

Process-Correlated Host Telemetry for Hybrid APT Detection and MITRE ATT&CK Mapping

Aditya Dubey*, Ayan Mondal*, Roshan Rishabh†, Gaurav Sharma†

*Indian Institute of Technology Indore

Email: mt2602193007@iiti.ac.in, ayanm@iiti.ac.in

†Military College of Telecommunication Engineering, India

Email: rrishabh.949m@gov.in, gauravkh07@gmail.com

Abstract—Advanced Persistent Threats (APTs) evade signature-based defences by changing indicators while preserving multi-stage host behaviours. Existing host detectors often require heavyweight provenance capture or clean benign corpora and seldom provide actionable MITRE ATT&CK mappings. This paper presents a reproducible hybrid framework anchored on Sysmon process-creation events, correlating network, DNS, file, registry and image-load activity by `ProcessGuid` into 26 interpretable features over 1,467 OTRF samples spanning eight tactics and 50 techniques; a rule gate and machine-learning classifiers assign tactic and technique hypotheses and attach ATT&CK mitigations. On a fixed 75/25 stratified split the Random Forest achieves 0.621 tactic-level accuracy and 0.444 macro- F_1 against 0.431 and 0.075 for the majority-class baseline, correlated behaviour contributing +0.037 macro- F_1 and repeated cross-validation giving 0.448 ± 0.038 . Applied without retraining to 783,367 MITRE APT29 emulation events it raises 112 alerts across all six mid-chain tactics, a coverage and volume result rather than a precision measurement, at 26,000–33,000 events per second within 361MB of memory. ATT&CK scope analysis finds 21 of 66 APT41 and 1 of 12 APT36 base techniques observable from host telemetry, exposing the modality’s limits for delivery-heavy threats.

Index Terms—Advanced Persistent Threat, MITRE ATT&CK, machine learning, behavioural analysis, Sysmon, intrusion detection, threat classification, adversary emulation.

I. INTRODUCTION

Advanced Persistent Threats (APTs) are prolonged, objective-driven intrusions that change binaries and Indicators of Compromise (IOCs) while retaining the behaviours needed for execution, persistence, lateral movement and exfiltration. Signature-based controls are therefore brittle and provide limited attack-stage context for defence networks and critical infrastructure [1]. Behaviour-based detection aligned with MITRE ATT&CK offers a more durable basis, but ATT&CK is a taxonomy rather than a detector [2].

Existing approaches leave complementary gaps: provenance systems depend on heavyweight whole-system auditing, learning-based log analysers often require clean benign corpora and yield only anomaly verdicts, and network classifiers omit host process and command-line semantics. The unresolved problem is to convert accessible host telemetry into interpretable, ATT&CK-labelled, mitigation-bearing detections that remain reproducible on commodity hardware.

This paper presents the hybrid host-behaviour framework shown in Fig. 1. Each Sysmon process-creation event is

fused, through `ProcessGuid` correlation, with its network, DNS, file, registry and image-load activity to form 26 interpretable features. A rule-based gate limits the events passed to machine-learning classifiers, which infer tactic and technique hypotheses and attach ATT&CK mitigations. The technical distinction is the linear-time fusion of an anchor’s command and process properties with its wider behavioural footprint. Evaluation uses open ATT&CK-labelled telemetry, baselines and feature-group ablation, an unseen APT29 emulation and scope analysis of APT36 and APT41.

Novelty statement. The contribution is not a new learning algorithm or provenance model: Sysmon telemetry, weighted rules, tree ensembles and the ATT&CK knowledge base are each established. It is their integration around a *single-hop*, linear-time process-level correlation that replaces whole-system provenance graphs, requires no benign training corpus and turns every process anchor into an ATT&CK tactic, a ranked technique shortlist and a mitigation set, with ablation evidence that this correlation adds tactic information beyond command-line and process attributes (Section V).

Contributions. This work provides: (i) an ATT&CK-labelled behavioural feature set derived from open atomic telemetry by process-identifier correlation; (ii) a hybrid rule-plus-ML engine mapping tactic and technique hypotheses to mitigations; (iii) tactic- and technique-level evaluation against baselines, with controlled ablation, an explicit class-imbalance analysis and cross-validated variance; (iv) a cross-dataset assessment of alert volume and attack-stage coverage on the unseen MITRE APT29 emulation, which carries no per-event ground truth and is therefore not a precision claim; (v) quantified ATT&CK scope analysis for APT36 and APT41; and (vi) a complete configuration and measured resource profile for reproduction.

