The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
7-9 September, 2023, Dortmund, Germany

APT Detection: an Incremental Correlation Approach

Salva Daneshgadeh Çakmakçı[1] , Georgios Gkoktsis [2] , Robin Buchta[3] ,
Kai Olver Detken Kleiner[1] , Felix Heine[3] , Carsten Kleiner[3]
[1] DECOIT GmbH & Co. KG, Bremen, Germany,
daneshgadeh@decoit.de, deltkan@decoit.de, www.decoit.de
[2] Fraunhofer SIT — ATHENE, Rheinstraße 95, 64295 Darmstadt, Germany,
george.gkoktsis@sit.fraunhofer.de, sit.fraunhofer.de
[3] Hochschule Hannover University of Applied Sciences and Arts, Hanover Germany,
{firstname.lastname}@hs-hannover.de

***Abstract* — Advanced Persistent Threats (APTs) are a growing and increasingly prevalent threat. Current detection systems focus primarily on individual procedures and create alerts on this foundation. To effectively detect APT attacks, which rarely consist of single activities, individual alerts must be correlated to comprehensively encapsulate APT activity and provide better situational awareness to the operators. We use this to initiate targeted and proactive countermeasures and thus improve overall security. This paper presents a correlation engine that uses alarms from standard rule-based systems and correlates them with each other. We evaluate the proposed solution using an APT scenario as an example and discuss the advantages and disadvantages of this approach. We argue that the fast, simple implementation, which is an add-on to SIEM, must be considered when evaluating the limited informative value of rule-based systems in the face of zero-day exploits or even sophisticated living-off-the-land attacks.**

Keywords* — *Advanced Persistent Threat (APT), Rule-based System, SIEM, Correlation Engine, APT Detection, Cyberattack Detection

I. Introduction

Advanced Persistent Threats (APTs) [1] are a growing concern for organizations of all sizes, as they pose a significant risk to their operations and reputation. The motivation of an attacker can be difficult to interpret, and organizations cannot rely on the assumption that they will not be targeted. Hence, all organizations must do their best to protect themselves from potential threats to ensure the continuity of their operations.

However, small and medium-sized enterprises (SMEs) face unique challenges in achieving effective cyber defense, as they may lack the necessary resources and the management motivation to invest in such solutions. Managed service solutions offer a practical solution to this issue by utilizing commercial off-the-shelf (COTS) data in a generalizable manner, making the cost of protection more predictable and manageable. This is also the area of existing solutions, such as Security Information Event Management (SIEM) systems, which specialize in traditional attacks.

The current state of the APT, as drawn from recent threat reports from the cybersecurity community [2]–[5], shows an evolution of the cybercrime ecosystem towards an industrialized space. This means that an end-to-end cyberattack may not originate from the same threat actor, but rather from multiple, as is the example of Ransomware-as-a-service. Specialized groups, such as initial access brokers [6] sell their services online, or even offer subscription based services. As such, detecting APT activity transcends stopping a single adversary poised on inflicting damage.

Rule-based systems, in particular, are well suited for protection against APTs because they extract indicators of compromise (IoC) from previously generated cyber threat intelligence, enabling a more targeted defense against specific attacks that are often a subset of APT end-to-end malicious activity. Unlike more advanced methods, such as artificial intelligence-based solutions, rule-based systems mostly do not rely on the normal behavior of a target area as a basis for threat detection. Despite their general volatility, their field of applicability is wide and cross organizational, since there is no retraining requirement, as is the case with AI-based methods. Furthermore, AI-based methods are in the developmental stage concerning APT detection and are not yet suitable for operational use. Only recently, in March 2022, an AI system using fuzzy hashing and deep learning was able to detect a neverbefore seen malware [7], a first, but important achievement towards AI powered cyberdefense.

Given that well-equipped APTs often operate using known tactics, techniques, and procedures (TTPs) and may utilize known patterns, rule-based systems have the potential to detect APT attackers alongside more traditional attackers effectively. The limitations of these methods will be discussed in detail later in this paper. However,

Identify applicable sponsor/s here. If no sponsors, delete this footnote.

the advantage is a comprehensive defense against known attack patterns, with a manageable level of administration that can be outsourced through managed services.

The approach we propose is based on alarm correlation. Alarms are generated from different rule-based systems and correlated with each other via a correlation engine so that serious incidents can be considered in a prioritized manner.

The research questions for this study are:

- How can existing rule-based systems be adapted to the requirements of APT attacks by correlating events?
- What is the trade-off of using rule-based systems in terms of APT detection?

The main contribution of the work is the investigation of existing rule-based methods in terms of the necessary adaptations to meet the increased requirements for APT attack detection. The solution approach, the hierarchical correlation of different rule-based detection engines, is evaluated practically, using an exemplary APT scenario. Furthermore, we discuss the advantages and disadvantages of the implementation and give an outlook for the further development in view of the new threat situation.

The remaining structure of the paper is as follows: Section II presents the related work in this line of research and argues our differentiating novelty. Section III details our methodological approach in detail, the technology and infrastructure that is used, the analytical methods, the threat model, and the exemplary attack scenario. Section IV presents the selected ruleset and attack platform and finaly, Section V contains our concluding remarks.

II. Related Work

There are different approaches in the field of APT attack detection, which we briefly present here. On one hand, there are approaches that try to pin APT attacks down to a specific point of the attack. Exemplary works for this approach are references [8]–[10]. [8] tries to detect APTs at the DNS level, [9] with malware detection and [10] by identifying command and control communication. These approaches do not cover the versatility of an APT and reduce a complex problem to one step, such as the initial access, which does not always have to be identifiable.

Another category that inherently maps correlations in the data structure are graph-based approaches, which can have different characteristics. For instance, Sleuth [11] uses a rule set to map the normal behavior of the system from provenance graphs to graph data and detects deviations accordingly, or threaTrace [12], which detects anomalies through Graph Neural Networks. In addition to provenance graphs, knowledge graphs can also be used to associate assets with events in the system. Here, the work [13] is an example, which builds an entire knowledge graph, and then uses rules to view this entire abundance of knowledge to identify APT attacks.

Methodologically closer to our approach are hierarchically arranged rule systems, such as [14] and [15]. Both systems use a hierarchical concept of rules to bring insights from log information to a cyber-kill-chain representation.

In addition to hierarchical control systems, there is another class of systems that correlate existing alarms. These include, among others, [16]–[18]. NoDoze [16] aims to reduce false positive alarms by generating dependency graphs from existing alarms. However, this approach is based on graph data to generate the dependency graphs afterwards, while we create correlations based on alerts only, with no intermediate steps. Reference [17] works on the alert correlation of IP addresses. The authors use a multi-step Markov property process. They use only the IP address to create correlations.

The closest to our work that we could identify is [18], where an approach similar to ours with different detection modules and a subsequent correlation of the individual events is taken. They use a Hidden Markov Model, whereas we rely on a simpler form of feature correlation. Additionally, our method does not require training the model on normal behavior data. Another key difference is that we do not solely consider network data, but the APT attack holistically, i.e., host data as well.

Overall, we differentiate ourselves from related work in that we focus on the requirements of small and medium-sized enterprises and try to address the high requirements of APT attack detection as best as possible. Therefore, we build on existing infrastructure as much as possible, i.e., by adding an extension to a SIEM system. The overarching goal is to make correlations simple and customizable, so as to increase the cost to those APTs on the lower spectrum of resources and capabilities, who may lack the agility to rapidly change their TTPs frequently.

In this research, we try to overcome some limitations of pure rule-based systems by means of an incremental model which can be easily integrated with state-of-the-art SIEM systems in the market. It enables real-time analysis of system and network levels alerts. The correlation engine correlates APT steps based on high-level tactical tags in addition to technique tags from MITRE ATT&CK to support the generalizability of the approach because it is almost impossible to enumerate all APT scenarios in technique level granularity. We employ a SIEM which facilitates the contextualization of alerts because SIEMs have already information about victim assets and users. We can define our own rules (simple, statistical, anomaly detection) to create low-level alerts. Most prominently, our approach does not require a time window threshold for the correlation of alerts. Instead, alerts with similar identifiers (e.g., hostname, IP address, port numbers, user info) can be even correlated years later.

III. Methodology

We developed a detection method and then established a virtual environment to test our proposed approach. The kill-chain model was adapted to understand the characteristics and behaviors of an attack and design a detection framework. Operating system audit logs and network traffic were collected and used by the detection framework. Three levels of abstractions were defined for the creation of rules as follows:

1) Zeek is used on the top of SIEM to monitor live traffic and signature-based anomaly detection.
2) Detection rules for matching an event (network traffic or endpoint event) with pre-defined conditions.
3) Correlation rules for correlating low-level rules which may hold the potential to evolve into an APT alarm.

