Supplementary Dataset Information and Results for the Paper You Cannot Escape Me: Detecting Evasions of SIEM Rules in Enterprise Networks

Rafael Uetz[*] Marco Herzog[*] Louis Hackländer[*] Simon Schwarz[†] Martin Henze[‡,*]

[*]*Fraunhofer FKIE* [†]*University of Göttingen* [‡]*RWTH Aachen University*

1 Enterprise Dataset Collection and Analysis

Our evaluation of AMIDES is based on four weeks of Windows process creation events (155 million) collected from an enterprise SIEM system, which are used as benign events for training and validation as described in Section 6 of the paper. Since these data cannot be published due to security and privacy concerns, we present details on our collection and analysis process in the following with the goal of enabling researchers to reproduce similar experiments in other environments.

Collection The enterprise operates a remote-controlled browser system running on approximately 400 Windows servers where each user receives a remote desktop environment to run a browser and associated applications for viewing documents, images, and other common files, thereby isolating the internal network from external security threats. These servers are equipped with Microsoft Sysmon to provide extended threat detection and forensic analysis capabilities [1]. All Sysmon events, including process creation events (ID 1), are first collected on dedicated servers using Windows Event Forwarding [1] and then ingested into an OpenSearch instance using Winlogbeat, from where we extracted them using the OpenSearch Python client.

Analysis After data extraction, we first performed several quick coherence checks to make sure the number, timeframe, and format of events are correct. Afterwards, we performed an in-depth analysis of the collected command line strings with the goal of finding malicious instances, particularly evasions. For this purpose, we proceeded in three steps:

1. We automatically removed duplicates from the 155 million command line strings, leaving us with 72 million.
2. We manually looked for command lines that appeared often with slight variations such as arguments with different timestamps or unique IDs. We wrote 112 regular expressions to match and join such similar command lines, ending up with 88 015 different command lines remaining.
3. We manually checked all of these command lines for suspicious or malicious behavior, but did not find any.

The whole analysis process took us approximately six weeks of full-time work. However, note that the effort for AMIDES users to check extracted training data for potential evasions is *significantly lower* for several reasons:

- The purpose of our analysis was not only to assure benignity, but also to develop a sensible feature extraction method for AMIDES. With this method in place, no regular expressions need to be created and only 159 706 unique feature vectors exist whose command lines need to be sifted.
- This number can be further reduced significantly by removing terms from the feature set that appear very rarely in the dataset and are therefore unlikely to appear in future events. Particularly, 64 % of terms only appear in one single command line (e.g., temporary file names).
- Lastly, when retraining AMIDES, only command lines including terms that never appeared before need to be sifted.

Summarizing, we are confident that our enterprise dataset is truly benign and that ensuring benignity of such a dataset is possible with reasonable effort for users of AMIDES.

2 Results for the Synthetic SOCBED Dataset

Here we present results for AMIDES on the synthetic SOCBED dataset, as mentioned in the paper. Since this dataset is unrealistic and rather small, the results are primarily meant to serve as a congruence check for researchers wishing to reproduce our evaluation using AMIDES.

Figure 1 shows AMIDES' classification performance and compares it to the benchmark approach that learns from attack events ("matches") instead of SIEM rules. We can see that both AMIDES and the benchmark approach achieve near-perfect classification at the default threshold (0.5). While the precision of the benchmark increases earlier, its recall also drops off more sharply. Yet, both approaches exhibit a broad threshold range with high precision and recall.

Figure 2 visualizes the results of the rule attribution step, again in the same way as presented in the paper. The results are very similar to those on the enterprise dataset despite the completely different set of benign events. However, the similarity is reasonable since both the utilized rules and evasions are the same, leading to similar attribution ranks.

Figure 3 depicts the influence of tainted training data. We can see that the recall for different fractions declines in a similar shape as seen on the enterprise dataset, but the classification performance at the default threshold (0.5) still remains very good (i.e., precision and recall are above 95 %) since the classes in the SOCBED dataset are almost balanced and therefore the precision curve declines slowly instead of falling off sharply as it does on the enterprise dataset.

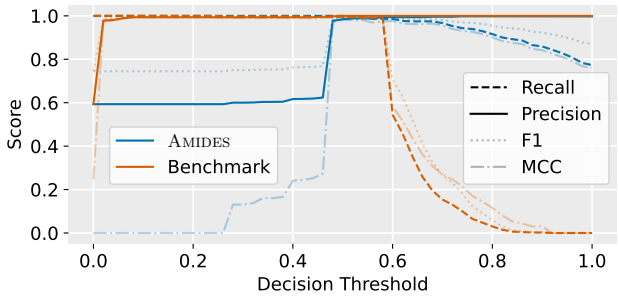

Figure 1: Classification performance

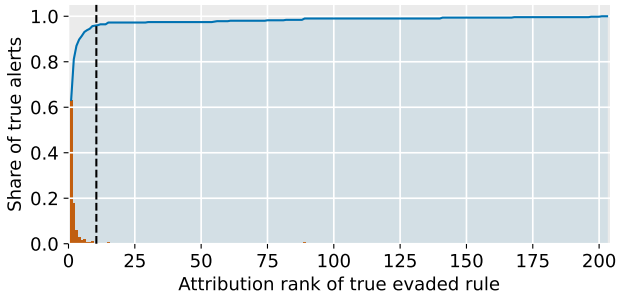

Figure 2: Rule attribution performance

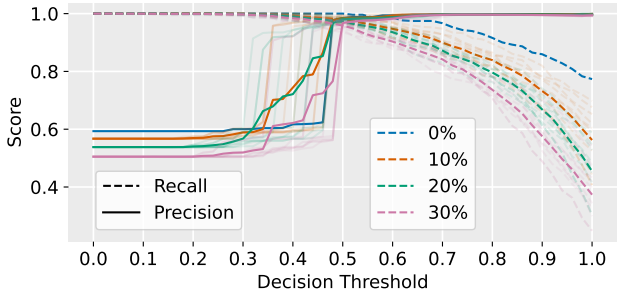

Figure 3: Influence of tainted training data

References

[1] Microsoft. Use Windows Event Forwarding to help with intrusion detection, 2023. https://learn.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection.