/0000-0002-8424-9932, jedv@ugr.es)

Elvira Castillo Fernández
(Dpt. of Signal Theory, Telematics and Communications, University of Granada, Granada, Spain
https://orcid.org/0009-0005-0235-5213, elviracastillo@ugr.es)

Germán Madinabeitia
(Dpt. of Telematics Engineering, University of Seville, Seville, Spain
https://orcid.org/0000-0001-6376-4620, german@trajano.us.es)

Abstract: Network monitoring systems can struggle to detect the full sequence of actions in a multi-step cyber attack, frequently resulting in multiple alerts (some of which are false positive (FP)) and missed actions. The challenge of easing the job of security analysts by triggering a single and accurate alert per attack requires developing and evaluating advanced event correlation techniques and models that have the potential to devise relationships between the different observed events/alerts.

This work introduces a flexible architecture designed for hierarchical and iterative correlation of alerts and events. Its key feature is the sequential correlation of operations targeting specific attack episodes or aspects. This architecture utilizes IDS alerts or similar cybersecurity sensors, storing events and alerts in a non-relational database. Modules designed for knowledge creation then query these stored items to generate meta-alerts, also stored in the database. This approach facilitates creating a more refined knowledge that can be built on top of existing one by creating specialized modules. For illustrative purposes, we make a case study where we use this architectural approach to explore the feasibility of monitoring the progress of attacks of increased complexity by increasing the levels of the hyperalerts defined, including a case of a multi-step attack that adheres to the ATT&CK model. Although the mapping between the observations and the model components (i.e., techniques and tactics) is challenging, we could fully monitor the progress of two attacks and up to 5 out of 6 steps of the most complex attack by building up to three specialized modules. Despite some limitations due to the sensors and attack scenarios tested, the results indicate the architecture's potential for enhancing the detection of complex cyber attacks, offering a promising direction for future cybersecurity research.

Keywords: network security monitoring, Intrusion Detection Systems, cyberattacks models, alert correlation, Mitre ATT&CK attack models
Categories: D.4.6
DOI: 10.3897/jucs.131686

1 Introduction

Cyberattacks are becoming more frequent and relevant, making network security a critical issue. Assets defense requires knowledge of the state of the network and systems. To detect incidents, CyberSecurity Officers (CSO) commonly use the so-called Network Security Monitoring systems (NSM) [Ghafir 19], which operate as a decision support system that provides situational awareness. NSMs use heterogeneous information from multiple sources, such as traffic flows, alerts generated by deployed Intrusion Detection Systems (IDS) [Garcia-Teodoro 09], or log traces collected from services of interest [Martins 22].

Ideally, an effective NSM system should enable the Chief Security Officer (CSO) to be alerted of incidents through a single, comprehensive message. This message should include relevant information for the CSO to assess the incident and access related information. However, existing NSM systems, which rely heavily on Intrusion Detection Systems (IDS)[Garcia-Teodoro 09], often produce a large volume of alarms for the same incident, many of which are false positives. Additionally, cyberattacks are typically composed of multiple actions or steps, complicating their detection [Navarro 18]. To address these challenges and accurately and efficiently identify incidents, it is possible to use advanced techniques and correlation models that can process vast amounts of data, aggregating and linking all events or indicators related to the same attack.

The reduction of several alerts to fewer relevant ones has been extensively addressed in the literature [Kotenko 23][Spathoulas 13]. The mainstream approach relies on real-time correlation of alert properties and time relationships [Navarro 18][Khosravi 20]. However, producing just one alert per incident remains an ongoing research area in cybersecurity [Wang 22]. Our approach towards this goal relies on using heterogeneous correlation methods that could facilitate including attack models in a multilevel approach. The basic idea consists of aggregating alerts and events in *hyperalerts* that create basic pieces of knowledge. These *hyperalerts* are then repeatedly correlated with other alerts, events, and previous *hyperalerts* using various methods that create more sophisticated pieces of knowledge. Each *hyperalerts* is assigned a specific level based on its reliance on earlier defined *hyperalerts*. While the idea of multilevel aggregation of alerts is not new [Soleimani 12][Husák 19], the sequence of steps or phases and their scope is usually pre-established by an underlying model or by the kind of attack to detect (e.g., APTs, Advanced Persistent Threats, in [Ghafir 19b] or [Khosravi 20]). In addition, the predetermined model decides the type of correlation technique needed on each step, such as time-based correlation of identical alerts [Spathoulas 13][Haas 19]. However, this model lacks the flexibility to incorporate new techniques or methods. The majority of previous research concentrates solely on the alerts and their connections, while some of the actions involved in a multi-step attack may go unnoticed by the IDS or even be legitimate. Therefore, it is essential to include data from arbitrary sensors and contextual information.

This study suggests a flexible architecture for an NSM that addresses the challenges outlined in [Zuech 15]. The architecture allows for advanced event correlation techniques to be iteratively employed over pre-existing alerts or *hyperalerts*, taking into account contextual information and enabling the incorporation of attack models. Our architecture provides flexibility in three key areas: the use of various correlation methods, iterative aggregation, and the inclusion of generic events such as alerts or packet-flow reports relevant to attack models. Additionally, we carry out a case study that shows how our iterative aggregation method can detect attacks of increasing complexity by increasing the hyperalert level.

In this paper, we extend our previous research [Castillo-Fernández 23] by incorporating new correlation methods to the generic architecture defined in the paper. In particular, we create an extra level of hyperalerts to relate the techniques and tactics outlined in the Mitre ATT&CK model with the knowledge created from IDS alerts and traffic flow reports. We also conduct an exploratory analysis on the feasibility of mapping the network IDS alerts to the techniques and tactics defined in the Mitre model. Ultimately, our goal is to study all the necessary inputs and techniques that would take to generate single event, or (*hyperalert*), that contained comprehensive information for each cyber incident.

The main contributions of this paper can be summarized as follows:

- We suggest an architecture for enabling complex correlation, and define several knowledge pieces with different correlation levels (hyperalerts) geared toward detecting attacks of different complexity.
- We carry out a preliminary study of the feasibility of inferring the techniques and tactics defined in the ATT&CK attack model by using only IDS alerts and a set of hyperalerts.
- We experimentally evaluate the architecture for cyberattack detection with three attacks that show how increasing the hyperalert level enables the detection of more complex attacks. In our last multi-step attack we are able to track the advance of the attack through most stages of the ATT&CK attack model.

The remainder of this article is structured as follows. The motivation for the proposal and some insights on previous work are presented in Section 2. Section 3 presents the proposed architecture for the flexible incorporation of correlation techniques and attack models. In Section 4, the proposed system is tested with traffic traces from a set of techniques and a set of basic correlation modules. Section 5 explores the application of the model to a multi-step attack. Finally, Section 6 concludes the paper and suggests further research.

2 Previous Work and Motivation

IDSs frequently generate excessive alerts, notably FP. To mitigate this, strategies for refining alert processing and data extraction have been proposed [Kotenko 23]. Alert correlation, an active research domain since the inception of IDSs [Valdes 01], aims to minimize False Positives by filtering out irrelevant alerts [Meng 14] by primarily targeting alerts' attributes in real time [Navarro 18].

Cyberattacks frequently involve multiple steps. While some steps may appear legitimate if viewed in isolation, others could be identified as attacks by monitoring sensors.