II. RELATED WORK

APT detection has progressed from direct signature and IOC matching towards methods that model relationships across events. Signature-driven controls remain efficient for known artefacts but are poorly suited to novel or mutated APT behaviour [1], [3], motivating approaches that preserve temporal, causal or contextual evidence across an intrusion.

Provenance and causal correlation. SLEUTH [4] represents COTS audit records as a dependency graph with tag propagation, HOLMES [5] correlates suspicious information flows

into campaign-level attack graphs and POIROT [6] aligns CTI query graphs with kernel-audit provenance by inexact graph matching, while UNICORN [7] and KAIROS [23] model whole-system provenance for anomaly detection, the latter with graph neural networks trained on benign activity, and ATLAS [8] adds sequence learning over audit-log causality. These systems recover multi-hop causal context far more effectively than isolated-event matching, but require comprehensive audit capture and provenance-graph construction, and generally do not emit native ATT&CK technique-to-mitigation output.

Learned log context and network behaviour. DeepLog [9] learns normal log-event sequences with an LSTM, Log2vec [10] embeds heterogeneous relationships among enterprise log entries and DEEPCASE [11] uses semi-supervised contextual analysis to reduce the events requiring analyst inspection. MLAPT [3] correlates alerts to detect and predict APT progression, while Kitsune [12] provides unsupervised online network anomaly detection through an autoencoder ensemble. Such methods improve contextual or unknown-threat detection, yet may depend on clean normal data or external alert streams and commonly stop at an anomaly or triage verdict. Similarly, classifiers built on CIC-IDS2017 [13] and UNSW-NB15 [14] operate on labelled network traffic but omit the host process and command-line semantics required for host-level TTP interpretation.

ATT&CK-oriented detection and open telemetry. ATT&CK supplies a common vocabulary for tactics, techniques and mitigations [2]. RapSheet [24] builds tactical provenance graphs over ATT&CK-tagged EDR alerts to prioritise triage, technique co-occurrence has been learned from threat reports [25], and supervised learning on Sysmon logs has been applied to single tactics such as lateral movement [26]. OTRF Security-Datasets [15] and Atomic Red Team [16] make labelled host telemetry and repeatable technique emulation accessible, but do not themselves provide a detector that fuses related host events and emits mitigation-bearing reports.

Positioning of this work. Relative to provenance systems, the proposed correlation is a single-hop join of a process-creation anchor with the events sharing its `ProcessGuid`: it avoids graph construction and runs in $O(N)$, but deliberately forgoes multi-hop causal reconstruction. Relative to behavioural log learners, it is supervised from open ATT&CK-labelled telemetry and needs no benign corpus. Relative to ATT&CK-oriented analytics such as RapSheet [24], which scores alerts already raised by existing technique rules, it learns tactic and technique hypotheses from correlated features and attaches mitigations, so the output is a mitigation-bearing report rather than an anomaly score [9], [12] or a triage ranking [11]. Table I summarises these differences. No state-of-the-art accuracy is claimed against systems evaluated on different corpora under different supervision, and the network and novelty stages of Fig. 1 remain specified but unevaluated (Section VI).

III. PROBLEM FORMULATION AND THREAT MODEL

Threat model. Windows Sysmon telemetry is collected on monitored endpoints and forwarded to an analytics server, with

a Suricata network sensor available in the operational deployment. The adversary executes process-level TTPs command execution, persistence, credential access, lateral movement that necessarily leave host artefacts. Delivery actions leaving no host process artefact fall outside the classifier’s scope and motivate the complementary network sensor; Section V quantifies this boundary for a delivery-heavy actor.

Formulation. Let $L = \{e_1, \dots, e_N\}$ be a time-ordered host event log, where each event e carries an event type $\text{id}(e)$ and a process identifier $\pi(e)$ (`ProcessGuid`). Let $A = \{e \in L : \text{id}(e) = 1\}$ be the set of process-creation anchors and let

$$C(a) = \{e \in L : \pi(e) = \pi(a), \text{id}(e) \in \mathcal{E}_c\}, \quad a \in A, \quad (1)$$

be its behavioural context, with $\mathcal{E}_c = \{3, 22, 11, 12, 13, 14, 7\}$ denoting network, DNS, file, registry and image-load events. The feature map $\phi : (a, C(a)) \mapsto x \in \mathbb{R}^d, d = 26$, produces the fused vector, so that the detection task is to learn

