Evaluating the Robustness of Attack Signatures for Alert Correlation

Stefanie Merz[1,2(✉)], Michael Heigl[2], Abhishek Subedi[2], Kumar Anand[1,2], Dalibor Fiala[1,3], and Martin Schramm[2]

[1] University of West Bohemia, Pilsen, Czech Republic
{stmerz,dalfia}@kiv.zcu.cz
[2] Deggendorf Institute of Technology, Deggendorf, Germany
{stefanie.merz,michael.heigl,abhishek.subedi,kumar.anand,martin.schramm}@th-deg.de
[3] CCA Group a.s., Pilsen, Czech Republic

Abstract. Intrusion Detection Systems (IDS) are efficient solutions that enable cyber attacks to be detected in hosts and networks. As cyber attacks are developing rapidly, anomaly-based IDS systems in particular offer many advantages, but also disadvantages, including alert flooding. Therefore the sequential step of alert correlation is necessary to reduce the number of alerts and to provide previously unseen attacks with important contextual information for further operator analysis. This paper investigates how robust attack signatures - representatives derived from the communication structure of alert clusters - are for known and unknown attacks. For this purpose, the publicly available datasets CIDDS-001, CIDDS-002 and CIC-IDS2017 were used, which share common but also different attack categories and were recorded at different times and in different networks. In the case of the comparison CIDDS-001/CIDDS-002, an accuracy of 85% for attack category assignment based on signatures was achieved, while the combination CIC-IDS2017/CIDDS-001 was only accurate in 27% of assignments.

Keywords: Intrusion Detection Systems · Network Security · Alert Correlation · Attack Signatures

1 Introduction

With increasing digitalization and the high level of interconnectivity in all areas, it is no longer sufficient to protect a company's digital assets from unauthorized malicious access with just a single line of defense, e.g. a firewall at the boundary of the companies digital infrastructure. In fact, the complex networking of systems and the connection of devices to the Internet offer an attacker a wide-ranging attack landscape, which makes subsequent security measures necessary. Network Intrusion Detection Systems (NIDS) have proven to be an efficient tool for monitoring and detecting anomalies in company networks. The detection

X. Yang et al. (Eds.): ICICT 2026, LNNS 2029, pp. 11–24, 2026.
https://doi.org/10.1007/978-3-032-28964-3_2

either happens based on known attack patterns, rules or based on deviations from normal behavior. The later is called anomaly-based network IDS (A-NIDS) and is capable of detecting previously unseen attacks, also called zero-day exploits. But despite this ability, A-NIDS, which are deployed at strategic nodes within the company's network, are prone to generate a high number of alerts, making manual alert analysis time-consuming and complex. Therefore further processing in the form of alert correlation is needed, in order to reduce alerts on the one hand and alert enrichment on the other, especially in an unsupervised setting.

This paper builds upon a twofold alert clustering (temporal and similarity-based), by introducing the generation of attack signatures, that represent each alert cluster, with a subsequent comparison to attack references. Thereby, this paper is based on our previous work [10], which was published as a journal article and introduces *Streaming Outlier Analysis for Attack Pattern Recognition (SOAAPR)*, a new framework for anomaly detection and attack pattern recognition in streaming data. Complementarily, the current paper provides a detailed analysis of the robustness and efficiency of the denoted alert signaturing as well as a scenario comparison part. Therein the generation of signatures sig_{com} and a scenario comparison are analysed using the publicly available datasets CIDDS-001 [23], CIDDS-002 [24] and the improved version of CIC-IDS2017 [26]. These datasets contain numerous attacks, including multiple time-gapped instances of the same attack scenario, and therefore enable a detailed verification of the robustness of the proposed attack signature, as distinct instances of the same attack scenarios are supposed to show greater similarity.

The research questions of our study can be summarised as follows:

- How robust is the signature-generation method when applied to the same attack type recorded at different points in time and with a different attack tool parametrization?
- If no reference exists for an attack category, can signature-based scenario comparison still yield some added value to the alerts?

The structure of this paper is as follows: Following the introduction, Sect. 2 reflects on the state of the art and related work. Section 3 describes the components relevant to this paper and the methodology of SOAAPR. The description of the experiments carried out is provided in Sect. 4, where the initial results are presented and then discussed (Sect. 5). The conclusion is given in Sect. 6 and further work is presented.