Consequently, multiple alerts may be triggered as a result of each individual action in the attack. Initial research on alert correlation targeted basic attacks, employing straightforward methods like tree-based techniques [Sahu 15]. Contemporary studies, however, are advancing towards complex correlation strategies, leveraging knowledge of attack sequences [Husák 19] [Soleimani 12] [Zhang 19] and incorporating Big Data/Artificial Intelligence techniques [Zuech 15]. Hidden Markov Models are particularly suited for recognizing multi-stage attacks, given their proficiency in sequential data analysis.

Our approach is based on creating pieces of knowledge from the available sensors by repeatedly aggregating *hyperalerts* of different levels, which could potentially accommodate the techniques and models related to multi-step attacks. The notion of *hyperalerts* and the application of models to relate them is not new [Navarro 18]. However, most of the existing proposals have fixed aggregation levels and models, while few adopt a hierarchical approach [Kaynar 16]. For example, in [Khosravi 20], hosts are ranked based on their likelihood to be exposed to APT attacks. They classify Security Information and Event Management (SIEM) alerts into meta-alerts according to an Intrusion Kill Chain (IKC) stage (reconnaissance, exploitation, operation, data collection and exfiltration) based on the alert type. Then, a causal relationship among attack events on the same host is determined based on temporal and IKC stage order. Similarly, in [Wang 21] the authors use the concept of alert semantics to understand the meaning of sensor alerts (e.g., Snort) and to obtain the attack stage of the host. Alerts are automatically mapped to a specific attack stage (scan, exploit, get-access-privilege, post-attack) based on the alert type. Their approach begins by correlating similar alerts for the same host. Next, they create an alert graph in two stages: first, by correlating alerts generated on the same host, and second, by correlating alerts across different hosts using causal correlation, considering the timestamp and attack stage for each host. A key difference between our work and [Wang 21] is that we include the notion level as a property of the hyperalert and define three levels of hyperalerts (rather than a single hyperalert). Another difference is that we include contextual information such as traffic flows (in addition to the alerts). The work by Bryan et al. [Bryan 20] focuses on minimizing alerts in multi-stage attacks by introducing a new log ontology for normalizing security sensor data and correlating events at the SIEM. They pinpointed crucial information elements from sensors across various domains for each phase of their kill chain. Modifying the SIEM baseline ontology with their approach significantly reduced alerts with minimal forensic value.

There are various attack models besides the Kill Chain [Al-Mohannadi 16]. The Mitres's ATT&CK model [Strom 17] categorizes attacks as a sequence of tactics and techniques, and is becoming more widely used in cybersecurity for threat modeling and cyberattack assessment. Recent works have used attack graph techniques to incorporate the ATT&CK model in the alerts correlation process [Sen 19], [Milakerdi 19]. For example, Wang et at.[Wang 22] proposed a novel method for APT multi-step attack reconstruction in large-scale networks for attack forensics and traceability. In their framework, edge servers initially collect and parse alerts from local sensors using a specific ontology that includes mapping alert types to the ATT&CK model's tactics and techniques. These alerts are then sent to the SOC for analysis, where alerts are correlated using attribute comparison and Word Mover's Distance algorithm. Alerts with clear correlations based on their sources or targets within a specific timeframe are then linked. These alerts are structured into a graph to identify and further minimize alerts by mapping communication relationships. Missing attack steps are identified within this graph using Monte Carlo Tree Search. Finally, the graph showcases multi-step attacks, allowing security experts to explore them by hosts or attack paths. In their study, they classify the alerts from NIDS Zeek to 14 distinct tactics of ATT&CK. However, the feasibility of

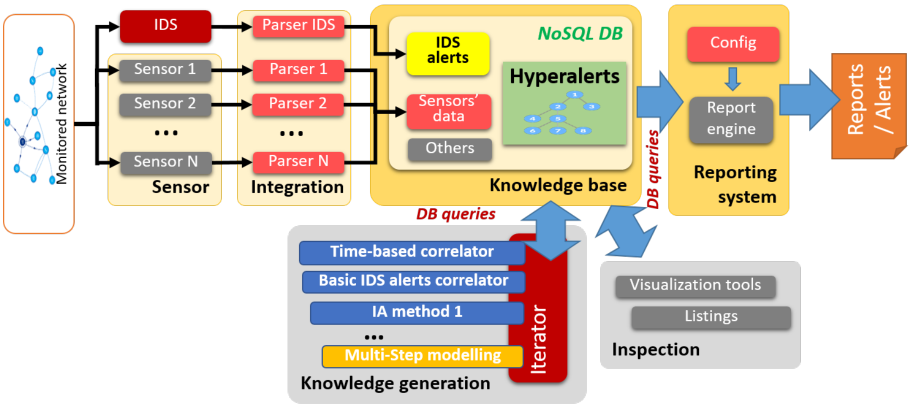

Figure 1: Proposed modular architecture [Castillo-Fernández 23].

such classification in general is not sufficiently addressed.

Our proposal, like [Wang 22], incorporates ATT&CK attack modeling into the correlation process. But rather than pursuing to reduce the number of similar alerts based on attribute similarity, our correlation approach is aimed at getting sufficient knowledge to be able to generate a single alert with the attack progress. Using our generic architectural framework, we define a new hyperalert of level 3 (i.e., integrates the hyperalerts generated at levels 0-2) that informs about the attack state according to the ATT&CK model (see Fig. 6), where states represent tactics, and transitions are based on observed techniques during the attack execution. Our method differs from that of works like [Sen 19] or [Wang 22] that depend on predefined action-based alert/attack graphs, as we emphasize the identification of specific techniques used in each attack phase based on available sensors.

As highlighted in [Castillo-Fernández 23], establishing a correlation between IDS alerts and ATT&CK techniques poses a challenge due to the fact that some techniques are not detectable via network IDS. Rather than taking for granted such association, our research explores the viability of associating sensor observations, predominantly network IDS alerts, with ATT&CK techniques and evaluates the benefits of integrating additional contextual information to enhance this linkage.

3 System Architecture

We aim to develop a flexible and scalable architecture for event correlation in NSM that use sensors data, such as alerts or reports, to create *hyperalerts* that enhance knowledge. As an attack progresses, the system iteratively applies various modules/functions to this data, enriching the knowledge base and elevating the *hyperalert* level. To facilitate this process, it is crucial to have all data types—from primary events and *hyperalerts* to other information—readily accessible. Hence, a database is essential for ensuring uniform and easy access to relevant data at every stage of the attack model.

3.1 Generic Architecture

The proposed generic architecture consists of the following modules (see Fig. 1):

- Sensors: Elements that generate data from network and environmental monitoring. Sensors typically include external elements that generate a variety of data, such as flow analyzers for providing netflow information, flow classifiers for reporting protocol information, and network logs. Most works in the literature, as we do, rely on an network IDS as main source of data.

- Integration: To integrate sensor data into the database, we use specific preprocessor modules for each sensor. These modules condition the data and ensure it has a consistent format in the common fields that are relevant for knowledge generation. The goal is to normalize the data and make it homogeneous.

- Knowledge base: Database containing all system information (sensing data, operative parameters, and hyperalerts). The information from the NSM is stored in a database, which facilitates its access and indexing according to various criteria. As the information to store is heterogeneous, a non-relational database is considered.

- Knowledge generation: It comprises correlation modules and an iterator component that translates the logic implemented on each module to database queries and data manipulation. Each module has a specific purpose and conducts correlation operations through database queries. The *iterator* module applies a sequence of operations through a sequence of queries to the database. Query results may include IDS alerts, packet flows and previous hyperalerts. Each operation result is given an index (ID) that is stored along with the data and enables traceability.

- Inspection: Tools to access the knowledge base. They should be able to select elements and traceback their components/aggregated items. For example, graph tools facilitate the identification of attacks.

