

What We Talk About When We Talk About Logs: Understanding the Effects of Dataset Quality on Endpoint Threat Detection Research

Jason Liu, Muhammad Adil Inam, Akul Goyal, Andy Riddle, Kim Westfall, and Adam Bates

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA

{jdlui2, mainam2, akulg2, rriddle2, kw26, batesa}@illinois.edu

Abstract—Endpoint threat detection research hinges on the availability of worthwhile evaluation benchmarks, but experimenters’ understanding of the contents of benchmark datasets is often limited. Typically, attention is only paid to the realism of attack behaviors, which comprises only a small percentage of the audit logs in the dataset, while other characteristics of the data are inscrutable and unknown.

We propose a new set of questions for what to talk about when we talk about logs (i.e., datasets): *What activities are in the dataset?* We introduce a novel visualization that succinctly represents the totality of 100+ GB datasets by plotting the occurrence of provenance graph neighborhoods in a time series. *How synthetic is the background activity?* We perform autocorrelation analysis of provenance neighborhoods in the training split to identify process behaviors that occur at predictable intervals in the test split. Finally, *How conspicuous is the malicious activity?* We quantify the proportion of attack behaviors that are observed as benign neighborhoods in the training split as compared to previously-unseen attack neighborhoods. We then validate these questions by profiling the classification performance of state-of-the-art intrusion detection systems (R-CAID, FLASH, KAIROS, GNN) against a battery of public benchmark datasets (DARPA Transparent Computing and OpTC, ATLAS, ATLASv2). We demonstrate that synthetic background activities dramatically inflate True Negative Rates, while conspicuous malicious activities artificially boost True Positive Rates. Further, by explicitly controlling for these factors, we provide a more holistic picture of classifier performance. This work will elevate the dialogue surrounding threat detection datasets and will increase the rigor of threat detection experiments.

1. Introduction

From its very beginnings, host intrusion detection research has been impeded by the availability of evaluation data. Industry is understandably reluctant to publish their own event logs, which may contain proprietary information, while generating benchmark datasets that are realistic in scale and complexity is tedious and unrewarding for researchers. When individual researchers or funded programs invest in dataset creation, design considerations often begin and end

with the sophistication of the attack behaviors. For example, the DARPA Transparent Computing program [37] hired professional penetration testers to conduct well-documented attack engagements. However, attacks comprise only a small proportion of a dataset, and it is much more difficult to understand the rest of the dataset’s contents.

While attacks are exciting and important to consider, we currently lack the means to holistically evaluate the quality of endpoint threat detection datasets. Even conceptualizing the contents of a log dataset, which may contain hundreds of gigabytes of system events (or more), is a challenge. Neglecting the contents of background activity is especially problematic for anomaly-based intrusion detection methods, a popular topic of academic research, because classification performance in anomaly detection is a function of the relationship between typical (benign) and atypical (attack) data. A more critical assessment of our benchmark datasets would help us to better evaluate the merit of threat detection systems, and may even help to explain why so many promising-looking methods have failed to transition to practice.

In this work, we propose a battery of principled and generalizable metrics for measuring the quality of endpoint detection datasets. Our approach builds on established threat investigation approaches using *data provenance*, which interprets audit logs as a causal relationship graph describing the history of system execution. We present novel methods for quantifying and analyzing the properties of these graphs to facilitate comparison across datasets. Our contributions are organized as a series of questions that we should be asking when we “talk about logs” — that is, when we carefully consider what these datasets can actually tell us about threat detection systems:

What activities are in the dataset? (§3) Even reasoning about the contents of a large intrusion detection dataset can prove challenging. We demonstrate that a “bird’s eye view” of the data can be obtained by tracking the occurrence of each process neighborhood in the provenance graph as a time series, enabling high-level reasoning about the background activity in a dataset. Using this approach, we recover undocumented information about popular datasets such as machine downtime and other experimental artifacts.

How synthetic is the background activity? (§4) We propose a metric for measuring the contribution of synthetic

TABLE 1: Properties of public benchmark datasets for endpoint threat detection. If datasets contain multiple event streams (e.g., network logs), only system log statistics are reported. Datasets that report syscall IDs without arguments cannot be used with provenance analysis. Label granularity describes the specificity of the ground truth labeling. Datasets used in this work are shaded. Unavailable information is marked N/A.

Dataset	Year	Platform	Duration	Size (GB)	Events (mil)	Syscall... IDs?	Args?	Label Gran.
UNM [24]	1998	Solaris/Linux	N/A	0.3	68	✓	✗	Process
DARPA'98 [16]	1998	Solaris	9 weeks	14	59	✓	✓	Process
DARPA'99 [42]	1999	Solaris	5 weeks	3	13	✓	✓	Process
ADFA-LD [13], [14], [15]	2013	Ubuntu 11.04	N/A	6	3	✓	✗	Log
-WD	2014	Windows XP	N/A	11	138	✓	✗	Log
-WD-SAA	2014	Windows XP	N/A	13	0.4	✓	✓	Log
StreamSpot [44]	2016	Linux	5 hours†	12	136	✓	✓	Log
E3 Cadets [31], [38]	2018	FreeBSD	2 weeks	9	41	✓	✓	Entity*
Theia	2018	Ubuntu 12.04	2 weeks	16	106	✓	✓	Entity*
Trace	2018	Ubuntu 14.04	2 weeks	136	835	✓	✓	Entity*
FiveDirections	2018	Windows 10	2 weeks	60	282	✓	✓	Entity*
E5 Cadets [31], [57]	2019	FreeBSD	2 weeks	250	1194	✓	✓	Entity*
Theia	2019	Ubuntu 12.04	2 weeks	159	1141	✓	✓	Entity*
Trace	2019	Ubuntu 14.04	2 weeks	723	4666	✓	✓	Entity*
FiveDirections	2019	Windows 10	2 weeks	682	3208	✓	✓	Entity*
OpTC [1]	2020	Windows 10	9 days	8280	16220	✓	✓	Entity*
ATLAS [3]	2021	Windows 10	2 days†	2	3	✓	✓	Entity*
ATLASv2 [51]	2023	Windows 10	5 days	21	32	✓	✓	Entity

* Entity labels are incomplete, imprecise, or qualitative.

† Logs span short, highly disjoint time periods

activity to the background activity in a dataset. Our approach recovers the periodicity of process neighborhoods in the provenance graph through autocorrelation. We find that we can predict the occurrence of nearly 60% of all process neighborhoods in the time series with a less than 10% false negative rate.

How conspicuous is the attack activity? (§5) Especially in anomaly detection, the (dis)similarity between typical and atypical data dictates classification performance. An attack that is extremely dissimilar to background activity lacks covertness and may be trivial to identify. To provide a measure of these effects, we compare process neighborhoods observed in the training (pre-attack) period to malicious neighborhoods observed in the test period. We discover a high degree of variance in attack covertness between datasets, ranging from 99% (E3 Trace) all the way to 0% (ATLASv2).

To demonstrate the value of these metrics, we conduct an exhaustive evaluation where we manipulate dataset quality metrics and observe the impact on state-of-the-art threat detection systems R-CAID [29], FLASH [58], and KAIROS [11] (§6), as well as a commodity-off-the-shelf Graph Neural Network (GNN). We go on to discuss how our findings should inform the generation of future datasets (§7), then consider prior work on security dataset metascience (§8) before concluding (§9).

2. Endpoint Threat Detection Datasets

Since the earliest intrusion detection research, classification performance evaluation required benchmark datasets. We summarize public datasets for endpoint (i.e., traditional computer) intrusion detection research in Table 1. Qualitative descriptions of the datasets can be found in Section A. Here, we call attention to the disparities in ground truth labeling practices between datasets. On the finer-grained

TABLE 2: Dataset attack labeling statistics, including the number of attack processes recovered and the proportion of attack processes to all processes per dataset.

Dataset	# Attack	% Attack
E3 Cadets	2821	2.6
E3 FiveDirections	9982	4.2
E3 Theia	47	0.02
E3 Trace	86	0.001
ATLASv2 H1	149	1.0
ATLASv2 H2	17	0.1
OpTC Host 201	11	0.07
OpTC Host 501	50	0.3

side, datasets label malicious processes or every malicious entity (including processes) implicated in the attack. Other datasets report only coarse-grained information about which log-file of system-wide activity contains the attack, i.e., both benign and malicious processes are present in the log. Within these categories, the precision and completeness of labels still vary. For example, DARPA TC and OpTC provide only qualitative attack descriptions that require trial-and-error to identify the specific log entities. ATLAS identifies only a subset of the malicious entities (2-3) per attack, when there are actually many more.

Standardizing Ground Truth Labels. As IDS become more advanced, the potential for researcher intent to influence evaluation results becomes a greater concern. Individual experimenters are often left to make their own decisions as to which events/entities should be designated as malicious, creating a conflict of interest. It also complicates comparing datasets, which is a central goal of this work.

To address these shortcomings, we make use of a standardized and evidence-based method of labeling in which *all processes on the attack chain are labeled as malicious*. In our approach, the human is still responsible for identifying the *root cause(s)* and *terminal impact(s)* of an attack – in other words, where the attack begins and ends – but not intermediary steps. We then use Fang et al.’s method of identifying a concise attack chain by taking the intersection of the forward trace from the root cause(s) and the backwards trace from the impact(s) [20]. All processes in this chain are labeled as *Malicious*, reducing experimenter degrees of freedom. However, due to the imprecision of provenance graphs, it is often the case that benign entities appear to have causal relationships with attack entities. It is unreasonable to expect a classifier to label these entities as malicious, because they are not, but it is also unreasonable to expect a classifier to ignore the apparent causal relationship to the attack chain. To resolve this dilemma, we introduce a *Contaminated* label to denote samples that should be withheld from the test set. We identify contaminated processes by conducting a forward trace from the root cause(s) of the attack, marking everything that is not already labeled as *Malicious* as *Contaminated*.

General statistics about the number of implicated attack processes from this method are shown in Table 2. Our labels, as well as a more detailed description of our labeling

methodology, are available online;¹ the evaluation in this paper can be interpreted as an experimental validation of our labeling methodology. Despite our standardized process, labeling remains a time-consuming task. Our goal is to eventually provide uniform labels for all endpoint threat detection datasets past, present, and future.