2 Related Work

The necessity of alert correlation for supporting alert reduction and manual analysis is highly recognized by the research community [2,3,6,11–13,16–19]. The research ranges from alert clustering, generation of representatives, e.g. meta-alerts, hyper-alerts or attack signatures, to the derivation of attack sequences, e.g. [14,21,27] and event prediction [1]. According to [25], alert-clustering methods are commonly grouped into similarity-based, sequence-based, and case-based

approaches. Albasheer et al. [1] extend this taxonomy to include similarity-, statistical-, knowledge-, and hybrid-based approaches, developing a detailed alert-correlation taxonomy. Especially in an unsupervised anomaly detection setting, temporal- or similarity-based clustering, based on minimal sets of network alert attributes provides an efficient solution, as less expert knowledge is required. Furthermore, the aggregation and information enrichment of alert clusters, including the assignment of potential categories, is essential for subsequent manual analysis. Especially in the unsupervised area, as alerts from NIDS most often do not contain any attack categories or severity levels. For that purpose most approaches subsequently form so-called hyper-/meta-alerts or signatures after aggregation.

Since the communication signature (sig_{com}) is conceptually related to hyper- and meta-alerts, we first review related work on the condensed representations of aggregated alerts. These approaches provide valuable alternatives, but differ from our focus on graph-based alert correlation (see Sect. 3). Secondly, related work dealing with graph-based alert correlation is particularly emphasized.

Hyper- and meta-alert generation: Castillo-Fernández et al. [4] developed a multi-layered approach in which low-level hyper-alerts are aggregated from IDS alerts and iteratively enriched through hyper-alert correlation and contextual information, e.g., flows or protocols. They extended their work by the correlation with attack models, e.g. MITRE ATT&CK[1] in their follow-up publication [22]. A featureless approach leveraging LSTM RNN and Latent Semantic Analysis (LSA) to map text-based alerts into vector space is proposed by [15]. The alerts in vector space are clustered, utilizing the DBSCAN algorithm [7], into hyper-alerts to generate an aggregated view of the attack scenario without relying on expert knowledge. Hyper-alerts are labeled according to the incidents IDs of the contained alerts, which reflects the corresponding attack scenario. In order to perform a causal analysis for APT detection, Khosravi and Ladani [14] process the alerts generated by a SIEM into meta-alerts, which correlate to the respective step within an APT attack sequence. Each meta-alert is a six-value tuple incorporating source and destination IP address and port information, as well as timestamp, and alert class. Based on the target IP address, the set of meta-alerts for a host is determined, which can be used for causal analysis. Landauer et al. [16], on the other hand, pursue a domain-independent correlation approach in which alerts are first aggregated into groups based on their temporal proximity. These groups are then represented as meta-alerts by merging the underlying alert attributes, whereby no specific alert format is required.

Graph-based Alert Correlation: Cormode et al. [5] designed a framework to use signatures in communication graphs, and identified three signature test schemes: persistence, uniqueness, and robustness. Their experiment focused on robustness of signature schemes to enterprise TCP network traffic, where communication graphs were built using traffic flows over time windows. To test robustness, edges were randomly inserted and deleted to obtain a perturbed graph $G_t^{'}$ from graph G_t. They then examine whether a graph's signature is

[1] https://attack.mitre.org/.

closer to its perturbed version than to signatures of others. Haas et al. [8] utilize a graph-based correlation approach, through community clustering with clique percolation and assign four different categories to the alert clusters, *one-to-one*, *one-to-many*, *many-to-one* and *many-to-many*, depending on the existing patterns within a communication graph. In their follow-up paper [9], this approach is extended by the motif patterns proposed by Milo et al. [20]. The occurrence of various motif patterns and their frequency within the communication graph are examined and converted into a representative signature. This approach was adapted and extended by SOAAPR, which is why it is discussed in more detail in Sect. 3.

While the work of Cormode et al. [5] is relevant, an evaluation of robustness of attack signatures for real-world cyberattacks, such as CIDDS and CIC-IDS2017 datasets, is still limited. To the best of our knowledge, no prior work rigorously evaluates the robustness of alert signatures across time-separated instances of the same attack or across different networks using publicly available IDS datasets.