Despite its simplicity, this architecture offers high versatility and effectiveness. While we only utilize two sensors and focus on a single knowledge generation module in this study, the architecture is openly designed to integrate new algorithms, correlation techniques, and sensors. Its greatest strength lies in the iterative application of algorithms based on outputs stored in the database and the organization of aggregated information into hyperalerts through a layered structure.

The database will provide input data for each module, and the resulting *hyperalerts* will be inserted back into the database. This process allows for aggregating primary elements, such as sensor data or previous hyperalerts, which creates different *levels* of aggregation and enables an iterative procedure. For example, a correlator module based on a time window could aggregate alerts generated during the exploration of a web service vulnerability by a user with the same IP address. The iterator module would first obtain a list of alerts within the time range, and then the correlator would be applied to each alert, checking for duplicates and grouping together similar alerts with the same victim IP and SID. It is worth noting that every alert or hyperalert generated is assigned and unique identifier (ID), which is used as a reference to be included in the results of future operations.

3.2 Definition of a Basic Set of Basic Hyperalerts Levels

We'll use two sensors in our study: a signature-based IDS as the primary data source for correlation, and a flow generator as a second sensor to provide contextual information from traffic flows.

In this paper, we propose a case of use with the following levels of hyperalerts:

- Level-0 hyperalert. A level-0 hyperalert aggregates all IDS alerts referred to the same victim host and SID over a time window. It includes the victim's IP address, start and end timestamps, the SID from the IDS alert, and the list of IDs (i.e., pointers) of the original alerts during the period under consideration.
- Level-1 hyperalert. A level-1 hyperalert aggregates all previous alerts or level-0 hyperalerts over a given time window that include the same IP as the victim. It includes the IPs and ports of the communication and the list of IDs that identify the corresponding alerts or level-0 hyperalert. A level 1 hyperalert also includes the network flows involving the victim's IP (if available). An incomplete example of a level-1 hyperalert can be found in Figure 4.
- Level-2 hyperalert. A level-2 hyperalert aggregates (by including their IDs) all alerts or hyperalerts (of levels 0 or 1) associated with the victim's IP **or with those hosts that communicated with the victim up to two hops** that happened during a specific time window. It also includes the traffic flows related to all these hosts during such period.

We believe that the previous set of hyperalerts could be useful in the detection of attacks or in forensic analysis. At a simple glance, we could see in the level 0 hyperalerts the triggered alerts, in the level 1 hyperalerts everything related to the victim, and in the level 2 hyperalerts everything that has interacted with anyone who has interacted with the victim. This basic set will also be accounted for in the knowledge detection for the detection of multi-step attacks developed in next Section.

3.3 Implementation Issues

Next, we will discuss some practical implementation aspects of the architecture. We have developed a proof of concept of the proposed architecture using Python. Although the system is designed to be open and incorporate multiple sensors of various kinds in the future, in a first approach, we will only count with the alerts generated by an IDS, along with the traffic packet flows. Deep packet inspection will be utilized to classify flows based on the carried protocol [El-Maghraby 17]. The tools chosen for these tasks are *Snort*[1], a widely used public domain NIDS, and *Tranalyzer*[2], a flow analyzer with multiple functionalities, including flow classification. The parsers in the integration block (Fig. 1) will preprocess and normalize the sensors' output data. Some fields, such as timestamps, IP addresses, etc., will be employed for indexing purposes.

The database has been structured in several collections that correspond directly to the data obtained from the sensors: *alerts* (Fig. 2) and the *flow information* (Fig. 3). Each alert and flow is saved as a document in its corresponding collection. New collections can be generated by each of the aggregator modules, e.g. level 1 hyperalerts (Fig. 4). The database server selected for this implementation was *MongoDB* due to its scalability and interoperability with many of the sensors.

The implemented set of basic aggregator modules that correspond to the levels 0, 1 and 2 hyperalerts described above. Thus, following the previous example, a first module groups the information from identical alerts (same IP and SID but different timestamps) produced during a time window. The result is a level 0 hyperalert that will be stored in the database. The second module creates level 1 hyperalerts by grouping level 0 hyperalerts

[1] https://snort.org
[2] https://tranalyzer.com

```
"_id" : ObjectId("605f7ab7853a36dee68117cb"),
"type" : "event",
"event" : {
        "classification" : "Detection of a Net Scan",
        "sensor-id" : 0,
        "event-id" : 10,
        "event-second" : 1591982026,
        "signature-id" : 1917,
        "source-ip" : "10.6.12.203",
        "destination-ip" : "239.255.255.250"
        ... (some fields ommitted)  }
"_id" : ObjectId("605f7ab7853a36dee68117cc"),
"type" : "packet",
"packet" : {
        "sensor-id" : 0,
        "event-id" : 10,
        "event-second" : 1591982026,
        ... (some fields ommitted)      }  }...
```

Figure 2: Alert register sample (selected fields), including information from the alert and the packet triggering the alert.

```
"_id" : ObjectId("605f7ac253c22c72d429d275"),
"dir" : "A",
"timeFirst" : ISODate("2020-06-12T17:13:23.347Z"),
"timeLast" : ISODate("2020-06-12T17:13:23.347Z"),
"srcMac" : ["00:11:75:68:42:d3"],
"dstMac" : ["98:40:bb:2a:f7:e5"],
"srcIP" : "10.6.12.157",
"srcPort" : 60444,
"dstIP" : "10.6.12.12",
"dstPort" : 389,
"nDPIclass" : "LDAP",
... (Netflow-like fields ommitted) }...
```

Figure 3: Flow register sample (selected fields).

based on the same IP address as the victim, and the traffic flows associated with these alerts and IPs. It should be noted that in this case, contextual information is added by including the flows report generated by Tranalyzer. This will generate an hyperalert including not only the alerts but all the interactions associated to the given IP that have triggered an alert. Next, level 2 hyperalerts are generated from lower-level ones. They will include the IDs of all the alerts and flows related to all the interactions generating alerts of the given IP and the IPs interacting with it up to 2 hops.

The proposed architecture allows defining flexible criteria for grouping the information contained in the database, whether alerts, flows, or hyperalerts of any levels, or additional information from other sensors included. Despite being simple, the modules implemented could help discover lateral movements (other IPs attacked by the same offending IP) or two-level command and control topologies (IPs controlled by IPs controlled by other IPs). The code of our propotype can be found at [framework 23].

4 Defining a Level-3 Hyperalert for Multi-step ATT&CK Attacks

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) model [Strom 17] is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. Each tactic corresponds to a phase in the

```
"_id" : ObjectId("60c8ca7f66d956ebab74bf20"),
"tupla" : {
        "srcIP" : "205.185.125.104",
        "destIP" : "10.6.12.203",
        "srcPort" : 80,
        "destPort" : 49739
},
"nAlerts" : 5,
"alerts" : [
{ "alert" : {
        "_id" : ObjectId("605f7ab7853a36dee68117eb"),
        "event" : {
                "event-id" : 24,
                "event-second" : 1591982119,
                ... } } },
... ],
"flow" : ObjectId("605f7ac353c22c72d429d4c5"),
"classificationProt" : "HTTP",
```

Figure 4: Level 1 hyperalert sample (selected fields).