$$f : \mathbb{R}^d \rightarrow \mathcal{G} \times \mathcal{T}, \quad f(x) = (\hat{g}, \hat{t}), \quad (2)$$

where $\mathcal{G}$ is the set of $K = 8$ ATT&CK tactics (attack stages) and $\mathcal{T}$ the set of techniques, followed by a deterministic mitigation lookup $\Theta : \mathcal{T} \rightarrow \mathcal{M}$ over the $|\mathcal{M}| = 44$ ATT&CK mitigations. The central hypothesis is that the correlated context $C(a)$ carries information not present in the anchor alone, i.e. $I(\phi(a, C(a)); g) > I(\phi(a, \emptyset); g)$; the ablation of Section V tests this hypothesis empirically.

IV. PROPOSED FRAMEWORK AND METHODOLOGY

The framework (Fig. 1) is a multi-sensor pipeline. A Windows endpoint runs Sysmon and a sensor host runs Suricata; logs are aggregated and transformed into behavioural feature vectors. The detection engine applies a rule-based suspicion gate followed by an ML classifier, maps the result to ATT&CK and emits a report with mitigations.

Evaluated versus deployment components. The experiments of Section V replay recorded OTRF and APT29 Sysmon logs and evaluate only Sysmon ingestion, `ProcessGuid` correlation, the 26-feature extractor, the rule gate, the two classifiers and the ATT&CK/mitigation mapping. The attacker and victim virtual machines, Suricata, Wazuh, the dashboard and the Isolation-Forest novelty stage of Section IV-B belong to the proposed operational deployment; they are not evaluated and no reported result depends on them.

Knowledge base. The ATT&CK v16.1 STIX bundle is parsed into 656 techniques, 160 groups and 44 mitigations, together with the TTP profiles of the studied groups.

Data and labelling. ATT&CK-labelled host telemetry is drawn from the OTRF Security-Datasets atomic Windows corpus [15]. Each constituent dataset exercises exactly one technique, which yields clean labels at both technique and tactic level without manual annotation.

Feature engineering. Every process-creation event is featured and enriched by evaluating (1) through a single-pass hash index on `ProcessGuid`, giving the 26 features of Table II. The resulting corpus holds 1,467 samples over 8 tactics and 50 techniques, of which 848 (57.8%) carry non-empty correlated context. Indexing and lookup are both $O(N)$

TABLE I: Capability Comparison with Representative Prior Work

Study	Input / correlation	Behav.	ML	ATT&CK	Unknown	Report	Gap addressed in this work
SLEUTH [4], HOLMES [5], UNICORN [7], KAIROS [23], ATLAS [8] RapSheet [24]	Kernel audit; provenance / causal graph	Yes	Partial	Partial	Yes	Partial	Whole-system capture and graph construction; no technique-to-mitigation output
DeepLog [9], Log2vec [10], DEEPCASE [11]	EDR alerts; tactical graph	Yes	No	Yes	Partial	Yes	Needs pre-existing rule alerts; no learned hypotheses
MLAPT [3]	Log-event sequences	Yes	Yes	No	Partial	Limited	Need clean benign corpora; stop at anomaly or triage verdict
Kitsune [12], CIC-IDS / UNSW [13], [14]	Network IDS alerts Packets / network flows	Yes Partial	Yes Yes	Limited No	Limited Limited	Limited Limited	Depends on external alert feeds Generic labels; no host semantics
Proposed	Sysmon; single-hop join	Yes	Yes	Yes	Partial [†]	Yes	Integrated; ATT&CK-mapped; commodity

[†] Assessed only as transfer to one unseen campaign (APT29, no ground truth); the Isolation-Forest novelty stage is specified but not evaluated (Section VI).

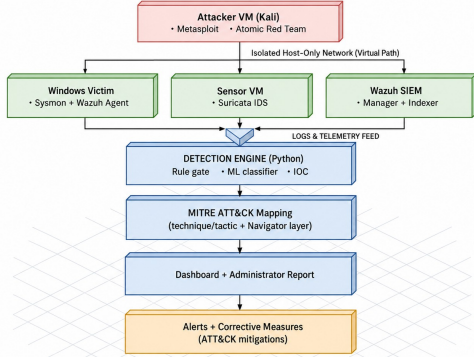


Fig. 1: Multi-sensor detection architecture. Only the Sysmon, rule-gate, ML and ATT&CK-mapping path is evaluated here; the other blocks are deployment-only (Section V).