A. Framework

This section includes details of our proposed approach. Figure 1 demonstrates the high-level architectural design of the employed SIEM for the purpose of APT detection.

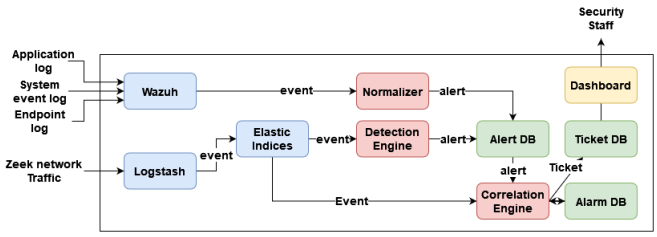

Figure 1. High-Level Architectural Design of the SIEM

Our proposed framework is embedded in a research-based SIEM. The Elastic stack (Elasticsearch, Logstash, Kibana, ELK) [19] is the backend of our SIEM for raw data storage, Visualization and parsing. The detection engine is a component for the execution of detection rules and the creation of alerts. Low-level alerts are stored in the Alert DB, a database for that specific purpose. The data provider of detection rules is Zeek® [20] which is an open-source software network analysis framework. Wazuh [21] is an open-source and enterprise-ready security monitoring solution for servers or devices, which can receive logs in text, windows event logs and Syslog formats. It includes around 2000 rules which are enriched to contain MITRE Technique IDs. The normalizer is a component for converting and matching alerts from Wazuh to the Elastic Common Schema (ECS) format [22]. Finally, the correlation engine correlates alerts utilizing correlation rules and storing them in an Alarm DB. A secondary responsibility of the Correlation Engine is to generate Tickets when the incremental risk of correlated alerts breaches a threshold level, marking them as suspicious alarms.

The technological toolstack is illustrated in Figure 2 for better clarity.

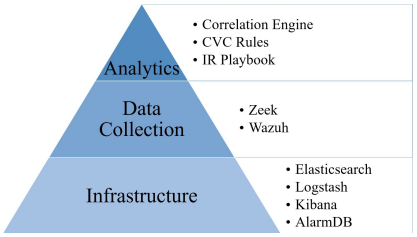

Figure 2. Technology tool stack of propsed architecture

B. Rules

Each rule consists of condition(s), a schedule, a risk score and tag(s). When conditions are met, alerts will be created. A Rule condition always consists of a query (simple or aggregation). For aggregation queries (e.g., number of reached ports based on distinct source and destination IP addresses), a threshold value also can be set. Therefore, a rule first executes the query and then compares the result to the threshold value. A schedule defines a time interval between each run of a rule. There are four different levels of tags in our system as follows:

1) MITRE ATT&CK tactics and techniques tags: Table I demonstrates seven potential steps of APT with their corresponding MITRE tactic identification numbers.
2) Protocol and application tag: It defines an application or a protocol which is used as attack vector.
3) Asset identifier tag: It provides information about the asset (e.g., IP address, host name) which technical resilience measures should be applied.
4) Affected asset tag: It provides asset information to the correlation.

C. Alert

An alert is a JSON object which includes information about the rule conditions, which caused the match and triggered an alert such as source/destination IP/Port number, the total number of sent/received bytes or cardinally of a data attribute. An alert also keeps information about timestamp, risk scores and associated tags.

D. Correlation of Alerts

The correlation engine correlates alerts with the same identifier tag on a regular basis (it is adjustable). For example, if there are two alerts with the same IP address *(source/destination)* and different MITRE ATT&CK tags, the correlation engine combines those alerts together and creates an alarm with an updated risk score. Finally, the Correlation Engine checks the risk score of every alarm in the last T minutes. If the corresponding risk score of an alarm in the Alarm DB exceeds a threshold value, a Ticket will be created for it. Subsequently, the system notifies users (e.g., security staff) about the Ticket. APT detection rules are specific rules that check if there is a combination of MITRE tags in an alarm comparable to the

APT kill-chain. For example, an alarm, and its associated ticket, will be generated for a host with both pass-the-hash (T1550.002) and domain account discovery (T1087.002) alerts.

E. Ticket

A ticket contains the summary of an incident including an alarm, the entire timeline of the correlated alerts, corresponding timestamp of ticket, the risk score of the incident, APT attack likelihood score and relative response playbooks.