3 Methodology

For a detailed overview on *Streaming Outlier Analysis for Attack Pattern Recognition* (SOAAPR), we recommend referring to the original article [10]. For this paper the components *Alert Clustering*, *Signature Generation* and *Signature Comparison* are relevant (highlighted in Fig. 1). While the *Alert Clustering* serves as a prerequiste to aggregate A-NIDS alerts into alert clusters C_i, the *Signature Generation* and *Scenario Comparison* components are the basis of this paper's discussions.

Signature Generation: The generation of signatures enables a compressed, fixed-size, privacy-preserving representation of a cluster, which is acutally an attack category. Taking the communication pattern, the most important features and the temporal behaviour within a cluster into account, SOAAPR introduces

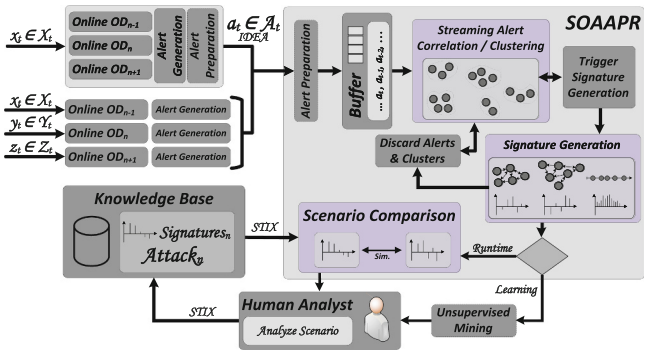

Fig. 1. Schematic overview of the Streaming Outlier Analysis for Attack Pattern Recognition framework [10] with highlighted relevant components.

the signatures sig_{com}, sig_{attr} and sig_{temp}. Those three signatures enable a condensed view on the cluster and allow for easy comparisons. As this work builds directly upon labeled network data, without performing preceeding ML-based outlier detection and streaming alerts, sig_{attr} and sig_{temp} are discarded in the ongoing process. For the generation of sig_{com} SOAAPR utilizes the concept of motif signatures proposed by Milo et al. [20] and adopted by Haas et al. [9]. In their approach, the communication relationship between two hosts is represented, utilizing their IP address (source/destination) and port (source/destination) information, as follows: $(ip_{src} : port_{src} \rightarrow ip_{dst} : port_{dst})$.

This notation enables the representation of an alert cluster C_i as a communication graph G_{com}, whereas nodes are IP and port information and directed edges are the communication relations between those nodes, e.g. information flow from attacker to victim or vice versa. The predefined, fixed-size motif patterns $m_n, n = 0, 1, ..., 15$, seen in Fig. 2, are used to divide the communication graph into subgraphs $G_m \subset G_{com}$. The number of occurrences of each motif pattern in G_{com} results in the signature sig_{com}.

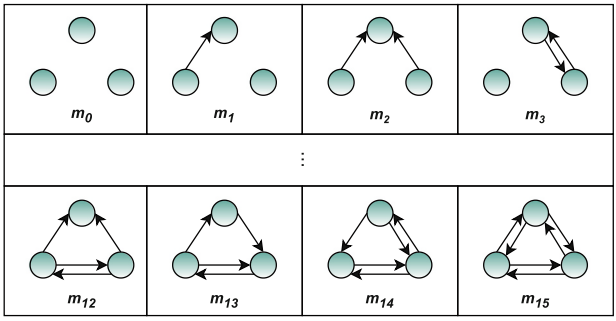

Fig. 2. Excerpt of different motifs created with three nodes (adapted from [9])

Signature Comparison: The signature comparison process takes place in two consecutive steps. In order to make signatures comparable, they are normalized using the Z-score method [20]. Subsequently the similarity of two normalized signatures can be calculated by the cosine similarity between their signature vectors, utilizing Eqs. (1) and (2). Where $\boldsymbol{M^Z_{1/2}}$ denotes the normalized signature vectors, $< \boldsymbol{M^Z_1}, \boldsymbol{M^Z_2} >$ the inner product and $||\boldsymbol{M^Z_{1/2}}||_2$ the Euclidean norm.

$$\text{sim}(\boldsymbol{M^Z_1}, \boldsymbol{M^Z_2}) = \frac{cos^{-1}(\phi)}{\pi} \tag{1}$$

$$cos(\phi) = \frac{< \boldsymbol{M^Z_1}, \boldsymbol{M^Z_2} >}{||\boldsymbol{M^Z_1}||_2 \cdot ||\boldsymbol{M^Z_2}||_2} \in [0, 1] \tag{2}$$

4 Experimental Evaluation

4.1 Experimental Setup

The experiments were performed on a virtual Ubuntu 22.04.02 system with an AMD EPYC 7543 2.6 GHz Processor. From the overall CPU resources the virtual machine was equipped with 16 cores, 32 GB RAM and a 150 GB hard drive. Python 3.10.12 was used for the implementation of *SOAAPR*'s alert correlation components.

The datasets for the experiments were selected according to the following criteria: public availability, timeliness (year of creation), availability of labels (direct or indirect), presence of the features IP (src/dst), port (src/dst) and timestamp, as well as, the multiple presence of the same or similar attack scenario. This resulted in the selection of the CIDDS-001 and CIDDS-002 datasets from Coburg University of Applied Sciences and CIC-IDS2017 dataset from the Canadian Institute for Cybersecurity.

CIDDS-001: This datasets contains labeled flow-data collected over the course of four weeks (03/2017 to 04/2017) for an *OpenStack* and *External Server* environment. During week 1 and week 2 several attacks of the types *portscan*, *pingscan*, *brute force* and Denial of Service (*dos*) have been performed.

CIDDS-002: Created by the same authors as CIDDS-001, over the course of two weeks (08/2017), this dataset contains the *Netflow* data of an internal *OpenStack* environment. Throughout both weeks attack scenarios of type *pingscan* and *portscan* (*SYN scan, ACK scan, UDP scan, FIN scan*) have been performed.

From the given 14 features of the CIDDS-001 and CIDDS-002 dataset, eight features are used: (i) *Src IP*, *Src Port*, *Dest IP*, *Dest Port* and *Date first seen* for clustering and signature generation, and (ii) *class/label*, *attackType* and *attackID* for alert preparation and evaluation purposes.

CIC-IDS2017: This dataset contains various attack data in these categories: *Brute Force* (FTP/SSH), *DoS*, *Heartbleed*, *Web Attack*, *Infiltration*, *Botnet* and *DDoS*. From the 91 features generated by *CICFlowMeter*[2], analogously the features *Src IP*, *Dst IP*, *Src Port*, *Dst Port* and *Timestamp* have been used for clustering and signaturing. Finally, the feature *Label* serves the purpose of data preprocessing and evaluation.

4.2 Evaluation Strategy

The evaluation is conducted under the assumption of an ideal binary classification of an unspecified IDS and an optimum clustering, where each cluster represents one attack. Therefore, knowing the groundtruth of the CIDDS-001, CIDDS-002 and CIC-IDS2017 datasets, ideal clusters of only attack data are generated and used for signaturing. The derived signatures sig_{com} are then iteratively compared to the attack categories of the other datasets. This is intended to examine the following assumption:

[2] https://github.com/ahlashkari/CICFlowMeter.

- signatures of clusters from the same attack type will show the highest degree of similarity
- signatures of clusters from different attack types will show the lowest degree of similarity
- the relationship between similar attack types (e.g. scan attacks) is reflected in their degree of similarity

5 Discussion of Results

The degree of similarity between two signatures sig_{com} is denoted by the cosine similarity, calculated as specified in Eq. (2). Iteratively, the signatures of a dataset are compared with a reference dataset, while the dataset with the most overlapping attack scenarios have been selected as the reference/baseline. The maximum value of cosine similarity indicates the reference category with which the signature best matches. An adjustable parameter *threshold* can be used to define the similarity value from which a mapping between the signature and a category takes place.

As the groundtruth is known, the accuracy *acc* of the signature mapping could be determined, utilizing the following Eq. (3), where CS denotes the correctly mapped signatures and ToS the total of all mapped signatures:

$$acc = \frac{CS}{ToS} \tag{3}$$

CIDDS-001 with Baseline CIDDS-002: The heatmap in Fig. 3 visualizes the signature comparison results for all CIDDS-001's signatures and CIDDS-002's references, while Table 1 shows the accuracy of the mapping process for threshold values of 0.5, 0.9 and 0.99 for cosine similarity. Setting a relatively low threshold of 0.5 for assigning categories from CIDDS-002 to CIDDS-001 signatures, enables a mapping of all signatures (70 in total), while only an accuracy of 0.57 could be achieved. Especially because CIDDS-001's *brute force* and *dos* signatures, where no reference category in CIDDS-002 exist, are wrongly mapped to a *portscan* or *pingscan* category. Increasing the *threshold* value results in a drop of mapped signatures (59/70 and 40/70), but also improves the accuracy as all unmapped signatures are from *dos* and *brute force* type. Therefore, setting the *threshold* to 0.99 could achieve an accuracy of 0.85. While all of the *brute force* signatures are no longer mapped to a category, some *dos* signatures are still mapped to *portscan* references.