TABLE II: Behavioural Feature Set (26 Features)

Group	Features
Command (11)	length, token count, Shannon entropy, digit/upper/non-alphanumeric ratios, encoded-command flag, URL and IP flags, suspicious-keyword count, parent-command length
Process (7)	image, parent image, LOLBin flags, parent-is-shell, integrity level, temp/appdata path indicators
Behaviour (8)	network connections, unique destination ports and IPs, public-IP flag, DNS queries, file writes, registry modifications, image loads

in the log length, so enrichment adds no asymptotic cost over a single linear scan the property that keeps the pipeline viable on commodity hardware.

Models. A Random Forest over the structured features provides an interpretable classifier, and a logistic regression over TF-IDF of the command line concatenated with those features a lexical contrast; both use the protocol of Table III.

A. Detection Algorithm

Algorithm 1 operationalises (1) by indexing Sysmon events by `ProcessGuid` and fusing command, process and context features for each process-creation anchor. Algorithm 2 then applies the hybrid detector and ATT&CK mitigation mapping.

Algorithm 1: Behavioural feature extraction

```

Input: host log L   Output: X, y
1: for e in L with EventID in {3,22,11,12,13,14,7}:
2:   index[ProcessGuid(e)].append(e)
3: for a in L with EventID = 1:
4:   x <- cmd_feats(a) + proc_feats(a) +
   ctx_feats(index[pi(a)])
5:   append x to X; technique(a) to y
6: return X, y

```

Algorithm 2: Hybrid rule + ML detection

```

Input: L, models M_tac, M_tech, map Theta, threshold tau
1: for (a,x) in enrich(L):
2:   S <- sum_i w_i r_i(a)
3:   if S >= tau:
4:     t^* <- argmax P(.|x; M_tech)
5:     g^* <- argmax P(.|x; M_tac)
6:     alert(a, t^*, g^*, Theta(t^*))
7: return report(alerts)

```

B. Mathematical Formulation

The command-line entropy feature, which separates obfuscated and encoded invocations from ordinary ones, is the Shannon entropy $H(s) = -\sum_i p_i \log_2 p_i$ of the symbol distribution $\{p_i\}$ of a command s [17], and lexical evidence is the TF-IDF weight $w_{t,d} = \text{tf}(t,d) \log(|D|/(1 + |\{d \in D : t \in d\}|))$ of term t in command d over corpus D [18]. Each tree of the Random Forest [19] splits so as to reduce the Gini impurity $G = 1 - \sum_k p_k^2$, and the ensemble of B trees votes $\hat{y} = \arg \max_k \sum_{b=1}^B \mathbb{1}[h_b(x) = k]$. The lexical model uses the softmax posterior $P(y = k | x) = e^{w_k^T x} / \sum_j e^{w_j^T x}$, trained by cross-entropy weighted by $\beta_k = N/(KN_k)$ so that the 632-sample Persistence class does not dominate the 12-sample Collection class. Behaviour unseen in training is scored by an Isolation Forest [20], $s(x,n) = 2^{-\mathbb{E}[h(x)]/c(n)}$; this novelty stage is specified for the operational deployment and is not exercised by any experiment in Section V. The rule gate accumulates weighted indicator rules $r_i \in \{0,1\}$ and alerts when

$$S(a) = \sum_i w_i r_i(a) \geq \tau. \quad (3)$$

The threshold τ trades recall against analyst workload: with $|A|$ anchors the expected number of alerts is $|A| \Pr[S \geq \tau]$, so τ is the single knob that fixes the alert budget independently of the classifier. With precision P , recall R and $F_1 = 2PR/(P +$

R) defined per class in the usual way, performance is reported as

$$\text{Macro-}F_1 = \frac{1}{K} \sum_{k=1}^K F_{1,k}, \quad (4)$$

macro-averaging being the informative criterion here because accuracy alone is inflated by the dominant Persistence class, as the baselines of Table IV confirm.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Protocol and metrics. The protocol is that of Table III. Tactic classification over eight classes is the primary task because attack stage drives the defender’s response. Accuracy measures the overall decision rate, whereas macro- F_1 in (4) weights each tactic equally and is therefore more informative under the observed class imbalance. Technique performance is reported as top-1 and top-3 accuracy for the 19 techniques having at least 10 samples. Because the APT29 corpus lacks per-event ground truth, its evaluation is restricted to alert volume and stage coverage.