3. What Activities are in the Dataset?

Given that public threat detection research is largely based on anomaly detection – the practice of identifying suspicious activity by learning a model of standard activity patterns – surprisingly little attention has been paid to the characteristics of background activity in benchmark datasets. Were the background activities generated procedurally (synthetically) or by human operators? If procedurally, how were activities made naturalistic? If by humans, even though the activities are trivially naturalistic, are they representative of the target systems we seek to defend?

Questions like these are hard to answer through the inspection of massive and obtuse system logs, so they simply haven’t been asked. While it would seem that data provenance and causal analysis would solve this problem by providing a means of visualizing log data graphically, visualizing an entire dataset of hundreds of millions of events results in a graph that is far too large for humans to interpret. For example, one of the smallest datasets we study, E3 Cadets, produces a graph with around 300,000 nodes and 500,000 edges in a single day, nevermind the remaining ten days. To answer high-level questions about dataset contents, we need a human-digestible method of visualizing entire datasets.

3.1. Plotting Provenance Over Time

We propose sampling the process neighborhoods in a provenance graph as a time series. Neighborhoods, representing behaviors in the graph, are defined by an n -hop backwards traversal from each process node. The backwards traversal defines the *inputs* to the program. Because all data entities (files, sockets, etc.) are at most 1-hop away from a process node, any value $n \geq 1$ will ensure that the entire graph is traversed in the neighborhood sampling procedure. The neighborhood is then assigned an x-value based on the start time of the process root,² and a y-value based on a monotonically increasing counter of observed neighborhoods. Future visits to neighborhoods with identical graph structures are assigned the same y-value.

Figure 1 shows an example of neighborhood identification for an excerpt of E3 Cadets using 1-hop neighborhoods. After generating the provenance graph from the audit log, process nodes are visited in chronological order and assigned a y-value (Label). If and when `sshd` creates another `bash` process with the same data inputs, this second `bash` instance

will also be assigned Label 2. Choosing an appropriate value for n is a tradeoff between over- and under-approximating the differences in process behaviors. Higher n -hop values will result in the generation of more neighborhoods that occur less frequently, while smaller values will generate fewer neighborhoods that occur more frequently. We experimented with n values 1 – 3, and determined that $n = 1$ provides the most intuitive visual snapshot of dataset contents; we report these plots throughout the remainder of the paper. In contrast, $n = 2$ provided greater differentiation between datasets for our subsequent analyses. We show an example $n = 2$ plot in Figure 6 of the Appendix.

The resulting time series for E3 Cadets is shown in Figure 2a. Each y-value indicates a unique provenance neighborhood in the dataset; a point indicates a neighborhood was observed at a given time in the engagement. We shade the times when the system was actively under attack in red. The individual neighborhoods primarily form two patterns of behavior — horizontal stripes and vertical spikes. The horizontal stripes indicate extremely frequent operating system activities; many of these stripes are likely to be default behaviors of FreeBSD, as opposed to background activity that was explicitly injected by the Cadets performers. On the other hand, the vertical spikes that occur every 24 hours indicate many different processes that were simultaneously launched at the same time, demonstrating that the background activity configured by the Cadets performers was injected procedurally by script. The same time series can also be viewed as a heatmap to better highlight extremely frequent process behaviors. For each heatmap plot, such as that of Cadets (Figure 3a), we calculate frequency by counting the number of occurrences of a given neighborhood in a 1-hour period. Using the heatmap version of the time series, it becomes clear that process-level anomaly detection results on the Cadets dataset would be skewed towards a small number of extremely high frequency process behaviors.

Handling Semantically Equivalent I/O Behavior. A commonly observed issue in the system auditing literature is that semantically equivalent behaviors may not lead to the same data accesses due to various sources of non-determinism. For example, a process being executed repeatedly may write to a temporary file containing a timestamp in the filename. To further reduce noise in our neighborhood sampling procedure, we employ Michael et al.’s method of templating file I/O into semantically equivalent groups [46]. Templates take the form of regular expressions over file paths that are generated tracing through paths by component from the basename to the root. If two paths differ by only one component, a template is created with a wildcard in this component. For example, the paths `~/ .cache/foo` and `~/ .cache/bar` would both be mapped to the template `~/ .cache/*` and be considered equivalent. We employ the same mechanism to group registry key access on Windows platforms. Unfortunately, network entities (IP addresses) are not organized in a semantically informative hierarchical structure, so this approach is not effective on network I/O. While we experimented with reverse DNS lookups of IP addresses, we found the information to be unreliable given

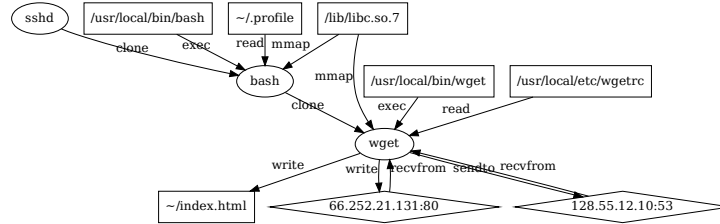
1. <https://bitbucket.org/sts-lab/reapr-ground-truth/>

2. Notice that there is a loss of precision here; the process execution time may occur much earlier than later input events to the process. However, we demonstrate that this is an effective means of visualizing large datasets.

```

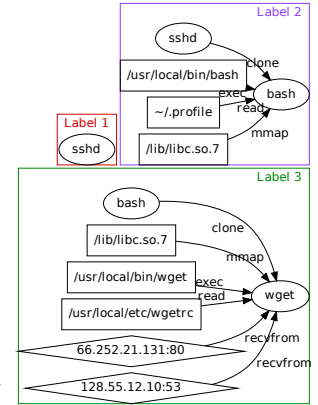
clone, t1, sshd, bash
exec, t2, bash, /usr/local/bin/bash
mmap, t3, bash, /lib/libc.so.7
read, t4, bash, ~/.profile
clone, t5, bash, wget
exec, t6, wget, /usr/local/bin/wget
mmap, t7, wget, /lib/libc.so.7
read, t8, wget, /usr/local/etc/wgetrc
sendto, t9, wget, 128.55.12.10:53
recvfrom, t10, wget, 128.55.12.10:53
write, t11, wget, 66.252.21.131:80
recvfrom, t12, wget, 66.252.21.131:80
write, t13, wget, ~/index.html

```



(a) Simplified audit log

(b) Provenance graph



(c) 1-hop neighborhoods

Figure 1: Illustrated visualization process from a simplified audit log to 1-hop neighborhoods for an isolated fragment of E3 CADETS. Note that we only label neighborhoods rooted at processes, and only consider node ancestors.

the age of some datasets. As a result, we choose to treat all calls to external IP addresses as semantically equivalent. We discuss the implications of this decision in Section 7. Additionally, we observed that long-running processes, such as services or browsers, tend to produce many neighborhoods containing subsets of the same inputs. We found that only producing one neighborhood label for the largest superset of inputs significantly reduced noise in our plots while preserving the patterns in neighborhoods.

3.2. Results

We report the provenance neighborhood plot results for all 9 of the surveyed datasets in Figure 2, with Figure 3 reporting the heatmap versions. The regular plots more clearly show spikes of co-occurring activities. On the other hand, the heatmaps indicate concentrations of high-frequency process behavior. When evaluating process-level classification tasks, the heatmaps also indicate how much each neighborhood is contributing to the denominator of key statistics. Using these visualizations, several previously unobserved phenomena in these datasets become clear:

Synthetic Workloads are Conspicuous. Through conversations with the performers, it is our understanding that all DARPA TC performers used automated workload generators during the attack engagements. The synthetic workloads are visible in these datasets, appearing as vertical spikes of process activities occurring in rapid succession. This pattern is also present in E3 FiveDirections, but is more difficult to see due to non-determinism in the paths of registry key edits. On the three *nix-based systems (E3 Cadets, Trace, Theia), most of the workload is repetitive enough to generate near-identical provenance neighborhoods on each invocation. Whether or not this syntheticity will bias threat detection results will depend on the detection system. Unfortunately, these datasets are used as all-purpose benchmarks for intrusion detection, so workloads that might unfairly advantage certain algorithms are problematic. For example, a classifier leveraging time series information might

inflate its performance by predicting that all activity that occurs during a particular hour of the day is benign.

System Tasks Dominate Background Activity. Process- and entity-level classification tasks are increasingly favored in research; they provide a more rigorous evaluation challenge and a more practical deployment model. However, the domination of *system tasks* over other process activities complicates matters. Here, we use the term system tasks to refer to activities the host engages in while it is in a quiescent state, i.e., processes that launch on their own without any form of user direction.³ These can be seen as horizontal stripes of activity that run the length of the entire time series. As can be seen in Figure 3, system tasks actually dominate the total process activity in most datasets, often occurring orders of magnitude more frequently than other process activity. As a result, any classification performance statistics that report on classification of negative samples (e.g., Acc., FPR, F1, AUC, etc.) disproportionately reflect highly monotonous system task behaviors. The ability of a classifier to correctly label repetitive system tasks as benign is indeed desirable, but these results now paint a false picture of the classifier’s ability to differentiate “noisier” observations generated by human-computer interaction (or its simulacrum). Unfortunately, this effect is present in datasets featuring human-operated workloads as well (ATLASv2).

Datasets Contain Unreported Experimental Artifacts. Visualizing provenance neighborhoods over time also highlights several experimental artifacts. The E3 Theia endpoint experienced considerable downtime over the course of the engagement. We also note that the workload generator appears to have been edited during the first and second periods of uptime. The original ATLAS attacks were collected not as a continuous stream of activity, but intermittently over the course of several months.⁴ Although the ground truth

3. The synthetic workloads are not “system tasks” for our purposes because they are intended to represent user interactions.

4. Unlike ATLAS, we elect not to use a discontinuous x-axis in the E3 Theia plot so that it was directly comparable to the other E3 engagements.