Taking a closer look at the *dos* attack, it established 10,000 connections to port 80 of a random target IP address. If a connection is established, TCP messages with static message content are sent. The attack thus takes place from a single attacker IP with a random source port (within ephemeral port range) to a fixed target port and a single target IP address. While the *portscans* are directed at all hosts in a randomly selected subnet, whereby the 1000 most frequent ports are scanned for each IP in this subnet. Therefore, a *portscan* is

initiated from a single source IP with a randomly selected port and is directed at many target ports of several target IPs. Graphically, the two attacks are thus structured differently, which is why the assignment made using cosine similarity must be regarded as incorrect.

It is also clear from the heatmap in Fig. 3 that signatures of the same attacks have the highest cosine similarity in most cases and that the cosine similarity between signatures of different attacks is lower. The color in the heatmap gradiates from yellow, indicating highest similarity, to a dark blue color, indicating less similarity. Therefore, it shows that, for example, the signatures of the brute-force attacks are less similar to the portscan signatures (heatmap tile with dark blue coloring) than, for example, the pingscan signatures (yellow-green coloring).

The networks that serve as the basis for CIDDS-001 and CIDDS-002 are similar (same IP address ranges), but differ in the number of subnets and the number of servers and clients per subnet. Also the attacks are executed in a similar fashion and parametrization. Taking this into account, it can be said that the assumptions made in Sect. 4.2 are fulfilled with 85% accuracy (with a threshold value of 0.99).

Table 1. Accuracy between signatures of CIDDS-001 with CIDDS-002 as a baseline

Threshold	ToS	Accuracy
0.5	70	0.57
0.9	59	0.68
0.99	47	0.85

CIC-IDS2017 with Baseline CIDDS-001: Analogously all cosine similarity values for CIC-IDS2017 signatures and CIDDS-001 categories are visualized in the heatmap in Fig. 4, while Table 2 represents the accuracy of the achieved signature-category mapping. For a *threshold* value of 0.5 all CIC-IDS2017 signatures, except *Heartbleed*, could be mapped with an accuracy of 0.25. Only *brute force* and the *portscan* signature could be mapped to the right category. What is noticeable here is that the accuracy does not improve when the threshold value is increased (to 0.9 and 0.99 respectively), while the number of assigned categories is reduced, which thus contradicts the assumption that attacks from different categories show the lowest degree of similarity.

Possible reasons for this could include different networks (despite normalization) and the different parameterization and execution of the attacks. CIC-IDS2017 relies on common tools such as *Patator*[3] for SSH and FTP brute force, *Hulk*[4], *GoldenEye*[5], *Slowloris*[6], and *Slowhttptest*[7] for DoS, *Low Orbit Ion Canon*

[3] https://github.com/lanjelot/patator.
[4] https://github.com/R3DHULK/HULK.
[5] https://github.com/jseidl/GoldenEye.
[6] https://github.com/gkbrk/slowloris.
[7] https://github.com/shekyan/slowhttptest.

(LOIC)[8] for DDoS and portscan, *Metasploit*[9] framework for the infiltration phase, as well as self-written scripts for the web attacks. The attacks within CIDDS-001, on the other hand, were simulated by custom Python scripts using libraries such as *socket*[10], *pexpect (pxssh)*[11] and *nmap*[12].

Table 2. Accuracy between signatures of CIC-IDS2017 with CIDDS-001 as a baseline

Threshold	ToS	Accuracy
0.5	25	0.24
0.9	21	0.29
0.99	11	0.27

Cosine similarity represents a percentual measure of the degree of similarity to the attack reference. This allows for additional levels of classification within an attack category in terms of the certainty of the assignment to the reference. For example, cosine similarity values that are close to the threshold indicate the probability that the signature has been incorrectly assigned. This can aid subsequent manual analysis, particularly in the unsupervised area. Other alert correlation methods, such as [8], assign communication relationships as categories or other techniques, as described in Sect. 2, generate hyper- or meta-alerts, which may contain an attack category based on the underlying alerts. A comparison with these techniques in terms of robustness was out of scope for this work.