Baseline comparison and ablation. Table IV shows that the majority-class baseline obtains the highest trivial-baseline accuracy (0.431) but only 0.075 macro- F_1 , exposing the effect of class priors. The proposed Random Forest reaches 0.621 accuracy and 0.444 macro- F_1 : a 0.190 accuracy increase and a $5.9\times$ macro- F_1 gain over that baseline. Logistic regression with TF-IDF reaches 0.599 and 0.434, respectively, so the structured Random Forest retains advantages of 0.022 accuracy and 0.010 macro- F_1 without relying on additional command vocabulary. The ablation provides direct evidence for the feature design. Command-only performance is 0.550/0.366; adding process and path context raises it to 0.583/0.407 and adding the eight correlated-behaviour features yields 0.621/0.444. Thus, every feature group contributes, while correlated behaviour alone adds +0.037 macro- F_1 , validating the feature-fusion hypothesis of Section III.

Class-wise analysis and the effect of imbalance. Resolved by tactic, per-class F_1 tracks training support closely: Persistence (0.80), Credential Access (0.68) and Defense Evasion (0.65) are highest, followed by Privilege Escalation (0.46), Execution (0.38) and Lateral Movement (0.37), while the sparsely represented Discovery (0.12) and Collection (0.00) classes remain weak. Support ranges from 632 Persistence anchors (43.1%, exactly the majority-baseline accuracy) to 12 Collection anchors (0.8%), a 53:1 ratio, and this affects the results in three distinct ways. First, accuracy is prior-dominated, so every comparison in Table IV is made on macro- F_1 . Second, the 25% hold-out leaves only 5 Discovery and 3 Collection test anchors, so those two per-class scores are high-variance estimates that a single prediction can move substantially, yet each carries one-eighth of the macro- F_1 ; they should be read as evidence of absent training support, not as capability estimates. Third, balanced weighting scales the Collection loss by $632/12 \approx 53$ relative to Persistence, which raises minority recall pressure but cannot supply the behavioural diversity missing from nine training anchors.

Balancing and evaluation strategies considered. Table IV compares the alternatives on the same split. Removing class weighting yields 0.654/0.471 and uniform balanced weighting 0.594/0.430, against 0.621/0.444 for the reported `balanced_subsample`; the 0.041 macro- F_1 spread is of the same order as the ± 0.038 fold-to-fold deviation of Table IV, so the three are not distinguishable at this corpus size and the a-priori choice to weight rare classes was retained rather than re-selected on the hold-out. Random undersampling was rejected on evidence: Collection contributes nine training records, so it reduces training to 72 samples and collapses performance to 0.371/0.309. Synthetic oversampling [27] was not applied because most of the 26 features are binary flags, small counts and one-hot process identities, for which an interpolated point corresponds to no realisable command line, and because interpolating between a handful of anchors that inherit a dataset-level label would replicate that label noise; for shallow models on Sysmon data, resampling has anyway been reported to give only marginal F_1 gains [26]. Finally, as a single 75/25 split is a fragile basis for an imbalanced eight-class problem, the protocol was repeated as stratified 5-fold cross-validation over five repetitions (Table IV): macro- F_1 is 0.448 ± 0.038 and accuracy 0.609 ± 0.026 over 25 folds, placing the single-split values of Table IV within one standard deviation of both means, so the reported result is not an artefact of one favourable partition. The residual deficit on rare tactics is a data-collection problem: no reweighting recovers a class of twelve samples.

Interpreting technique versus tactic output. The model attains 0.362 top-1 and 0.680 top-3 accuracy over the 19 techniques with at least ten samples, and the gap is diagnostic: of the 213 top-1 errors, 91 (42.7%) assign the record to a different technique *within the correct tactic*, so the tactic implied by the top-1 technique is right for 0.635 of test records against 0.362 at technique level. Confusion is concentrated inside tactics, between techniques sharing a command and process surface such as the persistence mechanisms driven through scheduled tasks or registry writes, and largely dissolves at tactic granularity. This fixes the operational reading: the tactic is the routing decision selecting the response playbook, supported by both the 0.621 direct and 0.635 implied figures, while the technique is a ranked hypothesis to be surfaced as the top-3 shortlist (0.680) that an analyst confirms against the raw command line carried in the alert, never as a single automated verdict.