F. Playbook

A playbook maintains a set of predefined, technical and/or organizational, tasks which should be performed by security staff to respond to, withstand and recover from a cyber incident. Playbooks are defined in MITRE tactic, technique and sub-technique levels. We defined an extra tag to enable a finer granularity level for some MITRE sub-techniques which address general applications or protocols such as Exploit Public-Facing Application (T1190). Here our application tag determines whether, for instance, SSH-Initial-Access, SMB-Initial-Access, SNMP-Initial-Access or SQL-Initial-Access playbook should be attached to the Ticket. The details of playbooks for various incidents are beyond the scope of this paper.

G. Threat Model and Attack Scenario

We model our adversary as an APT attacker, who also exhibits living-of-the-land (LOL) behavior. Table I summarizes the attack stages that we consider:

Table I. ATTACK STAGES OF THE END-TO-END ATTACK

Attack Stage	***Description***
Initial Access (tag: TA0001)	The attacker gains an initial foothold in the target network.
Reconnaissance (tag: TA0043)	The attacker explores their target environment and identifies assets, network topology and system information from OSINT sources before initial access.
Command and Control (tag: TA0011)	A communication channel between the compromised infrastructure and the attacker's infrastructure is established.
Discovery (tag: TA0007)	Once inside the target environment, the attacker attempts to explore and identify further targets, vulnerabilities, and valuable assets.
Privilege Escalation (tag: TA0004)	The attacker attempts to elevate their access privileges by gaining root/administrator privileges, or other.
Lateral Movement (tag: TA0008)	The attacker moves inside the target network. Both north-south and east-west movement is considered.
Exfiltration and Impact (tag:TA0043/TA0040)	The attacker proceeds to accomplish their objective by either exfiltrating the gathered information, or producing a malicious effect.

Table I also guides the process of scenario creation. We defined two requirements to safeguard the rigor of this work as follows:

1) The attack scenario should be a realistic representation of actual APT behavior, which encapsulates the entirety of the stage space in some form and includes LOL tactics.
2) The attack scenario should be fully testable with the MITRE CALDERA framework and its publicly available plug-ins.

After analysing multiple AttackIQ [23] posts on realistic adversary emulation of actual past use cases, we constructed an attack scenario against a hypothetical company with the following steps:

1) A spearfishing campaign is launched targeting specific users of the company, who had been previously profiled by the attacker by crawling through social media. A malicious attachment is included in the form of an MS Word file. As a result, an initial foothold is established.
2) The deployed payload checks in with a pre-existing C2 infrastructure and maintains a constant encrypted channel, using HTTPS.
3) Once checked in, the attacker manually enters LOL commands to explore the blue infrastructure, map the network and discover any useful data for exfiltration.
4) If the firewall separating the protected network segment is found, the attacker changes their strategy and Masquerades as a domain host account.
5) If the LOL commands do not reveal the protected segment and the firewall is not identified, the attacker scans the network with nmap.
6) At this stage, the Discovery phase is completed and the attacker attempts to move laterally to the protected segment. They use a variety of techniques to achieve that; in particular exploiting the SMB or WMI to deliver the backdoor to those hosts, or trying to directly connect with them with RDP or SSH.
7) Once access to the DC is achieved, they search for valuable data for exfiltration, alter the DC Access policies to either maintain persistence or disrupt the operation of the Active Directory, or decide to ultimately wipe the discs.
8) Once access to the objective target is achieved, they either decide to wipe the discs to rend the host unusable, or discover the databases and manually change their entries.

IV. EXPERIMENTAL METHODOLOGY

This section includes rules to detect different steps of attack scenario presented in section III-G and the correlation rules which were used to create a ticket with an APT likelihood score.

a) Drive-by Compromise (T1189):

- Detection of DNS rebinding-a malicious webpage utilize XSS vulnerability of user's browser to ma-

https://controlcompass.github.io/

nipulate resolution of domain names and access an internal service behind DMZ.
- Detection of a host which frequently makes DNS requests to suspicious dynamic domains.
- Detection of software updates from suspicious locations.

b) Drive-by Compromise (T1189):

- Detection of DNS rebinding-a malicious webpage utilize XSS vulnerability of user's browser to manipulate resolution of domain names and access an internal service behind DMZ.
- Detection of a host which frequently makes DNS requests to suspicious dynamic domains.
- Detection of software updates from suspicious locations.

c) Exploit Public-Facing Application (T1190):