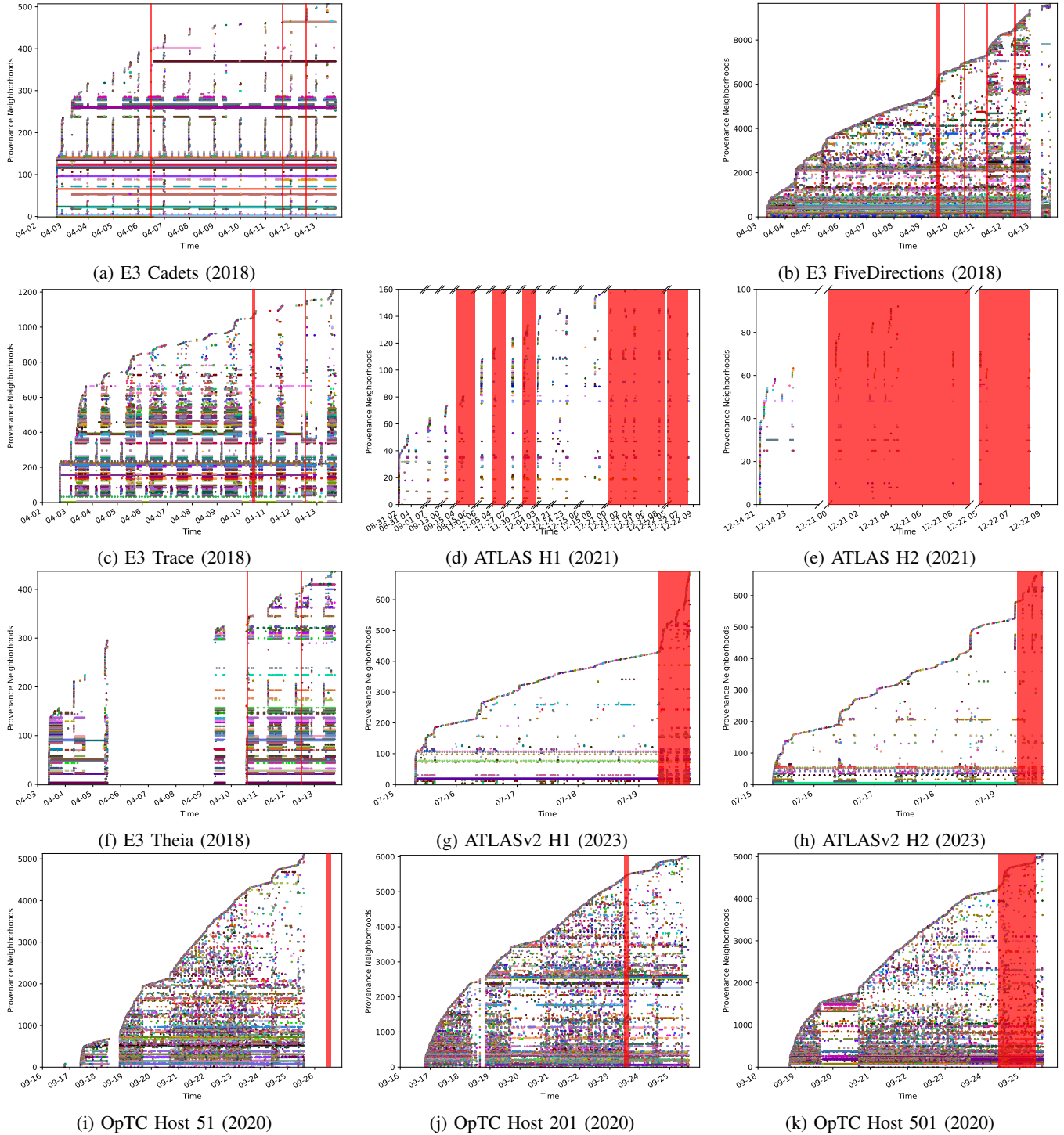


Figure 2: Provenance neighborhoods over time.

documentation for OpTC Host 51 describes the attack that was performed, there is no publicly-available data on the day of the attack itself. Other artifacts are more difficult to spot in print but can be detected through finer-grained analyses; for example, E3 Cadets suffered a brief period of downtime after the first attack, which caused the provenance neighborhood

of one system task to change for the remainder of the engagement. Similarly, the OpTC datasets show brief periods of downtime and shifts in behavior. Generating datasets is tedious and error prone, such that mistakes and oversights are bound to happen. However, these artifacts also have the potential to influence results. The Theia dataset appears

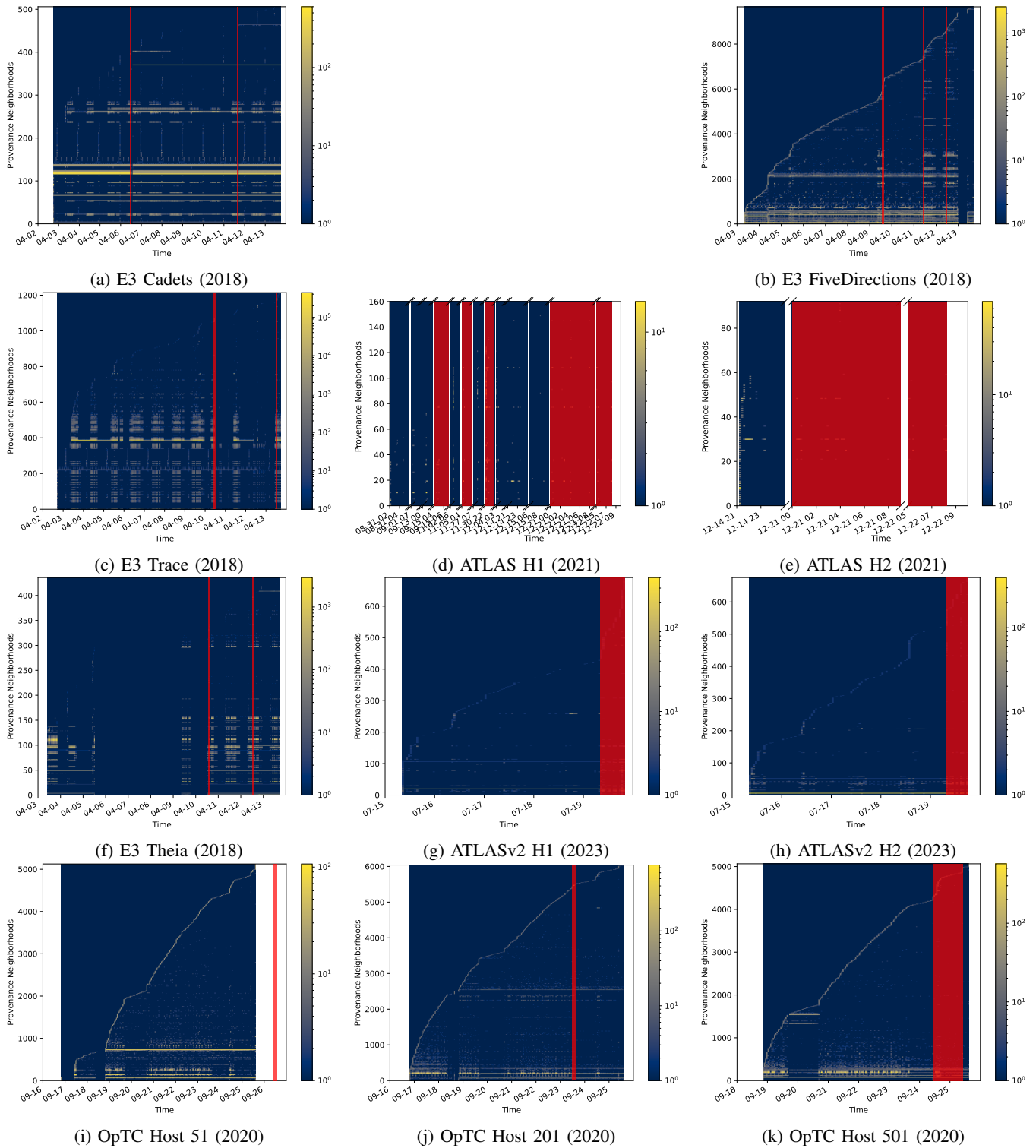


Figure 3: Provenance neighborhoods by time and frequency. Frequency is calculated in periods of 1 hour, or 10 minutes for shorter datasets (ATLAS). While the Figure 2 plots more clearly demarcate neighborhoods from one another, the heatmap versions can be used to observe the frequency of neighborhood occurrences at a given time point. Note that the heatmap scale is not standardized, but is adjusted per dataset to highlight high-frequency areas.

to be “out of distribution” with itself — does this skew results in favor of the classifier, or provide a more rigorous challenge? In Cadets, does the implicit causal relationship between the system task and the first attack phase inflate the false positive rate? How would a classifier performing time series analysis react to the ATLAS dataset? It is unlikely that any of these factors were a concern in evaluating the systems for which these datasets were originally generated. However, the datasets have taken on a life of their own since their original release. Without documenting these artifacts, it is not possible for future users to understand and interpret classification results.

4. How Synthetic is the Background Activity?

Our visualization of provenance neighborhoods over time laid bare the extent to which the vertical spikes of synthetic workloads (e.g., Figure 2a) differ from the jitter of human operated workloads (e.g., Figures 2g-2h). However, the DARPA E3 generators represent an extreme example of syntheticity where all system interactions are simultaneously induced once every 24 hours. One can imagine more subtle and naturalistic generators that spread activities out randomly over the day or contain random behaviors. It would be useful to have an objective means of comparing synthetic generators and measuring their distance to a human operator baseline.

4.1. Neighborhood-Level Autocorrelation

One measure of the naturalness of background workloads is the extent to which the provenance neighborhood time series aligns with itself. We can achieve this using autocorrelation, which calculates the correlation of some series with itself at all time lags along the x -axis [8]. At time lags where the correlation is high, the occurrence of the neighborhood is predictable by previous observations, indicating this lag describes the the period of repetition. For example, in E3 Cadets, neighborhood 150 repeats 4 times per day, matching a period of 6 hours, while neighborhood 200 repeats every day, matching a period of 24 hours. The horizontal stripes, like neighborhood 110, are constantly repeating, so any period would work.