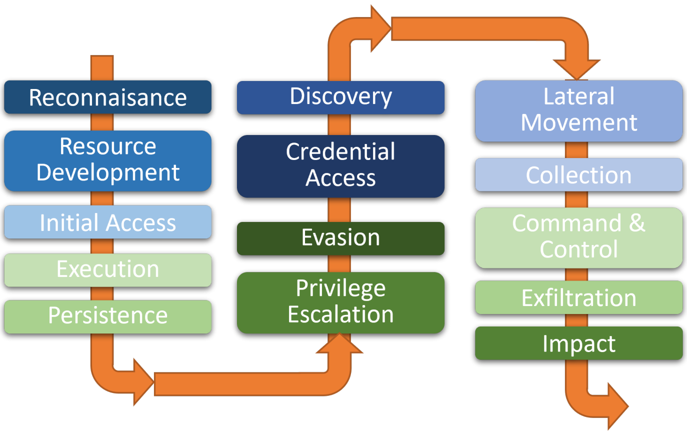

Figure 5: Tactics and it sequenciation in ATT&CK model.

adversary's attack life-cycle (see Fig. 5) and the adversary's technical goals at each phase (i.e., the "why" of a technique). On the other hand, techniques are the "how" in the model, describing the actions adversaries may take to achieve their objectives within a tactic. MITRE ATT&CK is continuously updated to reflect the evolving tactics, techniques, and procedures (TTPs) used by threat actors in real-world campaigns. As such, it is a living model.

Since attack techniques can be related to one or more tactics, an initial approach could be to use a finite state automaton (FSA) model where the tactics were states and the techniques would trigger transitions between states. A illustrative example is shown in Fig. 6.

In the remainder of this Section, we explore the feasibility of creating a new level 3 hyperalert based only on an underlying FSA extracted from the Mitre ATT&CK model and the sensors in place in this work. This new hyperalert would inform about the current tactic in a multi-step attack so the cyber operator can monitor how the attack progresses in different stages.

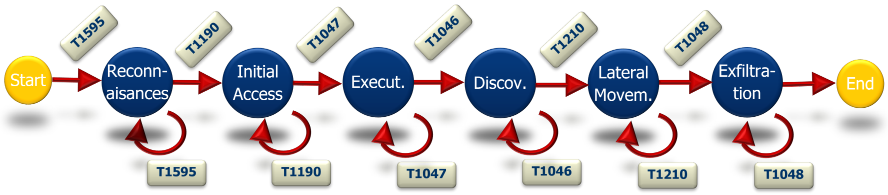

Figure 6: Example model (finite states automaton) for an attack as a function of involved techniques (attack 3).

4.1 On the Design of the Hyperalert

An initial description of this hyperalert would be:

- Level 3 hyperalert. It identifies an attack tactic from the ATT&CK model (see Fig. 5) associated with the alerts or hyperalerts (levels 0, 1, 2) generated during a time window and related to a single victim host.

If possible, a level 3 hyperalert should carry information about the event(s) that triggered a state transition, the information collected from the sensor (i.e. IDS signature ID – SID) and the associated ATT&CK technique. Please note that the FSA model should also had to consider the possibility that some states could not be visited because they have not been implemented in the attack or because they have not been observed with the sensors deployed. Anyhow, level 3 hyperalerts should include the following information:

- Sequence of related Alerts and Hyperalerts over the time window that justify the inference of the technique, including timestamp and involved IPs addresses.
- Network flows related to the hosts referenced in hyperalters level 1 or higher, including its corresponding protocols (using deep packet inspection), timestamp, address tuple, duration and number of packets.
- State reliability level and evidences according to FSA model.

Please note that we assume that an attack technique can be inferred from the information in the lower-level (hyper)alerts, and that such a technique determines the tactic. The following Subsection investigates the extent to which this assumption holds and elaborates on the process of generating level 3 hyperalerts counting exclusively on the alerts generated by the IDS sensor through an experimental study.

4.2 On the Feasibility of Mapping Signature-based IDS Alerts to a Tactic in the ATT&CK Model

To apply an FSA model for tactic detection, it is essential to map techniques to observable events, which in this study are Snort alerts. Although numerous works assume that this mapping is possible (e.g., [Zhang 22, Shawly 20]), we did not find any study that investigates the extent of this claim.

4.2.1 Experimental Study: Techniques Detectability through IDS Alerts

The first question is to what extent attack techniques can be detected exclusively via alerts generated by a signature-based IDS such as Snort. To study this question, we carried out the following tasks:

- We collected the network traffic from 48 instances of attacks, corresponding to 31 different techniques according to the classification by the Mitre ATT&CK framework. Out of these 48 instances of attack, 33 were executed locally, wherein an appropriate scenario was recreated, the attack was carried out, and the corresponding traffic file was captured in pcap format (the pcap files are available in [Data 23]). The remaining 15 instances of attack were obtained from the dataset CIC2018 (CSE-CIC-IDS2018), designed for IDS experimentation. Further details are provided in Appendix A.

- The network IDS Snort (version available in August 2023) has been utilized to analyze the pcap files from each attack. The Talos Community rules (07/27/2023) signatures repository was used (default configuration). All alerts and their respective identifiers (SIDs) were gathered for each attack instance.

- To identify false alarms (FP), we have also included the dataset CIC2018. This public dataset includes legitimate traffic across 17 days and it is designed to detect malicious activity. It includes traffic profiles according to user profiles and widely used protocols such as: HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP. This traffic was then injected into the network IDS. The SIDs in the resulting alarms will be considered false positive.

Table 1 summarizes the main results (more detailed and comprehensive results can be found in Appendix A). It shows for all the possible techniques (*Tech*) associated to each tactic, how many of them that can be detected via network traffic (*NetTec*) according to the Detection DataSource field of the techniques in the Mitre matrix. A first and self-evident result based on Table 1 is that certain tactics, such as Privilege Escalation, cannot be detected by network IDS as they do not leave any trace on the network. We can observe that only 69 out of 227 can be detected through *network traffic*. This means that only 30% of the total number of different techniques leave trace in the network traffic. In our experimental study, we have included attacks instances belonging to 31 out of these 69 techniques, covering 13 out of the 14 possible tactics (excluding only Privilege Escalation)

Column (*Covered*) from Table 1 shows how many of the network-detectable techniques have been included in our study and column (*Instances*) shows the number of attack instances implementing these techniques. The next column (*Ins-Detected*), shows how many of these attack instances were detected by our IDS (i.e., at least one alert). The next column (*Tech-Detected*) shows the number of NetTech for which at least one instance has been detected. Finally, the last column shows the number of different signature identifiers (SIDs) included in the alerts generated by Snort (*#SID*). For the sake of readability, we provide in Appendix 1 (Table 4) more details about the implemented techniques, the tactics they belong to, and the number of different SIDs obtained for each tactic.

The main result of this experiment is that we were only able to detect less than half of the attacks instances (20 out of 48) using the default ruleset configuration. The same holds with the techniques: only 15 out of 31 are detected. If we extrapolated these results to uncovered techniques, we could say that about less than 50% of the attack techniques

Tactic	*Tech*	*NetTech*	*TechCovered*	*Instances*	*Inst-Detected*	*Tech-Detected*	# SID
Reconnaissance	10	3	2	4	4	2	90
Execution	14	3	3	3	1	1	1
Resource Development	8	2	2	2	1	1	12
Initial access	9	4	2	4	2	2	5
Persistence	19	6	2	3	0	0	0
Credential Access	17	5	2	5	1	1	1
Collection	17	2	2	2	1	1	2
Impact	13	5	4	12	5	3	12
Command and Control	16	16	3	3	0	0	0
Defense Evasion	42	9	2	2	0	0	0
Lateral Movement	9	5	2	3	1	1	1
Discovery	31	3	3	4	3	2	330
Exfiltration	9	7	2	3	1	1	3
Privilege Escalation	13	0	0	0	0	0	0
	227	69	31	48	20	15	457

Table 1: Enterprise ATT&CK matrix tactics detection results

that leave trace in network traffic (which are a 30% of the potential techniques) are missed by Snort using the default ruleset. Overall, there were four tactics in which we were not able to detect attacks in our experiment.