Cross-dataset alert volume and stage coverage. The trained engine was applied without retraining to the unseen MITRE APT29 adversary-emulation corpus [22]. This corpus provides no per-event ground truth, so the experiment is reported strictly as alert volume and attack-stage coverage; it is not a measurement of precision, recall or detection rate, and no such quantity is derived from it anywhere in this paper. Table Va reports 46 alerts from 196,081 Day-1 events and 66 from 587,286 Day-2 events. In total, 112 alerts are raised from 783,367 events, corresponding to an alert rate of 1.4×10^{-4} , or approximately one alert per 7,000 events; this is an analyst-workload figure,

TABLE III: Reproducibility Configuration. Every Reported Number Uses These Settings

Item	Setting
<i>Data</i>	
Training corpus	OTRF atomic Windows [15], 89 of 100 host datasets; 1,467 EventID-1 anchors, 8 tactics, 50 techniques; 848 (57.8%) with non-empty context
Labels	ATT&CK v16.1 [2]; one technique per source dataset
Held-out corpus	APT29 emulation [22] Day 1 and Day 2, 783,367 events, used for inference only
<i>Protocol</i>	
Split	stratified 75/25, test_size=0.25, random_state=42
Seed	42 for the split, the forest and all dummy baselines
Technique subset	the 19 techniques with ≥ 10 samples
Cross-validation	repeated stratified 5-fold, 5 repeats, seed 42 (Table IV)
<i>Rule gate (3)</i>	
Threshold	$\tau = 2$
Weights w_i	encoded/base64 command 2; ≥ 2 suspicious keywords 2 (exactly one, 1); public-IP connection 2; LOLBin image 1; LOLBin spawned by shell 1; execution from temp/appdata 1; URL or IP in command 1; registry modification by LOLBin 1; high integrity from shell 1

TABLE IV: Tactic-Level Results: Baselines, Proposed Models, Feature-Group Ablation, Cross-Validated Variance and Class-Balancing Alternatives

Method / feature set	Accuracy	Macro- F_1
<i>Trivial baselines</i>		
Dummy (most frequent)	0.431	0.075
Dummy (stratified)	0.232	0.099
Dummy (uniform)	0.128	0.090
<i>Learned models (full feature set)</i>		
Logistic regression + TF-IDF	0.599	0.434
Random Forest (proposed)	0.621	0.444
<i>Ablation (Random Forest)</i>		
Command only (11 features)	0.550	0.366
+ Process / context (18 features)	0.583	0.407
+ Correlated behaviour (26 features)	0.621	0.444
<i>Repeated stratified 5-fold cross-validation, 5 repeats</i>		
Mean over 25 folds	0.609	0.448
Standard deviation	0.026	0.038
<i>Class-balancing strategy (identical split and seed)</i>		
No class weighting	0.654	0.471
class_weight=balanced	0.594	0.430
Random undersampling (9/class, $n = 72$)	0.371	0.309

not an accuracy figure. Alerts occur in all six mid-chain tactics on both days, and Fig. 2a shows that no single stage dominates the total. The technique hypotheses attached to those alerts are consistent with APT29's documented tradecraft LSASS credential dumping (T1003.001), WMI/DCOM lateral movement (T1047, T1021.003), scheduled-task and WMI-subscription persistence (T1053.005, T1546.003), UAC bypass (T1548.002) and event-log tampering (T1562.002) and none of this data was used for retraining. Because the labels are unverified at event level, these technique identifiers are model hypotheses on gated events rather than confirmed detections, and the defensible reading of the experiment is that a host-only engine trained on atomic single-technique telemetry continues to raise a small, stage-diverse alert volume on an unseen multi-stage campaign. A low alert rate alone, it should be stressed, does not imply a low false-positive rate [28].

Item	Setting
<i>Models</i>	
Random Forest	400 trees, Gini, max_features= $\sqrt{d}$, unlimited depth, min_samples_split=2, min_samples_leaf=1, class_weight=balanced, subsample
Categorical enc.	one-hot on image, parent_image; min_frequency=5, max_categories=40, unknown ignored
Numeric features	passthrough, no scaling
Logistic reg.	$C = 3.0$, L2, lbfgs, max_iter=4000, class_weight=balanced
TF-IDF (command)	char_wb, n-grams (2, 4), min_df=2
TF-IDF (image)	word tokens [a-z0-9._\-\+], min_df=1
<i>Environment</i>	
Software	Python 3.14.7, scikit-learn 1.9.0 [21], NumPy 2.5.2, pandas 3.0.5
Hardware	Intel Core Ultra 5 226V, 8 cores / 8 threads, 2.1 GHz base, 16 GB RAM, Windows 11; no GPU