To perform autocorrelation, we first bin observations in the continuous time series of a dataset into discrete fixed-width bins. For our analysis, we use a bin width of 15 minutes. We reasoned that 15 minutes was wide enough to capture the entire burst of synthetic activity in the E3 engagements, while still being narrow enough to capture other variance in background activity. We then find the period T for every individual neighborhood’s time series via the lag t that maximizes the autocorrelation coefficient ρ , i.e., the correlation coefficient of the series with itself shifted by t . However, ρ for low t tends to be inflated by temporal locality of process activity. Small shifts of the histogram often match themselves relatively well because process activity rarely lasts for only exactly one bin, so the front end of a block of activity ends up matching against its tail end. To avoid identifying these spurious lags as the period, we restrict

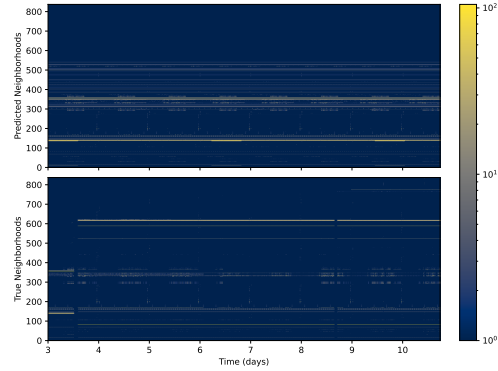


Figure 4: Autocorrelation-based prediction of the test period time series of Cadets, based only on the training data. The prediction visually replicates many of the observed traits of Cadets, such as the vertical spikes and horizontal stripes of behaviors. However, it is naturally unable to anticipate sudden changes in behaviors during the testing period, such as crashes or new neighborhoods.

T to be greater than the first t where $\rho \leq 0$. A negative ρ indicates this lag primarily causes a mismatch; e.g., the shifted data tends to be large when the original is small, and vice versa. Therefore, when ρ becomes negative, the lags t have passed the point where activities match themselves. It is also possible for T to be a multiple of the apparent true period; for example, because E3 Cadets neighborhood 200 repeats every day, we could also describe it as a pattern of two instances that repeats every 48 hours. The larger t results in fewer comparisons for the autocorrelation due to less overlap in the finite-length sequence, which can result in unexpectedly higher ρ . For our purposes, it is not a problem if we identify a multiple of the visually intuitive period.

4.2. Dataset-Level Autocorrelation

We described how to find the period of an individual neighborhood via autocorrelation. However, audit datasets are often used for learning-based methods where we train, validate, and test models on different subsets of the data, generally partitioned by time. We would like to know whether the training behavior unexpectedly differs from the test data, but it is unclear how to measure this using the autocorrelation coefficient, which requires an entire neighborhood’s time series. We initially envisioned this task as being suitable for some recurrent neural network-based autoencoder; for example, the autoencoder could be trained on neighborhood appearances in the training data and then predict appearances in the test data, with the reconstruction loss used as a syntheticity measure of the neighborhood. However, we found that our time series models were so concise that there was insufficient training data available for an autoencoder to perform well. Instead, we adapt our autocorrelation method to perform a similar task to that of the autoencoder.

We first partition the dataset chronologically into a training and test split, with the training split ending immediately

TABLE 3: Autocorrelation reconstruction loss for the testing period averaged per neighborhood (By Neighborhood) or weighted by the number of processes per neighborhood (By Process).

Dataset	Loss (By Neighborhood)	Loss (By Process)
E3 Cadets	0.570	0.578
E3 FiveDirections	0.581	0.677
E3 Theia	0.664	0.843
E3 Trace	0.511	0.257
ATLASv2 h1	0.664	0.163
ATLASv2 h2	0.607	0.290
OpTC Host 201	0.230	0.500
OpTC Host 501	0.368	0.478

prior to the first attack. We then use autocorrelation to recover the period T of each neighborhood purely within the training data. We combine the period T with the occurrences of neighborhoods in the training data to predict future occurrences in the test set. Care is taken to maintain alignment between the training and test splits by starting from the end of training and using T -hour slices. Figure 4 shows an example of our autocorrelation-based reconstruction of the testing period for E3 Cadets.

To measure the agreement of our predictions with the actual testing data, we define the reconstruction loss of a neighborhood to be the predictor’s false negative rate, calculated as follows: Let y_N be the true time series of test occurrences of neighborhood N , and $\hat{y}_N$ be the predicted time series. Then the reconstruction loss L_N across all test bins t is:

$$L_N = \frac{\sum_t \max(y_t - \hat{y}_t, 0)}{\sum_t y_t}$$

In other words, for every bin t , the reconstruction loss is minimized if the predictor correctly anticipates all test processes, with false positives (guessing more processes than present in the true test period) having no effect on the loss. This loss function would not be appropriate for a learning-based mechanism, which could minimize loss by always predicting the maximum frequency; however, our autocorrelation-based model cannot optimize in this way and defaults to a null prediction. The false negative rate as loss is very interpretable: a loss of $X\%$ means the predictor missed $X\%$ of the test processes in that particular neighborhood. We can define the overall reconstruction loss of a dataset either by neighborhood, where we take the mean loss per neighborhood, or by process, where we weigh the mean by the number of processes in each neighborhood. Table 3 shows the overall reconstruction losses for our studied datasets.

4.3. Results

Syntheticity of Background Activity is Measurable.

The reconstruction losses for these datasets in Table 3 corroborate our observations from Section 3. Due to distributional shifts, Theia is difficult to predict; the per-process loss of 84% indicates we could only correctly predict the occurrence of 16% of the processes in the test set. Trace

TABLE 4: Attack Covertiness per dataset.

Dataset	# Covert	# Conspicuous	% Covert
E3 Cadets	2793	28	99.01
E3 FiveDirections	7929	2053	79.43
E3 Theia	0	47	0.00
E3 Trace	0	86	0.00
E3 Trace-np	0	86	0.00
ATLASv2 H1	7	142	4.70
ATLASv2 H2	0	17	0.00
OpTC Host 201	5	6	45.45
OpTC Host 501	4	46	8.00

and Cadets are easier to predict, matching their apparent syntheticity, although Cadets’s loss is larger due to differences in the training and testing data. Although the individual neighborhoods of ATLASv2 are more difficult to predict, the loss per process is actually quite high due to the dominance of repetitive system tasks in these datasets. Interestingly, although OpTC is visually very chaotic, individual neighborhoods in the OpTC hosts are generally quite predictable, showing the importance of having different periods per process.

5. How Conspicuous is the Attack Activity?

A final consideration of dataset quality is how the attack activity relates to the test data. As exemplified by the “Living off the Land” threat paradigm, we expect sophisticated intrusions to appear highly similar to typical endpoint activities. To what extent are the simulated attackers in benchmark datasets living off the land? Once again, analysis of provenance neighborhoods over time provides us with a general measure for quantifying these differences.

An idealized living-off-the-land attack, through the lens of our analyses, would manifest as a set of attack processes comprised entirely of benign neighborhoods observed during training. Thus, we define a “covert” attack process as belonging to a neighborhood that also appeared in the benign training data. In contrast, a “conspicuous” attack process belongs to a neighborhood that first appears in the test data. Intuitively, we predict that conspicuous attack processes are easier for classifiers to flag as anomalous than covert attack processes.

We report the proportion of covert attack processes when using 2-hop neighborhoods in Table 4. The covertness of E3 Trace is conspicuously high. Upon further investigation, we realized that this was due to the execution partitioning [41] of `firefox`, which leads to a 100x explosion in the number of “processes” (execution units) in the dataset. Almost 20% of E3 Trace processes are Firefox execution units, and because Firefox is involved in the attack, many execution units are attack behaviors. The presence of execution units in Trace complicates matters of dataset comparisons, so we performed a transformation of Trace that merges all execution units back into the parent processes. We additionally include this non-partitioned version, Trace-np, throughout the remainder of the paper and will release it upon publication. In Trace-np,

we see that without execution partitioning, Trace becomes the least covert of the studied E3 datasets.

5.1. Results

Attack Covertiness varies across datasets. We see significant variation in attack covertiness across our studied datasets. The E3 datasets range from the highly-covert Cadets, at 99% covert, to the highly conspicuous Theia, at 11% covert. When departitioned, Trace is actually the most conspicuous, with only a single covert attack process and an overall covertiness of 1%. Finally, FiveDirections lies towards the upper end of covertiness at 78%. Although the ATLASv2 datasets were designed to retain similar background activity in the benign data collection period and attack periods, the attack is comprised primarily of neighborhoods never encountered during training. In particular, ATLASv2 H2 does not have any covert attack samples — all attack neighborhoods are first encountered during the attack period. This suggests that detecting anomalies in the ATLASv2 datasets may be particularly easy relative to the other datasets. Finally, the OpTC hosts are in the middle in terms of covertiness, both hovering around 50% covert.

6. Validating Dataset Quality Metrics

Our exploration of threat detection datasets has led to the development of three dataset quality assessments: identification of experimental artifacts (§3), Benign Synthenticity (§4), and Attack Covertiness (§5). We now set out to measure the extent to which these metrics impact threat detection evaluation results.

6.1. Experimental Set-Up

We validate our approach by benchmarking against a variety of state-of-the-art IDS, including R-CAID [29], FLASH [58], and KAIROS [11], as well as a baseline GNN. The baseline GNN is a Graph Attention Network (GAT) [59], a type of GNN that uses attention heads to weigh nodes, reducing noise. It generates embeddings for nodes summarizing their n -hop parent neighborhoods, which are similar to our provenance neighborhoods, though not exactly identical due to our heuristics. We experimentally determined that 3 layers (corresponding to 2-hop neighborhoods) and 8 attention heads performed best on our datasets, and thus use our 2-hop provenance neighborhood metrics. We embed vertex labels using `doc2vec` [40] by treating each path segment as a separate word, as in prior provenance-based systems [60]. We use 64-dimensional vectors and 100 training epochs for `doc2vec`.

For R-CAID, FLASH, and KAIROS, we reused the authors’ selected hyperparameters from the original papers. We do not perform further hyperparameter tuning on these systems; our goal is not to critique or compare these classifiers, but instead to understand how the performance of individual classifiers change as dataset characteristics are

TABLE 5: AUROC advantage when removing the artifact neighborhood from E3 Cadets. All models perform better, with most models gaining > 0.2 AUC with the experimental artifact removed.

Artifact?	GNN	R-CAID	FLASH	KAIROS
Yes	0.502	0.384	0.370	0.414
No	0.771	0.719	0.586	0.481
Gain	+0.269	+0.335	+0.216	+0.067

manipulated. Therefore, we note that classification results for prior work in our experiments may not reflect their optimized classification performance. Our baseline results for each model on each dataset is reported in Figure 5.