Regarding false positives identified within the legitimate traffic dataset, it is worth mentioning that the utilization of Snort, employing an identical set of rules, resulted in the generation of 14,857 alarms, corresponding to 60 different SIDs. Should these SIDs have been filtered out during a signature tuning process aimed at minimizing false positives — a common practice in real-world scenarios [Díaz-Verdejo 22] - only 10 out of the 48 attack instances (constituting 20%) and 6 out of the 31 employed techniques would have been detected.

The interested reader can find detailed results of this experiment in Appendix A (Table 4).

4.2.2 Identifying the Technique through Alerts' SIDs

A second question to analyze is the correlation between techniques and SIDs (i.e., can we identify a technique given an SID?). The results presented in Table 4 show that out of the 457 distinct SIDs, 87% (397) belong to a single technique, while the remaining 13% (60) have been observed in two techniques (with only two instances of a single SID being identified across three different techniques).

There are two techniques with peculiar results: Active Scanning, which caused 46 different SIDs, and Network Service Scanning (with 328 different SIDs associated). For the rest of the detected attacks, on average, we had two different SIDs. Thus, a simple SID would not be enough to detect the technique in such cases.

Therefore, it can be inferred that, to a large extent (but not always) one SID is always related to a single technique.

4.2.3 On the Relationship between Mitre's Techniques and Tactics

We explored the network-traceable techniques in Mitre Enterprise ATT&CK matrix and found that only five techniques were used across multiple tactics: External Remote

Services, BITS Jobs, Traffic Signaling, Pre-OS Boot, and Adversary-in-the-Middle. These techniques were used in various tactics such as Initial Access, Persistence, Defense Evasion, Credential Access, and Collection.

Thus, our results show that 95% of the network traceable techniques are utilized exclusively in a single tactic, indicating a nearly one-to-one correspondence between the techniques and the tactics[3].

4.3 Discussion and Limitations

The experiment carried out has significant limitations. It overlooks techniques that leave no network trace, implementing only slightly less than half of all techniques and relying solely on default Snort signatures. However, we believe that the results obtained are representative enough to conclude that:

1. Mitre's techniques could (to a large extent) determine the transitions between states of the model,
2. SIDs from Snort alarms do not suffice to reliably identify the technique used in attacks. More than half the techniques in our experiment went undetected, and the association between SID and technique is not one-to-one.

The conclusions above suggest that additional context information (such as the number of flows or protocols used) is necessary to complement SIDs information in order to reliably identify attack techniques (specially for scanning attacks). Although our level 3 hyperalert (described above) includes context information (flows), it is necessary further investigation about which sensors would be more suited to complement the information from IDS alerts. Another line of investigation is the tuning of Snort rules.

5 Case Study: Multistep Attacks

To further illustrate the potential of the multilevel approach proposed and the hyperalerts defined above, we have carried out a case study with three real-life multi-step attacks. The first one is a simple attack between an attacker and a victim involving only two nodes. This attack can be identified with a level 1 hyperalert. The second attack involves more than two nodes and its identification requires level 2 hyperalerts. The first two scenarios aim to test L1 and L2 level hyperalerts' ability to aggregate data. The third one is a web-based attack resulting in data exfiltration. This scenario will be analyzed in greater depth and focuses on how ATT&CK modeling can be integrated by identifying techniques through level 3 hyperalerts that include not only the IDS sensor but also information about the traffic flows.

[3] Please note that according to Table 1, 31 techniques belonging to different tactics have been covered, but only 29 different techniques have been implemented. The reason is that one of the implemented techniques (Traffic Signaling) is used in 3 different tactics. The same happens with the total number of techniques that can be detected by network traffic although the overall number is 69 in the table, there are only 62 different techniques.

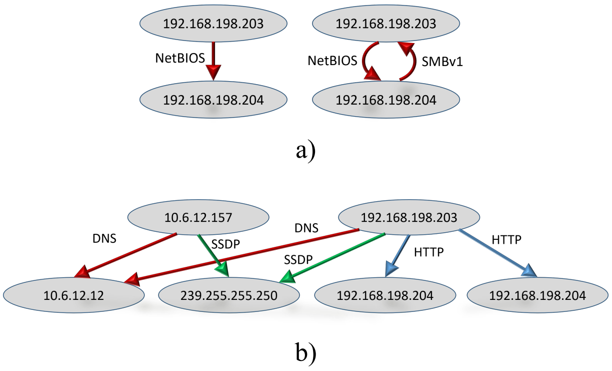

Figure 7: Results (insight) for case study: a) Interaction graph for Eternal Blue Double Pulsar attack, b) Idem for Zloader.

5.1 Implemented Attacks

5.1.1 Attack 1: Remote access using Eternal Blue Double Pulsar

The first attack tested considers a network in which we have installed a backdoor. We have executed *Eternal Blue Double Pulsar* [Sheila 17] attack followed by a DLL injection using *Double Pulsar*. Both steps have been carried out using *metasploit*[4]. This attack includes three tactics: (*initial access, execution* and *persistence*).

In this case, only the attacker IP and the victim IP are involved, so it is expected that all events associated with alerts, flows and victim IP are grouped into a single hyperalert of level 1. The results showed that 5 alerts were triggered by Snort under 2 different SIDs. This resulted in two level 0 hyperalerts (same SID and victim IP) and a single level 1 hyperalert (from 2 LVL0 hyperalerts sharing the same victim IP). During the aggregation of the level 0 hyperalerts, information of 23 flows related to the victim IP was added to the level 1 hyperalert. Fig. 7.a) graphically shows the relationships (IPs, flows and identified protocol) involved in the single level 1 hyperalert found in this case based on these traffic flows.

5.1.2 Attack 2: Zloader Infection

The second attack is executed by replaying a selected capture from the *Traffic Analisys Exercices* collection at *malware-traffic-analysis.net*[5]. We have chosen an infection by *Zloader* malware (exercise for the day 2020-06-12), as it involves various victims (IPs) and several steps.

In this case, the tactics involved are *initial access, execution, defense evasion* and *command and control*. As various IPs are involved, at least level 2 hyperalerts are expected to be needed to merge all the related information.

During the execution of the attack, we obtained 111 IDS alerts and 881 flows involving 6 different IPs. Level 0 analysis grouped all the alerts in 6 hyperalerts (i.e. there were only 6 different SIDs) that, in turn, resulted in 6 level 1 hyperalerts (for six different IPs).

[4] https://www.metasploit.com
[5] https://malware-traffic-analysis.net

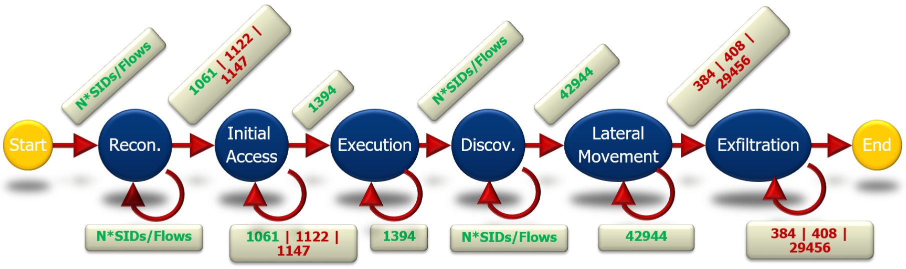

Figure 8: ATT&CK FSA implemented model for attack 3.

Note that level 1 hyperalerts include the information for all the flows in the interactions generating alerts among two nodes. Finally, a single level 2 hyperalert is obtained.

Unlike the previous scenario, this second attack generated level 2 hyperalerts, which allowed the identification of all the IPs involved in the attacks and the nature of their interactions — see Fig. 7.b) —.

5.1.3 Attack 3: Data Exfiltration using Caldera

The third attack is a data exfiltration, a multi-step attack involving multiple tactics. Our goal is to be able to analyze its progress through level 3 hyperalerts. For this, we have considered the ATT&CK tactics executed from *Caldera*, a tool from Mitre, using some of its available attack techniques. This way, we can break down the attack by technique and facilitate the mapping between techniques and tactics to the events observed.

The scenario implemented uses a vulnerable web server as the entry point, and up to 6 different tactics are executed through its corresponding attack technique. As the tactics and techniques are known, setting a FSA to model the attack should be straightforward, but the FSA defined is based on the results from the previous Section. After conducting experiments (refer to Appendix A), we have determined that changes between states will primarily be triggered by the SID of the alerts generated by the IDS. However, for states that involve scanning techniques (such as reconnaissance and discovery), we will rely on context information from the traffic flows instead. In particular, an unexpectedly high number of protocols (e.g., more than 50) or data flows (e.g., more than double than the last observation). The resulting FSA is ilustrated in Fig.8 and includes the conditions found to be effective to trigger state changes. Those in green represent the ones that have been triggered during the experiment.

In Table 2, we can see the amount of events discovered after the execution of each tactic through its respective technique. This includes the total number of alerts (identified by different SIDs) in the column Ndiff and some of the SIDs from the alarms produced. To achieve maximum detection, Snort was used with all Talos rules activated. The table also includes the number of different flows generated and the different protocols seen. Interestingly, two tactics (discovery and exfiltration) did not trigger any alerts, even with the maximum detection settings. However, these tactics did generate associated flows. It can also be observed significant differences between tactics. For example, discovery (T1046) and reconnaissance (T1595) generated a large number of flows and protocols.

Thus, using only the activated SIDs is the simplest solution to identify some techniques and generate the corresponding level 3 hyperalert. In this case, the hyperalert only incorporates these alert identifiers (SIDs) as relevant information. Table 2 shows

	Alerts			Flows	
Tactic	Techn.	Ndif	SIDs	N	Types (protocols)
Reconnaissance	T1595	46	825, 835, 839, 845, 849, 853, ···, 43290	13093	DNS, ··· (+50 types)
Initial Access	T1190	4	1061, 7070, 13990, 19439	156	HTTP
Execution	T1047	1	1394	1736	HTTP, MDNS, IGMP, IGMPv6
Discovery	T1046	0	-	131366	POP, RTSP, ··· (+150 types)
Lateral Moveme.	T1210	2	1394, 42944	4	ICMP
Exfiltration	T1048	0	-	50	HTTP, FTP_CONTROL

Table 2: Alerts and flows for each tactic/technique (attack 3).

the SIDs activated in each technique for the multistage attack considered, as well as the number of flows and their types. Unfortunately, this solution cannot be extended to all states.

Three relevant issues can be observed. First, as already mentioned, some techniques do not trigger IDS alerts. Secondly, depending on the technique, the number of SIDs and alerts is highly variable. As seen in Table 2, some techniques trigger up to 46 SIDs, while others only trigger 1. Thirdly, if the detection process is assessed in depth, we find that, in many cases, a generic detection has been carried out. As an example, the technique *T1190* triggered a detection by *SQL xp_cmdshell attempt*. That is, alerts are activated for accessing the resource *xp_cmdshell*. Therefore, as already shown in Section 4, it is impossible to identify the technique used in all cases solely by the activated SIDs.

In contrast to the previous section's results, no SIDs were found for the T1046 discovery technique, which was detected by our FSA thanks only to the flows. SIDs were also not observed in the exfiltration technique. We have already seen that the detection of this technique depends on the protocol used and, in this case, it has not been detected.

In Table 3, we have summarized the progress of the attack by analyzing the level 3 hyperalerts generated and the results of our experiment. Our findings indicate that we were able to detect five out of the six states (or tactics) of the multi-step attack, which suggests that this approach could be effective in practical scenarios. We also observed that including contextual information from the traffic flows was critical in transitioning between certain states of the model (reconnaissance and discovery), as illustrated in Figure 8. Finally, we were only unable to detect the last state of the attack using level 3 hyperalerts, what illustrate the potential of this proposal on detecting the progression of a multi-step attack. Anyway, once the lateral movement state is reached, it would be possible to search for some specific flow types involving the IPs in the level 3 hyperalert, e.g. HTTP and FTP_CONTROL, which can be associated to a transition to exfiltration state. That is, additional contextual information can be helpfull in the cases in which no alert is triggered.

6 Conclusions and Further Work

We have presented an architecture articulated through the ideas of *hyperalert*, processing of stored information rather than real-time event correlation, and the generation of knowledge via specialized modules. The proposed architecture is applicable to NMS and

ID	**Tactics**	Trigger By Technique	**Identified by**
1	Reconnaissance	T1595	flows
2	Initial Access	T1190	SID 1061
3	Execution	T1047	SID 1394
4	Discovery	T1046	flows
5	Lateral Movem.	T1210	SID 42944

Table 3: Level 3 Hyperalerts.

provides flexibility and potential for incorporating new sensor elements and correlation techniques, including ATT&CK attack modeling.

We have adapted the proposed architecture to accurately identify the advancement of a multistage attack by utilizing a new level of hyperalerts (level 3). We conducted extensive experimentation to distinguish the various stages of an attack, leveraging the ATT&CK model and observable data from NMS systems such as IDS alarms and flow/protocol characteristics.

The case study conducted has shown the effectiveness of this approach, even with the use of basic correlator modules. With the help of a visualization tool developed, the CSO only needed to analyze a level 2 hyperalert to investigate the attacks. Furthermore, the study validated the usefulness of the model for new level 3 hyperalerts.

This work is only the first step from the authors toward developing a system that uses multi-stage attack modeling and additional context information to generate more meaningful and relevant alerts. We plan to extend our architecture in various ways in future research. First, incorporating and evaluating additional correlation techniques that introduce extra intelligence. Second, expanding our exploratory analysis for ATT&CK to new attack instances and new data sources, like Host IDS. The next step will also include an in-depth analysis of ATT&CK mapping to consider different rulesets and more realistic data traffic to refine the False Positive collection. On the other hand, the inclusion of flows in the aggregation processes shows the importance of including contextual information from additional sensors to establish relationships between assets and events.

Acknowledgements

Work supported by Spanish MICIN/AEI/10.13039/501100011033 under Project PID2020-115199RB-I00.