6 Conclusion and Future Work

This research takes up SOAAPR, which was presented in our previous journal article, and examines in particular the components of *Signature Generation* and *Scenario Comparison*. To determine the signatures, the alerts within a cluster are converted into a communication graph and dedicated communication patterns, so-called motifs, are examined. The resulting fixed-size, anonymized and resource-efficient signature can be compared with references in the sequential step of the scenario comparison in order to assign an attack category.

The publicly available datasets CIDDS-001, CIDDS-002 and CIC-IDS2017 were selected for the analysis according to the criteria of public availability, timeliness, labels, and the presence of the attributes IP src/dst, port src/dst and timestamp. During the evaluation, the clustering is considered ideal and conducted based on the data's groundtruth. All three datasets show intersecting

[8] https://github.com/NewEraCracker/LOIC.
[9] https://www.metasploit.com/.
[10] https://docs.python.org/3/library/socket.html.
[11] https://pexpect.readthedocs.io/en/stable/.
[12] https://nmap.org/.

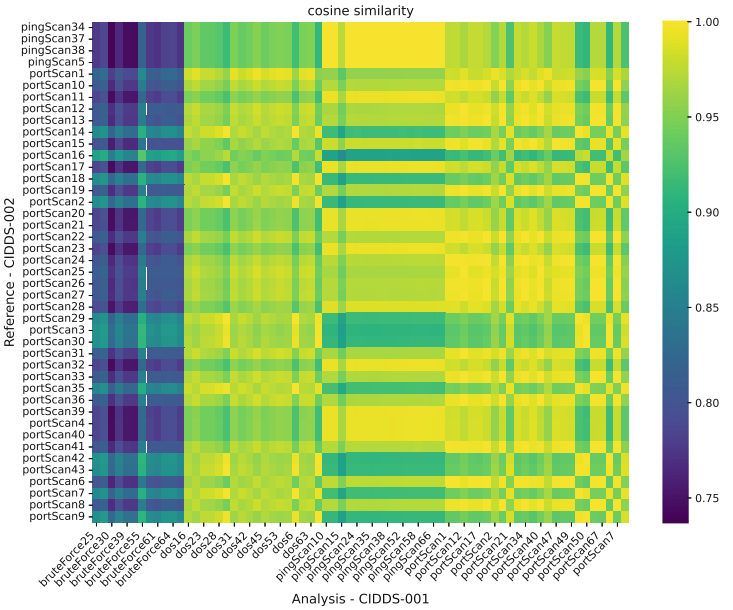

Fig. 3. Cosine similarity scores for CIDDS-001 and CIDDS-002

and unique attack categories, thus, for certain signatures there exists no reference. Nevertheless, these signatures were considered, as this is the case with novel attack types in reality and valuable initial information can be provided to a human operator through a comparison and partial assignment to references.

In the analysis of the CIDDS-001 signatures with the CIDDS-002 references, an accuracy of 85% was achieved with a threshold value of 0.99, as, additionally to the right mapping, signatures for which no references exist were not assigned to a category. In contrast, when analyzing the CIC-IDS2017 signatures with CIDDS-001 as a reference with the same threshold value, only 27% of the signatures could be accurately assigned. We therefore don't deem the signatures sig_{com}, transferred to other networks and execution, to be robust.

In our future work, we will investigate additional datasets with further attack categories, including evolving threats, as well as deployment in operating environments (company network, system operators, etc.) to extend the robustness and usability analysis. In order to draw meaningful conclusions about robustness on realistic, noisy data, the outlier detection and alert clustering should no longer be considered ideal, which was out of scope for this publication. Binary classification of ML-based outlier detection algorithms on noisy data should, for example, generate impurities in the form of false positives and false negatives. The alerts thus generated (true and false positives) should be clustered, due to their temporal occurrence and the similarity of their alert attributes. At this stage suboptimal clustering can further distort the attack clusters. Subsequently, a signature can be derived for these distorted clusters and their robust-