TABLE V: Cross-Dataset Alert Volume and Stage Coverage on the Unseen APT29 Emulation (no per-event ground truth; not a precision or recall measurement) and ATT&CK Scope for the Target Groups

<i>(a) APT29 emulation [22], engine applied without retraining</i>				
	Events	Alerts	Stages	Alert rate
Day 1	196,081	46	6	2.3×10^{-4}
Day 2	587,286	66	6	1.1×10^{-4}
Total	783,367	112	6	1.4×10^{-4}
<i>(b) ATT&CK profile of target groups (indicative scope)</i>				
Group	Base tech.	Tactics	In scope	Coverage
APT41 (G0096)	66	14	21	32%
APT36 (G0134)	12	5	1	8%

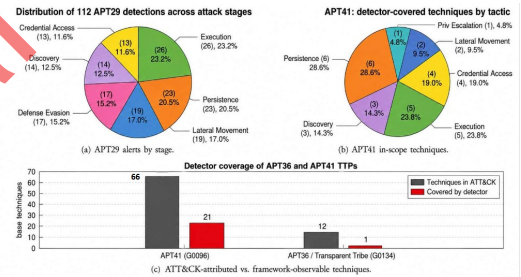


Fig. 2: Cross-dataset and target-actor analysis: (a) the 112 APT29 alerts by attack stage (coverage, not verified detections); (b) the 21 in-scope APT41 techniques by tactic; (c) attributed versus observable technique counts.

Actor scope and implications. Table Vb and Fig. 2b,c quantify the threat-model boundary. Host behaviour covers 21 of APT41's 66 attributed base techniques (32%) across seven tactics; coverage for APT36 is 1 of 12 (8%), consistent with its delivery-heavy public profile. Process-correlated host telemetry is thus applicable to execution-resident activity, whereas delivery-oriented behaviour requires the complementary network and e-mail sensing of Fig. 1.

VI. LIMITATIONS AND REPRODUCIBILITY

Scope and validity. The evaluated classifier is Windows- and Sysmon-specific and anchors analysis on process-creation events; delivery activity leaving no host process artefact is

outside its scope, as are the deployment-only components of Fig. 1. The atomic corpus introduces label noise because every anchor inherits its dataset's technique, including benign setup activity, so technique labels are ranked hypotheses while tactic is the reliable output. The APT29 experiment carries no ground truth and transfer is tested on a single campaign; the APT36/APT41 figures measure ATT&CK scope, not operational detection performance.

Reproducibility. Table III states the complete configuration behind every number in this paper, and all inputs are public [2], [15], [22]. One detail materially affects exact replication: a fixed-seed Random Forest draws its per-node feature subsets in column order, so the order in which the 26 features are presented is itself part of the configuration, and permuting it leaves results equal only to within the fold-to-fold variation of Table IV (0.621 against 0.616 accuracy for two orderings of the identical feature set). The released scripts fix the ordering per model and regenerate every table and figure; they are available from the corresponding author on request.

Computational cost. With N events, $|A|$ anchors, R gate rules and a forest of B trees of depth D , Algorithm 1 needs one pass to build the `ProcessGuid` index and one to featurise anchors, i.e. $O(N)$ time and at most $O(N)$ index memory; Algorithm 2 adds $O(R)$ per anchor for the gate and $O(BD)$ per gated anchor for the forest, giving $O(N + |A|R + |A_\tau|BD)$ overall, where A_τ is the gated subset. As commodity reproducibility is a stated motivation, the constants were also measured on the hardware of Table III. Training the reported Random Forest on 1,467 samples takes 5.6 s and inference over the 367-record hold-out 183 ms. End-to-end detection over the APT29 corpus, including decompression, JSON parsing, `ProcessGuid` indexing, featurisation, the rule gate and classification of gated records, takes 7.4 s for the 196,081 Day-1 events and 17.8 s for the 587,286 Day-2 events (mean of three runs, spread below 0.4 s): a throughput of 26,000–33,000 events per second on eight CPU cores with no GPU. Peak working set is 361 MB and is effectively unchanged across the threefold difference in event count, because the loader streams the log and keeps only process-creation anchors and per-`ProcessGuid` counters, so memory scales with the number of live processes rather than with log length. That bounded memory profile, more than the throughput, is what makes Fig. 1 deployable on an ordinary endpoint.