Appendix A

Table 4 shows detailed information on the utilized techniques and subtechniques used in the 48 different attack instances implemented. The table also shows the attack instance source (column SOURCE: can be locally generated or CIC2018 dataset), the number of different signatures found in the alerts (column #SID), number of alerts (colum Alerts) and example of signature identifiers (column SID examples). In order to generate the local attack instances, we have utilized a variety of tools to implement the attack techniques, including nmap, Dirb, GoBuster, wget, Metasploit, Hydra, SQL injection, Sqlmap, Knockd, Arpspoof, Macof, hping3, XMRIG, dnscat2, Caldera, ssh,

Instance	TACTIC	TECHNIQUE	(SUB)TECH ID	#SID	#SID FP	SOURCE	SID examples	Alerts
1	Reconnaissance	Active Scanning	T1595.002	2	2	local	1421,1418	4
2	Reconnaissance	Active Scanning	T1595.003	44	0	local	1434,1433,…	9562
3	Reconnaissance	Active Scanning	T1595.003	43	0	local	1434,1433,…	62
4	Reconnaissance	Search Open Technical Databases	T1596.001	1	0	local	255	1
5	Execution	User Execution	T1204.001	0	0	local	-	0
6	Execution	Windows Management Instrumentation	T1047	1	1	local	1394	18
7	Execution	Exploitation for Client Execution	T1203	0	0	CIC2018	-	0
8	Resource Development	Compromise Accounts	T1586.001	0	0	local	-	0
9	Resource Development	Compromise Infraestructure	T1584.005	12	12	CIC2018	1448,42340,…	4719
10	Initial Access	Exploit Public-Facing Application	T1190	0	0	local	-	0
11	Initial Access	Exploit Public-Facing Application	T1190	3	0	local	1061,1122,…	5
12	Initial Access	Exploit Public-Facing Application	T1190	0	0	CIC2018	-	0
13	Initial Access	Phishing	T1566.001	2	0	CIC2018	1292,46983	3
14	Persistence	Server Software Component	T1505.001	0	0	local	-	0
15	Persistence	Server Software Component	T1505.005	0	0	local	-	0
16	Persistence, C&C,DE	Traffic Signaling	T1205.001	0	0	local	-	0
17	Credential Access	Brute Force	T1110.001	0	0	local	-	0
18	Credential Access	Brute Force	T1110.001	0	0	CIC2018	-	0
19	Credential Access	Brute Force	T1110.001	1	1	CIC2018	650	1
20	Credential Access	Adversary-in-the-Middle	T1557.002	0	0	local	-	0
21	Credential Access	Adversary-in-the-Middle	T1557.002	0	0	local	-	0
22	Collection	Data from Configuration Repository	T1602.001	2	2	local	1411, 1417	792
23	Collection	Data from Information Repositories	T1213	0	0	local	-	0
24	Impact	Data Manipulation	T1565.001	0	0	local	-	0
25	Impact	Endpoint Denial of Service	T1499.02	1	0	local	40063	92
26	Impact	Endpoint Denial of Service	T1499.02	0	0	CIC2018	-	0
27	Impact	Endpoint Denial of Service	T1499.02	0	0	CIC2018	-	0
28	Impact	Endpoint Denial of Service	T1499.03	0	0	CIC2018	-	0
29	Impact	Endpoint Denial of Service	T1499.03	0	0	CIC2018	-	0
30	Impact	Network Denial of Service	T1498.001	1	1	local	1917	8
31	Impact	Network Denial of Service	T1498.001	1	1	local	402	15
32	Impact	Network Denial of Service	T1498.001	1	1	CIC2018	402	10134
33	Impact	Network Denial of Service	T1498.001	0	0	CIC2018	-	0
34	Impact	Network Denial of Service	T1498.001	0	0	CIC2018	-	0
35	Impact	Resource Hijacking	T1496	1	1	local	254	2
36	Command and Control (C&C)	Application Layer Protocol	T1071.004	0	0	local	-	0
37	Command and Control (C&C)	Non-Standard Port	T1571	0	0	local	-	0
38	Defense Evasion (DE)	System Binary Proxy Execution	T1218.010	0	0	local	-	0
39	Lateral Movement	Exploitation of Remote Services	T1210	1	1	local	42944	1
40	Lateral Movement	Exploitation of Remote Services	T1210	0	0	local	-	0
41	Lateral Movement	Exploitation of Remote Services	T1210	0	0	CIC2018	-	0
42	Discovery	Network Service Scanning	T1046	319	1	local	1071,1242,…	1846
43	Discovery	Network Service Scanning	T1046	9	7	local	384,453,…	9
44	Discovery	Remote System Discovery	T1018	2	2	local	1421,1418	2
45	Discovery	Network Service Discovery	T1046	0	0	CIC2018	-	0
46	Exfiltration	Exfiltration Over Alternative Protocol	T1048.003	3	3	local	29456,384,…	48
47	Exfiltration	Exfiltration Over Alternative Protocol	T1048.003	0	0	local	-	0
48	Exfiltration	Exfiltration Over C2 Channel	T1041	0	0	local	-	0

Table 4: Enterprise ATT&CK Matriz attack instances detection results.

Regsvr32.exe, Nikto, and hping3. For further information on the pcap traffic generated during each attack, please refer to [Data 23].

Based on the data, we can state that nearly 50% of the attacks didn't trigger alarms and went undetected by the IDS (when using default rules). However, it is important to keep in mind that the effectiveness of detection varies depending on the technique being used. For example, technique T1048.003 can be detected if exfiltration is over ICMP, but not over FTP.

We have utilized legitimate traffic from the CIC2018 dataset to verify the false positives of the IDS, identifying 60 Snort rules that generate false positives. These correspond to the following SIDs: 1045, 11968, 1257, 1280, 1325, 1390, 1394, 1411, 1413, 1417, 1418, 1419, 1420, 1421, 1444, 1447, 1448, 1616, 1867, 1917, 2003, 2004, 2049, 2339, 2418, 254, 27899, 28555, 28556, 28557, 29456, 31136, 365, 366, 368, 372, 373, 382, 384, 385, 396, 399, 401, 402, 404, 408, 409, 41978, 42255, 42340, 42944, 449,

Tactic	Technique	Technique ID	SubTech.	Tool	#SID	SID	#Alerts
Discov.	Network Service Scanning	T1046	-	Nikto	319	1071, · · · , 43285	1846
Discov.	Network Service Scanning	T1046	-	nmap	9	257, 384, 408, 451, 453, 598, 1418, 1420, 1421	9
Reconn.	Active Scanning	T1595(.002)	Vulnerab. Scanning	nmap	2	1418, 1421	4
Reconn.	Active Scanning	T1595(.003)	Wordlist Scanning	Dirb	44	825, 835, 839, 845, 849, 853, 879, 882, 885, 886, 887, 895, 896, 937, 940, 993, 1016, 1025, 1071, 1129, 1141, 1145, 1201, 1206, 1213, 1218, 1231, 1288, 1301, 1433, 1434, 1489, 1543, 1520, 1551, 1521, 1606, 1662, 1826, 1852, 1877, 2062, 43285, 43290	9562
Reconn.	Active Scanning	T1595(.003)	Wordlist Scanning	GoBuster	43	825, 835, 839, 845, 849, 853, 879, 882, 885, 886, 887, 895, 896, 937, 940, 993, 1016, 1071, 1129, 1141, 1145, 1201, 1206, 1213, 1218, 1231, 1288, 1301, 1433, 1434, 1489, 1520, 1521, 1543, 1551, 1606, 1662, 1826, 1852, 1877, 2062, 43285, 43290	62

Table 5: Details for different scanning techniques implemented instances.

451, 453, 566, 579, 613, 648, 649, 650.

In column # SID FP of Table 4, we include the count of detected SIDs that are among the previously listed SIDs generating false positives. Had these rules been removed, the detection capability would be reduced by half.

It is important to mention that when using scanning techniques (T1595, T1046), a lot of alerts are generated (328 for scanning network services and 46 for active scanning). This can make it challenging to associate these techniques (and the corresponding tactic) with a single SID. Table 5 breaks down the outcomes of the different attack instances used for these two techniques. Nikto, which is used for website scans, generated numerous alerts with various SIDs (319). In contrast, nmap-scanned network services generated only nine alerts that had different SIDs. A similar problem arises during the recognition phase, where there is a noticeable difference in the number of alarms generated between vulnerability searches and website wordlist scans. The former only produced 4 alarms with 2 unique SIDs, whereas the latter created numerous alarms. This indicates that the scan detection process should consider more data beyond just SIDs, such as network flows. More details on implemented attack instances and pcap files used are available in [Data 23].