Our experiments make use of the following datasets: E3 Cadets, FiveDirections, Theia, Trace, and Trace-np; ATLASv2 H1 and H2; and OpTC Host 201 and Host 501. ATLAS(v1) hosts are excluded due to the lack of contiguous training data, while OpTC Host 51 is excluded due to the lack of attack data. We use the same training and test splits from our earlier analysis, i.e., the training split begins at the start of the dataset and runs until immediately prior to the first attack. The training split is further divided chronologically into training and validation at a 90%/10% ratio. Our experiments were conducted on a workstation running Ubuntu 20.04.4 LTS with a 64-core AMD Ryzen Threadripper PRO 3995WX processor, three Nvidia RTX A6000 GPUs, and 1 TiB of RAM.

6.2. Experimental Artifacts

6.2.1. Cleaning Up Artifacts in Cadets. In Section 3, we identified that the behavior of a high-volume background task appeared to have been indirectly influenced by an attack-induced machine crash. Specifically, neighborhood 1, representing `bash reading gather_stats.sh` and `kernel_mem.txt`, disappears after the machine crash, replaced by neighborhood 369, which is `bash reading` from a file descriptor for a file missing from the logs, presumably due to loss from the crash. This results in 100,929 benign test processes that do not appear at all in the training data. The new neighborhood does not have a causal link to the attack behavior, and thus cannot be said to be malicious, but is clearly anomalous due to its absence in the training data. Fortunately, our analysis provides an easy way to clean up the Cadets test data — simply cull the anomalous benign neighborhood from evaluation. The impact on classification results is shown in Table 5, reported as change in AUC for classifier ROC curves. With 3 of the 4 classifiers improving by $+0.2$ AUROC or more following the removal of this neighborhood, it is clear that this individual process behavior was dramatically impacting the classification performance of Cadets evaluations. We use this “cleaned up” Cadets for the remainder of our experiments.

6.2.2. Distributional Shifts in Theia. We also observed prolonged downtime in the E3 Theia dataset, with a pos-

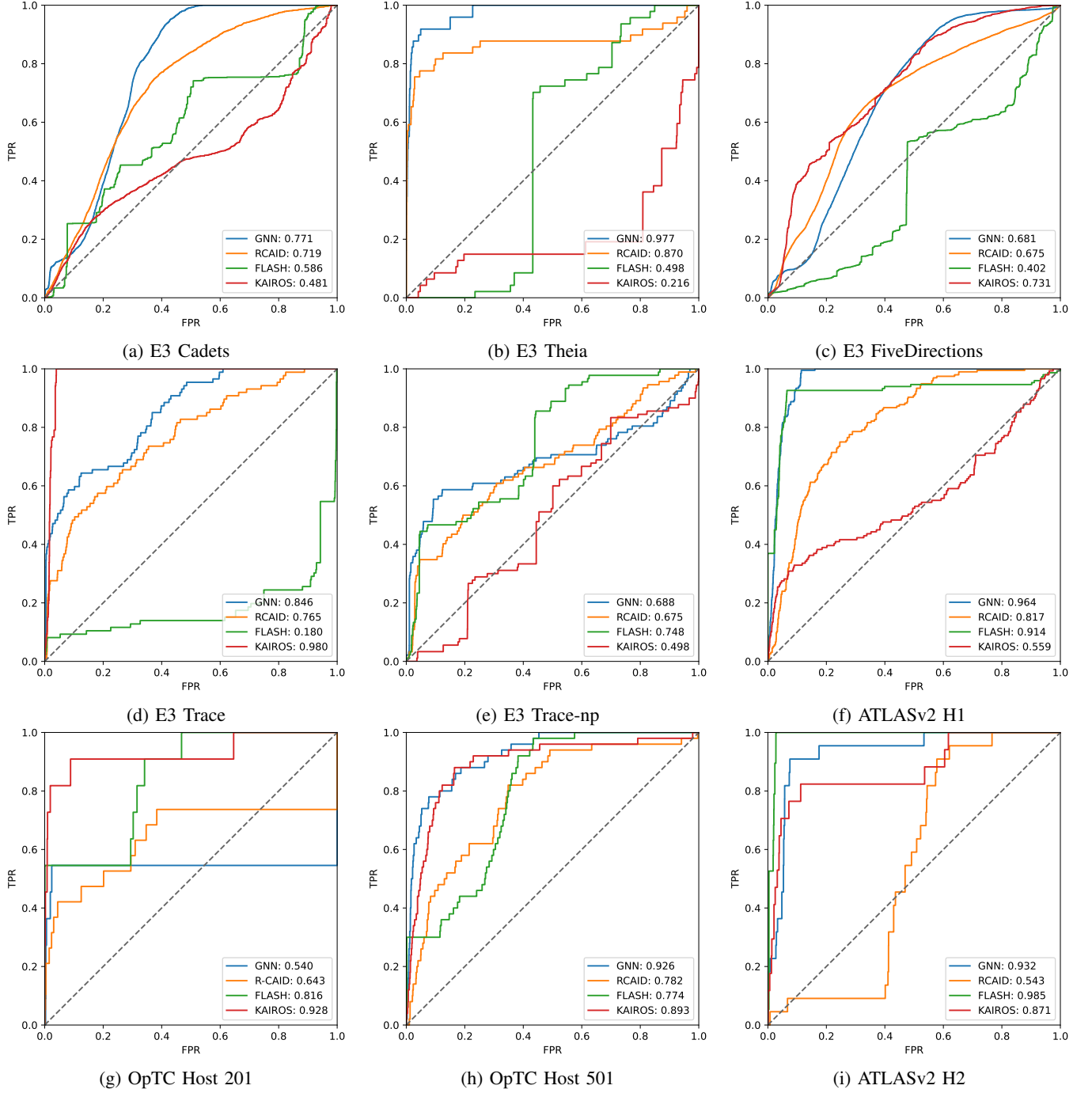


Figure 5: Baseline ROC curves for each dataset. We also report the AUC (Area Under the ROC Curve) for each model.

sible reconfiguration of the workload generator that led to distributional shifts in background activity between the 3 periods of uptime. Using the GNN, we now consider the extent to which these kinds of previously-invisible changes in distribution could bias classification results. To do so, we first divide Theia into the three periods of machine uptime visible in Figure 2f: T1 (Apr. 3–6), T2 (Apr. 9–10) and T3 (Apr. 10–14). We then manipulate whether the GNN is

trained and tested on (non-overlapping) data from the same time period or different time periods. Training data from each time slice is comprised of all benign samples from a 24 period from April 3–4 (T1), April 9–10 (T2), and April 11–12 (T3). When testing a time period against itself, we use the remainder of the time period; because T2 is only ~24 hours, the validation and test splits are the same. Finally, every test includes all the attack samples of the entire dataset.

TABLE 6: Effect of varying the train/test time periods on E3 Theia classification results. The performance over all non-training data is reported under “All”, and the difference per time period from this AUROC is reported under “Diff.”

Train	Test period						All AUC
	T1 (Apr. 3–6) AUC	Diff.	T2 (Apr. 9–10) AUC	Diff.	T3 (Apr. 10–14) AUC	Diff.	
T1	0.955	+0.088	0.833	-0.034	0.856	-0.011	0.867
T2	0.750	-0.056	0.977*	+0.171	0.838	+0.032	0.806
T3	0.742	-0.100	0.912	+0.070	0.915	+0.073	0.842

* Due to the short duration of T2, the validation set was reused for test.

TABLE 7: AUROC advantage when evaluating on the most synthetic benign behaviors versus the least synthetic. The GNN and R-CAID both exhibit a > 0.7 AUC advantage average across all datasets.

Dataset	GNN	R-CAID	FLASH	KAIROS
E3 Cadets	+0.534	+0.436	-0.152	-0.115
E3 FiveDirections	-0.024	-0.003	+0.171	-0.050
E3 Theia	+0.040	-0.023	-0.039	+0.049
E3 Trace	-0.113	-0.078	-0.132	+0.304
E3 Trace-np	-0.064	+0.017	+0.061	-0.015
ATLASv2 H1	+0.141	+0.385	+0.073	-0.273
ATLASv2 H2	+0.088	+0.635	+0.018	-0.003
OpTC Host 201	+0.006	+0.052	+0.125	+0.003
OpTC Host 501	+0.100	+0.206	-0.091	+0.044
Average	+0.079	+0.181	+0.004	-0.006

The results of this experiment are tabulated in Table 6, with the highest AUROC per time period written in green. For each training time period, testing against data from the same time period provided the most favorable result. The largest differences are seen between T1 and T2/T3, likely indicating that the workload reconfiguration happened prior to the start of T2. In comparison to testing against all data, AUROC for T1 increases from 0.867 to 0.955 (+0.09), T2 increases from 0.806 to 0.977 (+0.17), and T3 increases from 0.842 to 0.915 (+0.07). When replicating this experiment on equivalent time periods in E3 Trace, which does not contain these artifacts, we find minimal variation in AUROC. Unfortunately, it is not common practice in the literature to document experiments to this level of detail, creating the possibility that two different algorithms evaluated on the E3 Theia dataset are reporting incomparable results.

6.3. Benign Syntheticity

Moving on, we set out to determine the impact of high volumes of synthetic process behaviors on entity-level threat detection. We hypothesize that models will classify highly synthetic behaviors as benign more easily than noisier behaviors, and thus a higher concentration of synthetic behaviors in a dataset could inflate overall classification performance. To test this, we manipulate the composition of the test set for each classifier and dataset. While all tests contains all attack samples, we sort the benign samples by benign syntheticity and then perform *equal area binning* to create multiple benign test splits of roughly the same

TABLE 8: AUROC advantage when evaluating only on conspicuous attack processes versus covert attack processes. All models perform better on conspicuous attack processes.

Dataset	GNN	R-CAID	FLASH	KAIROS
E3 Cadets	+0.184	+0.179	+0.104	+0.159
E3 FiveDirections	+0.110	+0.120	+0.097	+0.101
E3 Theia*	-	-	-	-
E3 Trace*	-	-	-	-
E3 Trace-np*	-	-	-	-
ATLASv2 H1	-0.004	-0.050	+0.135	+0.293
ATLASv2 H2*	-	-	-	-
OpTC Host 201	+0.990	+0.147	+0.197	-0.127
OpTC Host 501	-0.072	+0.018	-0.049	-0.068
Average	+0.242	+0.083	+0.061	+0.072

* No covert attack data

number of samples. This equal area binning is best-effort due to variance in both the amount of nodes with a particular loss and the number of nodes that fall within specific loss ranges. Because at least one loss value contains 20% or more of the samples in each dataset, we set the target bin area to 20%, ideally resulting in five bins, although datasets with too few distinct loss values can have fewer bins.