- Detection of Remote Procedure Call (RPC) from/to Internet.
- Detection of Remote Desktop Protocol (RDP) from/to Internet.
- Detection of SMB (Windows File Sharing) activity to Internet.
- Detection of Virtual Network Computing (VNC) from Internet.
- Detection of Telnet port activity.
- Detection of Zoombombing-starting a meeting without a passcode-.
- Detection of suspicious commands executed via a web server.
- Detection of the Proxyshell attack on Microsoft Exchange server to access server side or internal network services behind DMZ.
- Detection of web servers that spawn shell processes (e.g., '\cmd.exe', '\nslookup.exe').

d) External Remote Services (T1133):

- Detection of unexpected child processes of dns.exe (is the process of Windows DNS server service).
- Detection of SSL-VPN connection which the URL field of the related event matches the known IoC for Fortigate-SSL-VPN vulnerabilities.
- Detection of a running Chrome VPN extension that registers to a malicious VPN repository (e.g., gkojfkhlekighikafcpjkiklfbnlmeio, Hola Free VPN, under the folder Software\Wow6432Node\Google\Chrome\Extensions.

e) Hardware Additions (T1200):

- Detection of a rogue DHCP server in internal network.
- Detection of ARP Poisoning
- Detection of a port mirroring activity on Cisco network devices.
- Detection of a USB plugin in Windows.

https://www.fortiguard.com/psirt/FG-IR-18-384

f) Phishing (T1566):

- Detection of an email with known suspicious sender IP address/domain name.
- Detection of an email attachment with known malicious hash.
- Detection of 'outlook.exe'process which starts a suspiciois child process (e.g.,netsh.ex, ipconfig.exe).
- Detection of a HTML file which was opened with a browser process within 5 minutes after downloading.
- Detection of a successful commands and scripts execution by AWS systems manager.
- Detection of a Windows scripting process (cscript.exe or wscript.exe) that executes a PowerShell script.
- Detection of 'outlook.exe'process which starts asuspiciois child process (e.g.,PowerShell.exe, cscript.exe).

g) Valid Account (T1078):

- Detection of excessive number of failed login attempts from public IP.
- Detection of a login attempt to a disabled account.
- Detection of a successful login from unusual countries-defined by a list-.
- Detection of a successful login at unusual time.
- Detection of a user login from various locations in short period of time.
- Detection of windows machine password reset by PowerShell on a remote computer.
- Detection of unusual user login.

h) Trusted Relationship (T1199):

- It requires rules to check activities of second or third-party external providers who is granted the elevated access.

i) Replication Through Removable Media (T1091):

- Detection of external disk drive or USB usage.

Considering attack simulation for the experiments, we opt for the MITRE CALDERA framework [24], as it fulfills the following requirements: i) it is a red teaming emulator, which is a sufficiently granular approximation of real APT activity, ii) it offers a transparent and repeatable experimentation methodology, as only the publicly available abilities are considered, and iii) it is open source and aligns well with our adopted attack pattern model, the MITRE ATT&CK model.

V. Conclusions

This paper presents a rule-based APT detection scheme, which correlates atomic intrusion detection alerts to form a high level APT intrusion alarm. Our detection infrastructure consists of the ELK stack as basis for data storage, processing and visualization, Zeek and Wazuh for host and network source data collection and an event correlation engine. The Control Validation Compass tool provided the basis for our detection ruleset. Then, informed

by the current state of the APT landscape, we constructed an attack scenario, testable with the public abilities of the MITRE CALDERA framework and developed a simple network testbed on Proxmox.

ACKNOWLEDGMENT

REFERENCES

[1] J. T. F. T. Initiative *et al.*, *SP 800-39. managing information security risk: Organization, mission, and information system view*. National Institute of Standards & Technology, 2011.

[2] S. X-Ops, *Sophos 2023 Threat Report Maturing criminal marketplaces present new challenges to defenders*. Sophos, 2022.

[3] Microsoft, *Microsoft Digital Defense Report 2022*. Microsoft Corp., 2022.

[4] Dragos, *ICS/OT CYBERSECURITY YEAR IN REVIEW 2021*. Dragos Inc., 2022.

[5] T. Micro, *Defending the Expanding Attack Surface Trend Micro 2022 Midyear Cybersecurity Report*. Trend Micro Research, 2022.