References

[Al-Mohannadi 16] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen, J. Disso: “Cyber-Attack Modeling Analysis Techniques: An Overview”; Proc. IEEE 4th Int. Conf. on Future Internet of Things and Cloud Workshops (FiCloudW) (2016), 69-76.

[Bryan 20] B.D. Bryant, H. Saiedian: “Improving SIEM alert metadata aggregation with a novel kill-chain based classification model“; Computers & Security, 94 (2020) 101817.

[Castillo-Fernández 23] E. Castillo-Fernández, J. Díaz-Verdejo, R. Estepa Alonso, A. Estepa Alonso, J. Muñoz Calle, G. Mabinabeitia: "Multistep Cyberattacks Detection using a Flexible Multilevel System for Alerts and Events Correlation"; Proc. 2023 European Interdisciplinary Cybersecurity Conference (EICC '23), ACM, New York (2023), 1–6. https://-doi.org/10.1145/3590777.3590778

[Data 23] Available at: *h*ttps://github.com/javmunca/Traffic_Network. Last accessed 29-sep-2023.

[Díaz-Verdejo 22] Díaz-Verdejo J, Muñoz-Calle J, Estepa Alonso A, Estepa Alonso R, Madinabeitia G. "On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks"; Applied Sciences. 2022; 12(2):852. https://doi.org/10.3390/app12020852

[El-Maghraby 17] R. T. El-Maghraby, N. M. Abd Elazim and A. M. Bahaa-Eldin: "A survey on deep packet inspection"; Proc. 2017 12th Int. Conf. on Computer Engineering and Systems (ICCES) (2017), 188-197.

[framework 23] Available at: *h*ttps://github.com/javmunca/Multilevel-Cyberattacks-Detection. Last accessed 14-march-2024.

[Garcia-Teodoro 09] P. García-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, E. Vázquez: "Anomaly-based network intrusion detection: techniques, systems and challenges"; Computers & Security 28 (2009), 18-28.

[Ghafir 19] I. Ghafir, V. Prenosil, J. Svoboda, M. Hammoudeh: "A Survey on Network Security Monitoring Systems"; Proc. 2016 IEEE 4th Int. Conf. on Future Internet of Things and Cloud Workshops (FiCloudW) (2016), 77-82.

[Ghafir 19b] I. Ghafir et al, "Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats"; IEEE Access, 7 (2019), 99508-99520.

[Haas 19] S. Haas, M. Fischer: "On the alert correlation process for the detection of multi-step attacks and a graph-based realization"; ACM SIGAPP Applied Computing Review 19 (2019), 5–19.

[Husák 19] Martin Husák, Jaroslav Kašpar: "AIDA Framework: Real-Time Correlation and Prediction of Intrusion Detection Alerts"; Proc. 14th Int. Conf. on Availability, Reliability and Security (ARES '19), 81 (2019), 1–8.

[Kaynar 16] K. Kaynar, K. 2016: "A taxonomy for attack graph generation and usage in network Security"; Journal of Information Security and Applications, 29 (2016), 27-56.

[Khosravi 20] M. Khosravi, B.T. Ladani: "Alerts correlation and causal analysis for APT based cyber attack detection"; IEEE Access, 8 (2020) 162642-162656.

[Kotenko 23] Igor Kotenko, Diana Levshun St.: " A survey on artificial intelligence techniques for security event correlation: models, challenges, and opportunities"; Artificial Intelligence Review, under review, doi: https://doi.org/10.21203/rs.3.rs-1975426/v1

[Martins 22] I. Martins, J.S. Resende, P.R. Sousa, S. Silva, L. Antunes, J. Gama: "Host-based IDS: A review and open issues of an anomaly detection system in IoT"; Future Generation Computer Systems 133 (2022): 95-113.

[Meng 14] Yuxin Meng, Lam-For Kwok: "Adaptive non-critical alarm reduction using hash-based contextual signatures in intrusion detection"; Computer Communications, 38 (2014), 50-59.

[Milakerdi 19] Milakerdi, S., et al.: "HOLMES: Real-time APT Detection through Correlation of Susipicious Information Flows"; Proc. 2019 IEEE Symp. on Security & Privacy, (2019) 1137-1152.

[Navarro 18] J. Navarro, A. Deruyver, P. Parrend: "A systematic survey on multi-step attack detection"; Computers & Security 76 (2018), 214-249.

[Sahu 15] S. Sahu, B.M. Mehtre: "Network intrusion detection system using J48 Decision Tree"; Proc. 2015 Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI) (2015), 2023-2026.

[Salah 13] S. Salah, G. Maciá-Fernández, J. E. Díaz-Verdejo: "A model-based survey of alert correlation techniques"; Computer Networks. 57 (2013) 1289-1317.

[Sen 19] Sen, O, et al.: "On using Contextual Correlation to detect Multi-stage Cyber Attacks in Smart Grids"; Sustainable Energy, Grids and Networks, 32 (2022) 100821.

[Shawly 20] T. Shawly, M. Khayat, A. Elghariani, A. Ghafoor: "Evaluation of hmm-based network intrusion detection system for multiple multi-stage attacks"; IEEE Network, 34(3) (2020), 240-248.

[Sheila 17] Sheila A. Berta: "How to Exploit Eternalblue & Doublepulsar to Get an Empire/meterpreter Session on Windows 7/2008"; Eleven Path, Tech. Report (2017).

[Soleimani 12] M. Soleimani, A. Ghorbani: "Multi-layer episode filtering for the multi-step attack detection"; Computer Communications 35 (2012) 1368–1379.

[Spathoulas 13] G. Spathoulas, S. Katsikas: "Enhancing IDS performance through comprehensive alert post-processing"; Computers & Security. 37 (2013) 176-196.

[Strom 17] B. E. Strom et al.: "Finding cyber threats with ATT&CK-based analytics"; The MITRE Corporation, Technical Report No. MTR170202 (2017).

[Valdes 01] A. Valdes, K. Skinner, K.: "Probabilistic Alert Correlation"; Proc. Recent Advances in Intrusion Detection (RAID 2001) (2001) 2212.

[Wang 21] X. Wang, X. Gong, L. Yu, J. Liu: "MAAC: Novel alert correlation method to detect multi-step attack"; In 2021 IEEE 20th Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom) (2021) pp. 726-733. IEEE.

[Wang 22] Wang, Y, Guo, Y., Fang, C.: "An end-to-end method for advanced persistent threats reconstruction in large-scale networks based on alert and log correlation"; Journal of Information Security and Apps., 71 (2022) 103373.

[Zhang 19] K. Zhang, F. Zhao, S. Luo, Y. Xin, H. Zhu: "An Intrusion Action-Based IDS Alert Correlation Analysis and Prediction Framework"; IEEE Access 7 (2019) 150540-150551.

[Zhang 22] X. Zhang, T. Wu, Q. Zheng, L. Zhai, H. Hu, W. Yin, C. Cheng: "Multi-step attack detection based on pre-trained hidden Markov models"; Sensors, 22(8) (2022), 2874.

[Zuech 15] R. Zuech, R., T. Khoshgoftaar, T., R. Wald: "Intrusion detection and big heterogeneous data: a survey"; Journal of Big Data 2 (2015).