The performance of models on these varying benign subsets is reported in Table 7, which reports the AUROC difference between the most and least synthetic bins. The GNN and R-CAID both perform markedly better on synthetic data, averaging a +0.07 or higher AUROC advantage, while FLASH and KAIROS exhibit a much smaller difference. The notable AUROC advantage exhibited by GNN and R-CAID validates our hypothesis that synthetic neighborhoods can inflate classification performance. On the other hand, the variance in advantage between classifiers suggests that certain classification algorithms may be more robust to noisy benign behaviors than others. This suggests a valuable new point of comparison when attempting to contrast prior work from novel threat detection algorithms.

6.4. Attack Covertiness

We identified covert attack processes based on whether the neighborhood appears in the training data or not. We hypothesize that covert attack processes will be harder to classify than conspicuous processes which never appear during training, as an unseen process is by definition anomalous. To test this hypothesis, we now invert the procedure from our previous experiment, keeping the benign test samples fixed and manipulating the test set. Specifically, we divide the attack samples between the set of neighborhoods that appear in training and the set that does not appear in training. We predict AUROC to increase when evaluating against conspicuous neighborhoods that do not appear in training.

The results of this experiment are reported in Table 8. Overall, we see that most models perform significantly better on conspicuous nodes as expected, with the GNN, R-CAID, and KAIROS all increasing in AUROC by at least 0.06. The GNN is most affected by covertiness, while

FLASH, KAIROS, and R-CAID are all more robust to covert attacks. In particular, the GNN completely fails to classify the covert attack processes in OpTC Host 201, while it can easily classify the conspicuous attack nodes. In contrast, the state-of-the-art IDS systems are more capable of detecting the covert attack behaviors for this dataset, though most are still advantaged. While the finding that unseen attack neighborhoods are easier to classify is not surprising, it does highlight an important consideration of dataset design that to date has gone unexplored.

7. Discussion

Our holistic investigation into the contents of endpoint threat detection datasets has revealed that there is much more to logs than the sophistication of the attack behavior. Our visualization of datasets by provenance neighborhoods over time (§3) has uncovered a variety of undocumented experimental artifacts. Our syntheticity metric (§4) provides a general solution for understanding how the sophistication of background workloads informs threat classification results. By considering the relationship between provenance neighborhoods that appear in training and test splits (§5), we provide a means of measuring the covertness of an attack. We now consider further implications of our findings.

7.1. Guided Evaluation of FLASH

We have demonstrated the potential for today’s datasets to produce misleading pictures of classification performance. Rather than discard these datasets altogether, we recommend that authors use them to explore performance across a continuum of scenarios, considering data falling into various covertness and syntheticity thresholds. As an example, we will walk through a hypothetical evaluation of FLASH. We choose FLASH because it performs well on many datasets, and we are not its authors.

First, we consider FLASH’s performance on E3 Trace. Notably, FLASH performs poorly on the original Trace, which is distinct for the execution-partitioned Firefox processes found in the dataset. Because FLASH builds node embeddings using all edges incident to a node, we hypothesize that its performance on non-partitioned data improves because of an increase in the number of incident edges, allowing FLASH to generate richer embeddings. When evaluating on the de-partitioned Trace dataset where all execution units are merged back into their parent process, we can see that FLASH outperforms RCAID and KAIROS. Due to the lack of availability of other execution partitioned datasets, we leave further exploration of this limitation to future work.

Moving on, we consider FLASH’s performance on covert attack behaviors as compared to conspicuous attack behaviors. However, Trace dataset does not contain any strictly covert attack samples because information flows from the attacker contaminate all other Firefox execution units. Instead, we use E3 Cadets, dividing the malicious attack samples between covert (previously observed processes) and conspicuous (previously unobserved) attack behaviors. FLASH is

able to detect the majority of malicious samples in both the conspicuous (AUC 0.689) and covert (AUC 0.585) samples, which is sufficient to detect each incident in the dataset. Further, when comparing against baseline systems, FLASH reports a lower conspicuous attack advantage, indicating its detection performance is less dependent on conspicuous behaviors.

Concluding our investigation, we use consider the extent to which FLASH’s performance depends on the syntheticity of evaluation data. For each dataset, we sort partition process samples into five sets based on their neighborhood’s reconstruction loss, then compare performance between the most and least lossy sets. Across all datasets, we find that FLASH’s average AUROC advantage on highly synthetic data is just +0.004, as compared to the baseline GNN’s +0.079 and R-CAID’s +0.181. This suggests that FLASH, along with Kairos (-0.006), offer more consistent classification performance that is somewhat more likely to generalize to naturalistic data streams. Thus, while FLASH is outperformed by baselines for some datasets, we conclude that it offers the most stable classification performance among state-of-the-art systems.

7.2. Findings and Recommendations

Synthetic Data can Inflate Anomaly Detection Performance. While collecting large-scale, natural benign data is challenging, we demonstrate that synthetic or otherwise predictable behaviors result in inflated anomaly detection performance. There is a clear disconnect between the popularity of anomaly detection systems in academia as opposed to industry, where rule-based detection is favored. This disconnect may in part stem from the lack of naturalism in datasets used to evaluate academic solutions; we may be wildly underestimating the false positive rates we should expect from noisy real-world activity. While we do not have a solution to data availability, our analyses permit researchers to situate their classification results in the context of data syntheticity; for example, we observed that FLASH’s performance tended to generalize across syntheticity levels better than R-CAID. In future work, we hope to partner with security operations centers in order to build a model that will allow us to extrapolate expected real-world classification performance from public dataset evaluations.

Recommendations for Dataset Producers. Researchers will continue to generate new datasets that will allow them to explore new questions in threat detection research. Our first recommendation to dataset producers is to better document the totality of activity within the dataset, not just the attacks. This includes, but is not limited to, the techniques presented in this work, which are straightforward to apply and will be open sourced upon publication. We also recommend the inclusion of design rationale — what is the workload intended to represent, and to what extent is it co-designed with the threat behaviors to create a challenging detection scenario? We are encouraged by Riddle et al.’s deliberate overlapping of attack and benign programs in ATLASv2 [51] and Alsaheel et al.’s documentation of background event counts in the original ATLAS work [4]. Finally, producers need to be transparent

about experimental artifacts, which are unavoidable in the creation of benchmark datasets.

Recommendation for Dataset Consumers. Given the scarcity of public data for threat detection research, it is difficult to be picky. Nonetheless, we recommend that experimenters approach datasets with more skepticism. Verifying the contents of datasets as an independent third party, as we have done here, will not only improve the interpretation of classification results, but may also inform dataset selection or even classifier design: the ATLAS datasets are poorly suited to one-class anomaly detection research due to the scarcity of clean benign training data; a detection algorithm leveraging time series data may need to avoid E3 Theia or ATLAS due to their discontinuities; a detection algorithm designed for *nix systems, favored by researchers, must explicitly design for the noise of registry key accesses if evaluating against E3 FiveDirections. Further, experimenters should more carefully document their experimental setups dataset-by-dataset. For example, we highlight how invisible experimenter choices can influence classification results for E3 Theia and Cadets. By extension, we recommend that readers interpreting classification results be aware of the influence of dataset quality and expect more rigorous documentation from research papers.

Finally, we note several limitations of the present study: **Imprecisions in Behavior Modeling.** Decisions made in our neighborhood analysis can mask variance that may actually prove useful to threat detection tasks. Specifically, we mask all IP addresses in network events due to the difficulty of identifying semantically related network accesses, and cluster accesses to similar file paths. Further, as these clustering steps largely grouped low-frequency events together, a classification algorithm may have learned that they provided limited information anyway. One could take the approximation of behaviors even further, e.g., by leveraging machine learning to learn behaviors in an audit dataset [63] or using more sophisticated clustering models [62]. Through iterative experimentation, we concluded that our heuristics struck the right balance between ease of interpretation and faithfulness to the raw event stream.

Measuring Naturalism via Periodicity. We used periodicity, measured via autocorrelation, as a proxy for the naturalism of a dataset. A dataset could easily be both highly synthetic and yet not as obviously periodic simply by introducing variable delays in the workload generator. Our intention is not to suggest non-periodic datasets are necessarily better, nor that datasets cannot be periodic; instead, we show that regularity in the dataset can simplify anomaly detection, and thus artificially inflate classification results. Furthermore, as we discovered in human-generated ATLASv2, periodic patterns also exist in real behaviors.

8. Related Work

Intrusion detection research is at least 40 years old, when Denning and Neumann discussed various anomalous behaviors that could be indicative of an attack [17]. Forrest et al. later broadened the scope to any unusual process behavior,

[23], with recent systems leveraging more sophisticated machine learning models [18], [19], [54]. Provenance-based IDS systems use the relationships between system objects to further inform detections, operating on causal paths [60], information flow via tag propagation [35], [36], or subgraphs [32], [44], [64], and possibly including domain knowledge of typical attack behaviors [33], [47]. Many of these systems are evaluated on the datasets we survey in this work.

Prior studies on Network IDS datasets have raised similar concerns about the realism of data and the similarity of benign and attack activity [9], [43], [45], [56]. Anomaly detection systems are already fundamentally disadvantaged by only training on benign samples [55], yet good, labeled data are sorely lacking due to sensitivity concerns and difficulty in curation [2], [6]. Sensitivity concerns are warranted by deanonymization attacks [48] which have even been demonstrated on anonymized Network IDS datasets [12]. Furthermore, researchers that generate datasets rarely release them for public use [65]. Many of these datasets are synthetic for ease of creation and use, but accurately modeling real data, such as the internet, is known to be challenging [22]. To our knowledge, we are the first to provide methods for measuring these problems in the endpoint threat detection space.

Researchers in digital forensics have identified an analogous lack of high-quality datasets [7], [25], [26], [28], [30], [61]. Many tools have been created to generate datasets with minimal artifacts [10], [50]. Generated forensic datasets have exposed issues with forensic tools, demonstrating the need for more data [49], [52]. However, the degree to which results on synthetic datasets translate to real operation is still unknown [27], [34].

While there has been significant discourse on dataset quality in Network IDS and digital forensics research, we found limited work that applies to Endpoint IDS. The host data of early DARPA datasets has been criticized, but only from the lens of system call sequences [15]. We have found surveys that discuss potential pitfalls in commonly used audit datasets [66], but with limited analysis on their potential ramifications on performance. Anjum and Hamelin analyze the distribution of types of events and the proportion of attack samples in OpTC, but do not analyze the actual behaviors therein [5]. Landauer et al. recently generated a synthetic dataset usable to evaluate IDS, including a brief comparison of their data with real data [39]. We were unaware of their openly available data while conducting our investigation.