VII. CONCLUSION

This work integrated single-hop `ProcessGuid` correlation of Sysmon process anchors and related host activity into an ATT&CK-aligned hybrid detector that assigns tactic and technique hypotheses and attaches mitigations. On the stratified hold-out the Random Forest reached 0.621 accuracy and 0.444 macro- F_1 against 0.431 and 0.075 for the majority-class baseline, correlated behaviour contributing +0.037 macro- F_1 , and repeated cross-validation placed macro- F_1 at 0.448 ± 0.038 . Tactic should drive the operational decision, while technique output is best used as a top-3 shortlist (0.680). Without retraining, the engine raised 112 alerts across six mid-chain tactics

in 783,367 unlabelled APT29 events, measuring alert volume and stage coverage rather than precision, at 26,000–33,000 events per second within 361 MB. Scope analysis covered 21 of 66 APT41 and 1 of 12 APT36 base techniques. Label noise, class imbalance, Windows-only evaluation and the unevaluated network and novelty stages bound these conclusions; future work will integrate Suricata and Sigma rules and evaluate novelty detection and operational precision on benign and lab-generated telemetry.

REFERENCES

- [1] A. Alshamrani, S. Myneni, A. Chowdhary and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges and research opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1851–1877, 2019.
- [2] B. E. Strom *et al.*, "MITRE ATT&CK: Design and philosophy," MITRE, Tech. Rep., Mar. 2020.
- [3] I. Ghafir *et al.*, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Gener. Comput. Syst.*, vol. 89, pp. 349–359, Dec. 2018.
- [4] M. N. Hossain *et al.*, "SLEUTH: Real-time attack scenario reconstruction from COTS audit data," in *Proc. 26th USENIX Security Symp.*, 2017, pp. 487–504.
- [5] S. M. Milajerdi *et al.*, "HOLMES: Real-time APT detection through correlation of suspicious information flows," in *Proc. IEEE Symp. Security Privacy*, 2019, pp. 1137–1152.
- [6] S. M. Milajerdi *et al.*, "POIROT: Aligning attack behavior with kernel audit records for cyber threat hunting," in *Proc. ACM CCS*, 2019, pp. 1795–1812.
- [7] X. Han *et al.*, "UNICORN: Runtime provenance-based detector for advanced persistent threats," in *Proc. NDSS*, 2020.
- [8] A. Alsaheel *et al.*, "ATLAS: A sequence-based learning approach for attack investigation," in *Proc. 30th USENIX Security Symp.*, 2021, pp. 3005–3022.
- [9] M. Du, F. Li, G. Zheng and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM CCS*, 2017, pp. 1285–1298.
- [10] F. Liu *et al.*, "Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise," in *Proc. ACM CCS*, 2019, pp. 1777–1794.
- [11] T. van Ede *et al.*, "DEEPCASE: Semi-supervised contextual analysis of security events," in *Proc. IEEE Symp. Security Privacy*, 2022, pp. 522–539.
- [12] Y. Mirsky, T. Doitshman, Y. Elovici and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. NDSS*, 2018.
- [13] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy*, 2018, pp. 108–116.
- [14] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.
- [15] R. Rodriguez and J. L. Rodriguez, "Security Datasets," OTRF, GitHub. [Online]. Available: <https://github.com/OTRF/Security-Datasets>
- [16] Red Canary, "Atomic Red Team," GitHub. [Online]. Available: <https://github.com/redcanaryco/atomic-red-team>
- [17] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
- [18] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
- [19] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [20] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 413–422.
- [21] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [22] OTRF, "APT29 evals open datasets," 2020, in [15], datasets/compound/apt29.
- [23] Z. Cheng *et al.*, "Kairos: Practical intrusion detection and investigation using whole-system provenance," in *Proc. IEEE Symp. Security Privacy*, 2024, pp. 3533–3551.
- [24] W. U. Hassan, A. Bates and D. Marino, "Tactical provenance analysis for endpoint detection and response systems," in *Proc. IEEE Symp. Security Privacy*, 2020, pp. 1172–1189.
- [25] R. Al-Shaer, J. M. Spring and E. Christou, "Learning the associations of MITRE ATT&CK adversarial techniques," in *Proc. IEEE Conf. Commun. Netw. Security*, 2020, pp. 1–9.
- [26] C. Smiliotopoulos and G. Kambourakis, "Machine learning for lateral movement detection using Sysmon logs: An empirical comparison of imbalanced and resampled data," *Int. J. Inf. Security*, vol. 25, no. 1, Art. no. 38, 2026.
- [27] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [28] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security Privacy*, 2010, pp. 305–316.