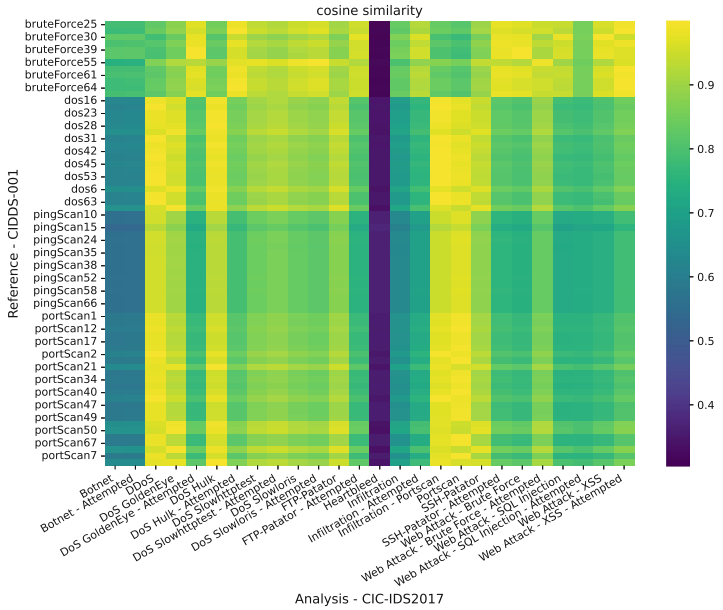

Fig. 4. Cosine similarity scores for CIC-IDS2017 and CIDDS-001

ness should be determined in comparison to optimal reference signatures. This allows to determine the degree of contamination at which a robust representative attack signature can still be achieved. To mitigate distortions and enhance signatures robustness, embedding techniques should be explored for alert preprocessing, enabling filtering and noise reduction through latent space representation of alert data. Processing on streaming data and incorporating ML models for outlier detection further allow the generation of sig_{temp} and sig_{attr}. Those signatures, taking into account the temporal occurence and the feature importance for the models decision, could enrich the attack context further. The benefit of a tuple consisting of the three signatures sig_{com}, sig_{temp}, and sig_{attr}, as opposed to the standalone signature sig_{com}, should therefore be exhaustively investigated. In addition, generalized approaches will be pursued to derive graphical attack baselines for networks, as it is not practically feasible to first carry out numerous attacks in a corporate network to generate data for references. Among other things, graph-based machine learning as well as publicly available vulnerability databases, e.g. CVE[13] and attack frameworks such as MITRE ATT&CK, will be explored for this purpose.

[13] https://www.cve.org/.

Acknowledgments. This research and the publication were funded by German Federal Ministry of Research, Technology and Space under grant number 16KIS1946 ("KMU-innovativ Verbundprojekt: Attribution von Cyberangriffen durch KI-erweiterte Angriffserkennungssysteme mit Alarm-Korrelation in industriellen Netzwerken - CAIDAN").

References

1. Albasheer, H., Md Siraj, M., Mubarakali, A., Elsier Tayfour, O., Salih, S., Hamdan, M., Khan, S., Zainal, A., Kamarudeen, S.: Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey **22**(4), 1494 (2022). https://doi.org/10.3390/s22041494
2. Alserhani, F.M.: Alert correlation and aggregation techniques for reduction of security alerts and detection of multistage attack. Int. J. Adv. Stud. Comput., Sci. Eng. **5**(2), 1 (2016)
3. Bateni, M., Baraani, A., Ghorbani, A.A.: Using artificial immune system and fuzzy logic for alert correlation. Int. J. Netw. Secur. **15**(3), 190–204 (2013)
4. Castillo-Fernández, E., Díaz-Verdejo, J., Estepa Alonso, R., Estepa Alonso, A., Muñoz Calle, J., Mabinabeitia, G.: Multistep cyberattacks detection using a flexible multilevel system for alerts and events correlation. In: European Interdisciplinary Cybersecurity Conference, pp. 1–6. ACM (2023). https://doi.org/10.1145/3590777.3590778
5. Cormode, G., Korn, F., Muthukrishnan, S., Wu, Y.: On signatures for communication graphs. In: IEEE 24th International Conference on Data Engineering, pp. 189–198. IEEE (2008)
6. De Alvarenga, S.C., Barbon, S., Miani, R.S., Cukier, M., Zarpelão, B.B.: Process mining and hierarchical clustering to help intrusion alert visualization **73**, 474–491 (2018). https://doi.org/10.1016/j.cose.2017.11.021
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, vol. 34, pp. 226–231. AAAI Press (1996)
8. Haas, S., Fischer, M.: GAC: graph-based alert correlation for the detection of distributed multi-step attacks. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 979–988. ACM (2018). https://doi.org/10.1145/3167132.3167239
9. Haas, S., Wilkens, F., Fischer, M.: Efficient attack correlation and identification of attack scenarios based on network-motifs. In: 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), pp. 1–11. IEEE (2019). https://doi.org/10.1109/IPCCC47392.2019.8958734
10. Heigl, M., Weigelt, E., Urmann, A., Fiala, D., Schramm, M.: Exploiting the outcome of outlier detection for novel attack pattern recognition on streaming data **10**(17), 2160 (2021). https://doi.org/10.3390/electronics10172160
11. Hofmann, A., Sick, B.: Online intrusion alert aggregation with generative data stream modeling **8**(2), 282–294 (2009). https://doi.org/10.1109/TDSC.2009.36
12. Husák, M., Kašpar, J.: AIDA framework: real-time correlation and prediction of intrusion detection alerts. In: Proceedings of the 14th International Conference on Availability, Reliability and Security, pp. 1–8. ACM (2019). https://doi.org/10.1145/3339252.3340513

13. Julisch, K.: Clustering intrusion detection alarms to support root cause analysis **6**(4), 443–471 (2003). https://doi.org/10.1145/950191.950192
14. Khosravi, M., Ladani, B.T.: Alerts correlation and causal analysis for APT based cyber attack detection **8**, 162,642–162,656 (2020). https://doi.org/10.1109/ACCESS.2020.3021499
15. Kidmose, E., Stevanovic, M., Brandbyge, S., Pedersen, J.M.: Featureless discovery of correlated and false intrusion alerts. IEEE Access **8**, 108,748–108,765 (2020)
16. Landauer, M., Skopik, F., Wurzenberger, M., Rauber, A.: Dealing with security alert flooding: using machine learning for domain-independent alert aggregation **25**(3), 1–36 (2022). https://doi.org/10.1145/3510581
17. Li, Z., Zhang, A., Lei, J., Wang, L.: Real-time correlation of network security alerts. In: IEEE International Conference on e-Business Engineering (ICEBE'07), pp. 73–80. IEEE (2007). https://doi.org/10.1109/ICEBE.2007.69
18. Liang, W., Chen, Z., Wen, Y., Xiao, W.: An alert fusion method based on grey relation and attribute similarity correlation **12**(8), 25–30 (2016). https://doi.org/10.3991/ijoe.v12i08.5958
19. Maosa, H., Ouazzane, K., Ghanem, M.C.: A hierarchical security event correlation model for real-time threat detection and response **4**(1), 68–90 (2024). https://doi.org/10.3390/network4010004
20. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks **298**(5594), 824–827 (2002). https://doi.org/10.1126/science.298.5594.824
21. Moskal, S., Yang, S.J., Kuhl, M.E.: Extracting and evaluating similar and unique cyber attack strategies from intrusion alerts. In: IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 49–54. IEEE (2018). https://doi.org/10.1109/ISI.2018.8587402
22. Muñoz-Calle, J., Alonso, R.E., Estepa Alonso, A., Díaz-Verdejo, J.E., Castillo Fernández, E., Madinabeitia, G.: A flexible multilevel system for mitre ATT&CK model-driven alerts and events correlation in cyberattacks detection **30**(9), 1184–1204 (2024). https://doi.org/10.3897/jucs.131686
23. Ring, M., Wunderlich, S., Grudl, D., Landes, D., Hotho, A.: Flow-Based Benchmark Data Sets for Intrusion Detection, pp. 361–369 (2017)
24. Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A.: Creation of flow-based data sets for intrusion detection. J. Inf. Warf. **16**(4), 41–54 (2017)
25. Salah, S., Maciá-Fernández, G., Díaz-Verdejo, J.E.: A model-based survey of alert correlation techniques **57**(5), 1289–1317 (2013). https://doi.org/10.1016/j.comnet.2012.10.022
26. Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018). https://doi.org/10.5220/0006639801080116
27. Wang, X., Gong, X., Yu, L., Liu, J.: MAAC: novel alert correlation method to detect multi-step attack. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 726–733. IEEE (2021). https://doi.org/10.1109/TrustCom53373.2021.00106