9. Conclusion

We develop and present methods for visualizing and measuring phenomena in endpoint threat detection datasets. Using these techniques, we identify previously undocumented artifacts in popular datasets and consider their implications on classification results. These findings are a step towards elevating the rigor and transparency of threat detection research.

Availability

Our ground truth labels are available at <https://bitbucket.org/sts-lab/reapr-ground-truth/>. Results from this work, as well as our analysis scripts, will be added to the repository to provide expanded characterization of the labeled datasets.

Acknowledgments

We would like to thank the (many) anonymous reviewers who reviewed our submissions for their helpful feedback to improve the paper. This work was supported in part by NSF CNS, award contract 2055127.

References

- [1] Operationally transparent cyber (optc) data release. <https://github.com/FiveDirections/OpTC-data>, 2020.
- [2] Sebastian Abt and Harald Baier. Are we missing labels? a study of the availability of ground-truth in network security research. In *2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, pages 40–55, 2014.
- [3] Abdullellah Alsaheel, Yuhong Nan, Shiqing Ma, Le Yu, Gregory Walkup, Z. Berkay Celik, Xiangyu Zhang, and Dongyan Xu. ATLAS: A sequence-based learning approach for attack investigation. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3005–3022. USENIX Association, August 2021.
- [4] Abdullellah Alsaheel, Yuhong Nan, Shiqing Ma, Le Yu, Gregory Walkup, Z. Berkay Celik, Xiangyu Zhang, and Dongyan Xu. ATLAS: A sequence-based learning approach for attack investigation. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3005–3022. USENIX Association, August 2021.
- [5] Md. Monowar Anjum, Shahrear Iqbal, and Benoit Hamelin. Analyzing the usefulness of the darpa optc dataset in cyber threat detection research. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies, SACMAT '21*, page 27–32, New York, NY, USA, 2021. Association for Computing Machinery.
- [6] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. Dos and don'ts of machine learning in computer security. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3971–3988, Boston, MA, August 2022. USENIX Association.
- [7] Frank Breiting and Alexandre Jotterand. Sharing datasets for digital forensic: A novel taxonomy and legal concerns. *Forensic Science International: Digital Investigation*, 45:301562, 2023.
- [8] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer International Publishing, 2016.
- [9] Carson Brown, Alex Cowperthwaite, Abdulrahman Hijazi, and Anil Somayaji. Analysis of the 1999 darpa/lincoln laboratory ids evaluation data with netadhdct. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–7, 2009.
- [10] Alberto A. Ceballos Delgado, William B. Glisson, George Grispos, and Kim-Kwang Raymond Choo. Fade: A forensic image generator for android device education. *WIREs Forensic Science*, 4(2):e1432, 2022.
- [11] Zijun Cheng, Qiujian Lv, Jinyuan Liang, Yan Wang, Degang Sun, Thomas Pasquier, and Xueyuan Han. Kairos: Practical Intrusion Detection and Investigation using Whole-system Provenance. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3533–3551, Los Alamitos, CA, USA, May 2024. IEEE Computer Society.
- [12] Scott E. Coull, Charles V. Wright, Fabian Monrose, Michael P. Collins, and Michael K. Reiter. Playing devil's advocate: Inferring sensitive information from anonymized network traces. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2007, San Diego, California, USA, 28th February - 2nd March 2007*. The Internet Society, 2007.
- [13] Gideon Creech. *Developing a high-accuracy cross platform Host-Based Intrusion Detection System capable of reliably detecting zero-day attacks*. PhD thesis, 2014.
- [14] Gideon Creech and Jiankun Hu. Generation of a new ids test dataset: Time to retire the kdd collection. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 4487–4492, 2013.
- [15] Gideon Creech and Jiankun Hu. A semantic approach to host-based intrusion detection systems using contiguous and discontinuous system call patterns. *IEEE Transactions on Computers*, 63(4):807–819, 2014.
- [16] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, and M. A. Zissman. Evaluating Intrusion Detection Systems Without Attacking Your Friends: The 1998 DARPA Intrusion Detection Evaluation. Technical Report ADA526274, Defense Technical Information Center, Jan 1998.
- [17] Dorothy Denning and Peter G. Neumann. *Requirements and Model for IDES — a Real-Time Intrusion-Detection Expert System*, volume 8. SRI International Menlo Park, 1985.
- [18] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17*, page 1285–1298, New York, NY, USA, 2017. Association for Computing Machinery.
- [19] Thijs van Ede, Hoojat Aghakhani, Noah Spahn, Riccardo Bortolameotti, Marco Cova, Andrea Continella, Maarten van Steen, Andreas Peter, Christopher Kruegel, and Giovanni Vigna. Deepcase: Semi-supervised contextual analysis of security events. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 522–539, 2022.
- [20] Pengcheng Fang, Peng Gao, Changlin Liu, Erman Ayday, Kangkook Jee, Ting Wang, Yanfang (Fanny) Ye, Zhuotao Liu, and Xusheng Xiao. Back-Propagating system dependency impact for attack investigation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2461–2478, Boston, MA, August 2022. USENIX Association.
- [21] H. H. Feng, O. M. Kolesnikov, P. Fogla, W. Lee, and Weibo Gong. Anomaly detection using call stack information. In *2003 Symposium on Security and Privacy, 2003.*, pages 62–75, May 2003.
- [22] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, 2001.
- [23] S. Forrest, S.A. Hofmeyr, A. Somayaji, and T.A. Longstaff. A sense of self for unix processes. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 120–128, 1996.
- [24] Stephanie Forrest. Computer Immune Systems Data Sets. <https://web.archive.org/web/20210224170422/https://www.cs.unm.edu/~immsec/data/synth-sm.html>, 1996.
- [25] Simson Garfinkel. Forensic corpora: a challenge for forensic research. *Electronic Evidence Information Center*, 2007.
- [26] Simson Garfinkel, Paul Farrell, Vassil Roussev, and George Dinolt. Bringing science to digital forensics with standardized forensic corpora. *Digital Investigation*, 6:S2–S11, 2009. The Proceedings of the Ninth Annual DFRWS Conference.
- [27] Thomas Göbel, Harald Baier, and Frank Breiting. Data for digital forensics: Why a discussion on “how realistic is synthetic data” is dispensable. *Digital Threats*, 4(3), oct 2023.
- [28] Patrik Gonçalves, Klara Dološ, Michelle Stebner, Andreas Attenberger, and Harald Baier. Revisiting the dataset gap problem – on availability, assessment and perspective of mobile forensic corpora. *Forensic Science International: Digital Investigation*, 43:301439, 2022.

- [29] Akul Goyal, Gang Wang, and Adam Bates. R-CAID: Embedding Root Cause Analysis within Provenance-based Intrusion Detection. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3515–3532, Los Alamitos, CA, USA, May 2024. IEEE Computer Society.
- [30] Cinthya Grajeda, Frank Breiteringer, and Ibrahim Baggili. Availability of datasets for digital forensics – and what is missing. *Digital Investigation*, 22:S94–S105, 2017.
- [31] John Griffith, Derrick Kong, Armando Caro, Brett Benyo, Joud Khoury, Timothy Upthegrove, Timothy Christovich, Stanislav Ponomorov, Ali Sydney, Arjun Saini, Vladimir Shurbanov, Christopher Willig, David Levin, and Jack Dietz. Scalable transparency architecture for research collaboration (starc)-darpa transparent computing (tc) program. Technical report, Raytheon BBN Technologies Corp. Cambridge United States, 2020.
- [32] Xueyuan Han, Thomas F. J.-M. Pasquier, Adam Bates, James Mickens, and Margo I. Seltzer. Unicorn: Runtime provenance-based detector for advanced persistent threats. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.
- [33] Wajih Ul Hassan, Adam Bates, and Daniel Marino. Tactical provenance analysis for endpoint detection and response systems. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1172–1189, 2020.
- [34] Graeme Horsman and James R. Lyle. Dataset construction challenges for digital forensics. *Forensic Science International: Digital Investigation*, 38:301264, 2021.
- [35] Md Nahid Hossain, Sadeq M. Milajerdi, Junao Wang, Birhanu Eshete, Rigel Gjomemo, R. Sekar, Scott Stoller, and V.N. Venkatakrishnan. SLEUTH: Real-time attack scenario reconstruction from COTS audit data. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 487–504, Vancouver, BC, August 2017. USENIX Association.
- [36] Md Nahid Hossain, Sanaz Sheikhi, and R. Sekar. Combating dependence explosion in forensic analysis using alternative tag propagation semantics. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1139–1155, 2020.
- [37] DARPA I2O. Transparent computing engagement 5 data release. <https://github.com/darpa-i2o/Transparent-Computing>, 2020.
- [38] Angelos D. Keromytis. Transparent computing engagement 3 data release. <https://github.com/darpa-i2o/Transparent-Computing>, 2018.
- [39] Max Landauer, Florian Skopik, Maximilian Frank, Wolfgang Hotwagner, Markus Wurzenberger, and Andreas Rauber. Maintainable log datasets for evaluation of intrusion detection systems. *IEEE Transactions on Dependable and Secure Computing*, 20(4):3466–3482, 2023.
- [40] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.
- [41] Kyu Hyung Lee, Xiangyu Zhang, and Dongyan Xu. High accuracy attack provenance via binary-based execution partition. In *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*. The Internet Society, 2013.
- [42] Richard Lippmann, Joshua W Haines, David J Fried, Jonathan Korba, and Kumar Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 2000. Recent Advances in Intrusion Detection Systems.
- [43] Matthew V. Mahoney and Philip K. Chan. An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In Giovanni Vigna, Christopher Kruegel, and Erland Jonsson, editors, *Recent Advances in Intrusion Detection*, pages 220–237, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [44] Emaad Manzoor, Sadeq M. Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 1035–1044, New York, NY, USA, 2016. Association for Computing Machinery.
- [45] John McHugh. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur.*, 3(4):262–294, nov 2000.
- [46] Noor Michael, Jaron Mink, Jason Liu, Sneha Gaur, Wajih Ul Hassan, and Adam Bates. On the forensic validity of approximated audit logs. In *Annual Computer Security Applications Conference, ACSAC '20*, pages 189–202, New York, NY, USA, 2020. Association for Computing Machinery.
- [47] Sadeq M. Milajerdi, Rigel Gjomemo, Birhanu Eshete, R. Sekar, and V.N. Venkatakrishnan. Holmes: Real-time apt detection through correlation of suspicious information flows. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1137–1152, 2019.
- [48] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, 2008.
- [49] Sebastian Nemetz, Sven Schmitt, and Felix Freiling. A standardized corpus for sqlite database forensics. *Digital Investigation*, 24:S121–S130, 2018.
- [50] Jungheum Park. Trede and vmpop: Cultivating multi-purpose datasets for digital forensics – a windows registry corpus as an example. *Digital Investigation*, 26:3–18, 2018.
- [51] Andy Riddle, Kim Westfall, and Adam Bates. Atlasv2: Atlas attack engagements, version 2, 2023.
- [52] Sven Schmitt. Introducing anti-forensics to sqlite corpora and tool testing. In *2018 11th International Conference on IT Security Incident Management & IT Forensics (IMF)*, pages 89–106, 2018.
- [53] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S P 2001*, pages 144–155, May 2001.
- [54] Yun Shen, Enrico Mariconti, Pierre Antoine Vervier, and Gianluca Stringhini. Tiresias: Predicting security events through deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18*, page 592–605, New York, NY, USA, 2018. Association for Computing Machinery.
- [55] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316, 2010.
- [56] Mahbod Tavallae, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6, 2009.
- [57] Jacob Torrey. Transparent computing engagement 5 data release. <https://github.com/darpa-i2o/Transparent-Computing>, 2020.
- [58] Mati Ur Rehman, Hadi Ahmadi, and Wajih Ul Hassan. Flash: A Comprehensive Approach to Intrusion Detection via Provenance Graph Representation Learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3552–3570, Los Alamitos, CA, USA, May 2024. IEEE Computer Society.
- [59] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.
- [60] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl A. Gunter, and Haifeng Chen. You are what you do: Hunting stealthy malware via data provenance analysis. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.
- [61] Kam Woods, Christopher Lee, Simson Garfinkel, David Dittrich, Adam Russell, and Kris Kearton. Creating realistic corpora for security and forensic education. *Proceedings of the ADFSL Conference on Digital Forensics Security and Law*, 01 2011.