[6] B. S. Vlad Stolyarov. (2022) Exposing initial access broker with ties to conti. [Online]. Available: https://blog.google/threat-analysis-group/exposing-initial-access-broker-ties-conti/

[7] E. Horvitz. (2022) Artificial intelligence and cybersecurity: Rising challenges and promising directions. [Online]. Available: https://blogs.microsoft.com/on-the-issues/2022/05/03/artificial-intelligence-department-of-defense-cyber-missions/

[8] M. S. Manggalanny and K. Ramli, "Combination of DNS traffic analysis: A design to enhance APT detection," in *2017 3rd International Conference on Science and Technology - Computer (ICST)*, 2017, pp. 171–175. [Online]. Available: https://doi.org/10.1109/ICSTC.2017.8011873

[9] G. Laurenza, R. Lazzeretti, and L. Mazzotti, "Malware Triage for Early Identification of Advanced Persistent Threat Activities," *Digital Threats*, vol. 1, no. 3, aug 2020. [Online]. Available: https://doi.org/10.1145/3386581

[10] A. Alageel and S. Maffeis, "Hawk-Eye: Holistic Detection of APT Command and Control Domains," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, ser. SAC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1664–1673. [Online]. Available: https://doi.org/10.1145/3412841.3442040

[11] M. N. Hossain, S. M. Milajerdi, J. Wang, B. Eshete, R. Gjomemo, R. Sekar, S. D. Stoller, and V. N. Venkatakrishnan, "SLEUTH: Real-Time Attack Scenario Reconstruction from COTS Audit Data," in *Proceedings of the 26th USENIX Conference on Security Symposium*, ser. SEC'17. USA: USENIX Association, 2017, p. 487–504. [Online]. Available: https://doi.org/10.48550/arXiv.1801.02062

[12] S. Wang, Z. Wang, T. Zhou, H. Sun, X. Yin, D. Han, H. Zhang, X. Shi, and J. Yang, "THREATRACE: Detecting and Tracing Host-Based Threats in Node Level Through Provenance Graph Learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3972–3987, 2022. [Online]. Available: https://doi.org/10.1109/10.1109/TIFS.2022.3208815

[13] K. Kurniawan, A. Ekelhart, E. Kiesling, G. Quirchmayr, and A. M. Tjoa, "KRYSTAL: Knowledge Graph-Based Framework for Tactical Attack Discovery in Audit Data," *Comput. Secur.*, vol. 121, no. C, oct 2022. [Online]. Available: https://doi.org/10.1016/j.cose.2022.102828

[14] S. Wen, N. He, and H. Yan, "Detecting and Predicting APT Based on the Study of Cyber Kill Chain with Hierarchical Knowledge Reasoning," in *Proceedings of the 2017 VI International Conference on Network, Communication and Computing*, ser. ICNCC 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 115–119. [Online]. Available: https://doi.org/10.1145/3171592.3171641

[15] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, "HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1137–1152. [Online]. Available: https://doi.org/10.1109/SP.2019.00026

[16] W. U. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, and A. Bates, "NoDoze: Combatting Threat Alert Fatigue with Automated Provenance Triage," in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/nodoze-combatting-threat-alert-fatigue-with-automated-provenance-triage/

[17] F. Xuewei, W. Dongxia, H. Minhuan, and S. Xiaoxia, "An Approach of Discovering Causal Knowledge for Alert Correlating Based on Data Mining," in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, 2014, pp. 57–62. [Online]. Available: https://doi.org/10.1109/DASC.2014.19

[18] I. Ghafir, K. G. Kyriakopoulos, S. Lambotharan, F. J. Aparicio-Navarro, B. Assadhan, H. Binsalleeh, and D. M. Diab, "Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats," *IEEE Access*, vol. 7, pp. 99 508–99 520, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2930200

[19] Elastic, "Elastic stack: Meet the search platform that helps you search, solve, and succeed," 2023. [Online]. Available: https://www.elastic.co/elastic-stack/

[20] Zeek®, "Zeek: An open source network security monitoring tool," 2023. [Online]. Available: https://zeek.org/

[21] Wazuh, "Wazuh: The open source security platforml," 2023. [Online]. Available: https://wazuh.com/

[22] Elastic, "Elastic field reference," 2023. [Online]. Available: https://www.elastic.co/guide/en/ecs/8.6/ecs-field-reference.html

[23] (2022) Attackiq. [Online]. Available: https://www.attackiq.com/

[24] A. Applebaum, D. Miller, B. Strom, C. Korban, and R. Wolf, "Intelligent, automated red team emulation," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016, pp. 363–373.