- [62] Zhiqiang Xu, Pengcheng Fang, Changlin Liu, Xusheng Xiao, Yu Wen, and Dan Meng. Depcomm: Graph summarization on system audit logs for attack investigation. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 540–557, 2022.
- [63] Jun Zeng, Zheng Leong Chua, Yinfang Chen, Kaihang Ji, Zhenkai Liang, and Jian Mao. WATSON: abstracting behaviors from audit logs via aggregation of contextual semantics. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.
- [64] Jun Zengy, Xiang Wang, Jiahao Liu, Yinfang Chen, Zhenkai Liang, Tat-Seng Chua, and Zheng Leong Chua. Shadewatcher: Recommendation-guided cyber threat analysis using system audit records. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 489–506, 2022.
- [65] Muwei Zheng, Hannah Robbins, Zimo Chai, Prakash Thapa, and Tyler Moore. Cybersecurity research datasets: Taxonomy and empirical analysis. In *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)*, Baltimore, MD, August 2018. USENIX Association.
- [66] Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. Provenance-based intrusion detection systems: A survey. *ACM Comput. Surv.*, 55(7), dec 2022.

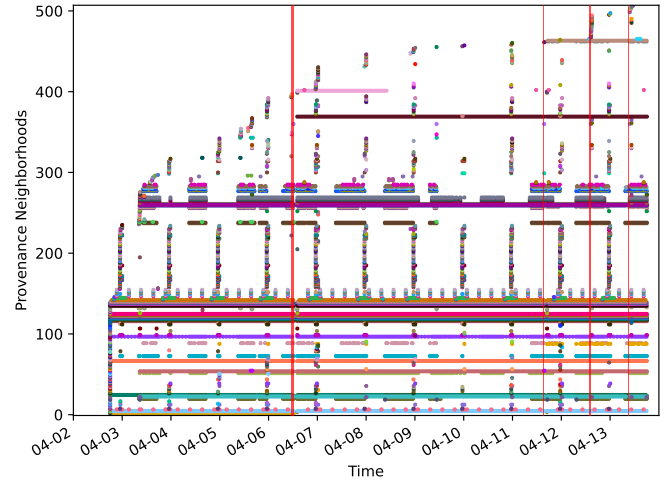
Appendix A. Dataset Descriptions

Early IDS work (e.g., [21], [23], [53]) evaluated against the UNM datasets that provide syscall traces for specific applications (e.g., `ftp`, `httpd`); unfortunately, they are not widely applicable today because only the syscall identifiers are reported in the dataset, not the arguments, preventing provenance-based approaches from using them. The DARPA'98 and DARPA'99 intrusion detection datasets were generated by MIT Lincoln Laboratory and retain comprehensive syscall information, but have fallen out of favor due to their age. A portion of the DARPA'98 network logs comprised the KDD Cup dataset but did not include syscall data. The Australian Defence Force Academy (ADFA) datasets contain attack simulations for Linux (LD) and Windows (WD) platforms, but do not retain syscall arguments except for a smaller addendum of stealth attacks (WD-SAA).

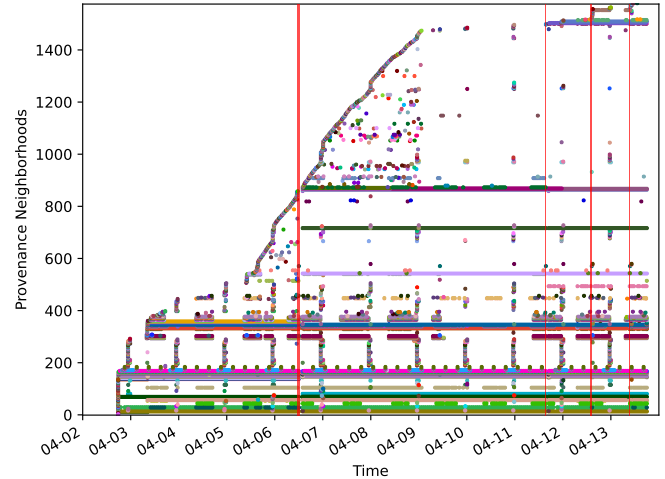
Commonly used datasets today begin with Manzoor et al.'s StreamSpot dataset [44], which despite its age and size remains in use due to the difficulty of differentiating benign from attack entities in the highly similar web browsing sessions contained therein. The first public release from the DARPA Transparent Computing Program, Engagement 3 (E3), provides a 14-day attack engagement conducted in April 2018 [31], [38]. Logs reported using a common data model from 4 different traditional computing endpoints are available. The engagement describes the penetration efforts of a professional red team that emulated the activities of a sophisticated threat group. An additional major dataset, Engagement (E5), was executed under similar conditions facing a more sophisticated version of the threat group. The DARPA Operationally Transparent Cyber (OpTC) program continued forward with the FiveDirections performers from TC, who deploy a 500 node Windows testbed that is attacked by another red team.

Finally, Alsaheel et al. release the attack engagements used in the evaluation of their ATLAS investigation system [3]. ATLAS is comprised of 10 attack chains including both single-host and two-host attacks, although the steps in the chains partially overlap. A notable limitation of the original ATLAS dataset was the lack of benign activity; while limited benign activity was included, it does not follow a realistic usage profile and only accounts for roughly 50% of the logs. To extend the uses of the dataset, Riddle et al. conducted a second engagement (ATLASv2) in which the attacks were conducted against a backdrop of naturalistic activity generated by human users [51].

Appendix B. n -hop Neighborhoods



(a) $n = 1$



(b) $n = 2$

Figure 6: Comparison of provenance neighborhood plots with $n = 1$ hops as compared to $n = 2$ hops for E3 Cadets. As expected, the use of 2-hop neighborhoods results in more unique neighborhoods; however, the same patterns emerge regardless of n . We show $n = 1$ figures because they are slightly easier to interpret, but our analysis results are based on $n = 2$ throughout the paper.

Appendix C. Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

C.1. Summary

The paper examines the quality of endpoint provenance datasets and how these qualities affect the evaluation of provenance-based intrusion detection systems (PIDS). It introduces new metrics and visualization methods to analyze key aspects of datasets, such as the synthetic nature of background activities and the covertness of attacks. The authors validate their approach with experiments on several public datasets and a range of state-of-the-art PIDS frameworks, and they propose a dataset cleanup methodology to mitigate bias in IDS performance evaluations.

C.2. Scientific Contributions

- Creates a New Tool to Enable Future Science
- Addresses a Long-Known Issue
- Provides a Valuable Step Forward in an Established Field

C.3. Reasons for Acceptance

- 1) The work brings new analytical tools to a long-standing problem in provenance-based detection research, which will benefit the community by offering additional perspectives on dataset quality.
- 2) The experimental evaluation, although limited in scope, demonstrates that the proposed metrics correlate with changes in PIDS performance, thereby

providing practical insights into the effects of dataset artifacts.

C.4. Noteworthy Concerns

- 1) The reliance on provenance graphs may not capture all important aspects of dataset quality. For instance, other dimensions such as event type diversity, temporal correlations, or detailed metadata (e.g., syscall arguments) are not fully considered.
- 2) The evaluation uses a relatively narrow selection of datasets, IDS models, and hyperparameter configurations. A broader experimental scope could better illustrate the generality of the findings.
- 3) Measuring syntheticity solely via autocorrelation might miss non-periodic or subtle synthetic patterns; additional measures such as entropy or clustering-based metrics could provide a more complete picture.
- 4) The discussion does not fully address how adversarial tactics in real-world scenarios may interact with the identified dataset artifacts. Expanding on the impact of evolving attacker behaviors would add further depth to the analysis.
- 5) While the visualization approach is innovative, some figures are difficult to interpret and may require additional manual annotation or clearer explanations to enhance their usefulness.
- 6) The paper does not explore the influence of real-world noise or the degradation of dataset quality over time as attacker tactics evolve, which could affect the long-term validity of the findings.
- 7) The paper does not discuss the runtime or scalability implications of the dataset analysis and IDS evaluations, which are critical for real-world deployment.