

SENTINEL: A Multi-Pathway Architecture for Detecting Living-Off-the-Land APT Attacks on Windows Command Lines

Ahad Bin Islam Shoeb^{*}, Kamrul Hasan[†], Jamal Uddin Tanvin[‡], Liang Hong[†], Imtiaz Ahmed[¶],
Md Arif Billah^{*}, and Al Amin[§]

^{*}University of Dhaka, Dhaka, Bangladesh, [†]Tennessee State University, Nashville, TN, USA

[‡]Military Institute of Science and Technology, Dhaka, Bangladesh, [§]Huston–Tillotson University, Austin, TX, USA

[¶]Howard University, Washington, DC, USA

E-mail: ahadbinislam-2019117810@cs.du.ac.bd, mhasan1@tnstate.edu, jamaluddintanvin@outlook.com, lhong@tnstate.edu, imtiaz.ahmed@howard.edu, mdarif-2019617815@cs.du.ac.bd, and aamin@htu.edu

Abstract—Living-Off-the-Land (LOTL) is the dominant evasion technique of Advanced Persistent Threat (APT) actors, exploiting legitimate Windows utilities to conduct malicious operations without deploying custom malware and enabling state-sponsored campaigns to maintain persistent access within military and critical defense infrastructure for extended periods. Existing detection methods fail against obfuscated commands and multi-stage attack sequences, as demonstrated by the Volt Typhoon APT campaign, which maintained undetected access to U.S. critical infrastructure for over 18 months using exclusively signed Windows utilities. We present SENTINEL, a multi-pathway architecture integrating BERT-based semantic encoding, character-level CNN for obfuscation invariance, inter-command attention for multi-stage pattern recognition, and autoencoder-based anomaly scoring. Evaluated on a balanced Volt Typhoon benchmark derived from Microsoft and CISA threat intelligence advisories, SENTINEL achieves 92.0% accuracy on documented state-sponsored attack commands and 91.2% on obfuscated variants, compared to 74.0% and 72.0% for standalone BERT. Per-class analysis reveals that models achieving over 98% overall validation accuracy on imbalanced data exhibit only 44–58% malicious recall on balanced adversarial sets. Character-level processing contributes 5.6 percentage points of obfuscation invariance, and the 8.0 percentage point gap over augmentation-only baselines confirms structural architectural value beyond data-driven robustness alone.

Index Terms—living-off-the-land attacks, LOTL detection, Windows command-line security, obfuscation robustness, Volt Typhoon, BERT, character-level CNN, anomaly detection, intrusion detection

I. INTRODUCTION

Living-Off-the-Land (LOTL) has emerged as the primary evasion technique of Advanced Persistent Threat (APT) actors. Rather than deploying custom malware, APT operators

exploit trusted Windows utilities, including PowerShell, WMI, certutil, and schtasks to conduct malicious operations while blending with routine administrative activity [1], [2]. Because these binaries are whitelisted by enterprise security controls and used identically by system administrators and adversaries, LOTL-based APT attacks are structurally resistant to signature-based detection, binary inspection, and sandbox analysis. The attack surface is not a novel executable but the administrative tooling already present in the target environment.

The strategic severity of this threat for military and defense networks was established in May 2023, when Microsoft Threat Intelligence documented the Volt Typhoon campaign: a PRC state-sponsored operation that compromised U.S. critical infrastructure across the communications, energy, and water sectors while evading commercial endpoint detection and response (EDR) products for over 18 months [3]. Palo Alto Networks Unit 42 corroborated the campaign scope and tactical progression [20], while joint CISA/NSA/FBI advisories identified systematic obfuscation and multi-stage command sequencing as the primary evasion mechanisms [1]. For defense networks where the same Windows administrative utilities underpin both legitimate operations and adversary tradecraft, LOTL represents a paradigm shift requiring detection capabilities well beyond what current solutions provide.

Existing detection approaches fail against sophisticated LOTL attacks for three structural reasons. **First**, token-based methods including TF-IDF and BERT [4] fragment obfuscated commands through base64 encoding, string concatenation, and alias substitution into non-informative subwords, causing accuracy to drop from over 98% on clean validation data to 52–74% on adversarial inputs. **Second**, single-command classifiers treat each command independently and cannot recognize that individually benign commands such as `whoami`, `net user`, `reg query`, and `ntdsutil` constitute an attack progression when executed in a coordinated sequence, mirroring Volt Typhoon’s reconnaissance-to-credential-access workflow [1]. **Third**, supervised classifiers trained on known patterns often

Accepted for publication in the Proceedings of the 2026 IEEE Military Communications Conference (MILCOM 2026). This is the authors’ accepted version; the final published version will appear in IEEE Xplore. © 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

produce highly confident misclassifications for novel tool combinations and encoding strategies absent from the training distribution, thereby providing no effective triage signal for zero-day LOTL techniques. [5].

We present SENTINEL (Semantic ENcoding with characTer-level INvariance and intEr-command modeling for LOTL detection), the first end-to-end architecture to jointly address all three failure modes through four complementary detection pathways: (1) a BERT-based semantic encoder [4] for contextual command intent modeling; (2) a character-level CNN [6] for obfuscation-invariant surface analysis; (3) an inter-command attention module for multi-stage attack correlation; and (4) an autoencoder-based anomaly detector for distributional shift triage.

The contributions of this work are as follows:

- 1) **A unified multi-pathway detector.** We design SENTINEL, to our knowledge the first single end-to-end trainable model that simultaneously confronts obfuscation invariance, semantic intent discrimination, multi-stage temporal correlation, and distributional-shift triage for Windows command-line LOTL detection, rather than addressing these failure modes in isolation.
- 2) **An intelligence-grounded balanced benchmark.** We curate a class-balanced Volt Typhoon evaluation set from government and vendor threat-intelligence advisories and use it to drive a per-class analysis that exposes malicious-recall failures which conventional imbalanced metrics conceal.
- 3) **An evaluation-methodology finding with operational stakes.** We demonstrate that headline validation accuracy exceeding 98% can co-occur with malicious recall as low as 44–58% on balanced adversarial inputs, showing that standard reporting practice systematically overstates readiness for defense-network monitoring.
- 4) **Quantified architectural (not data-driven) gains.** Through leave-one-out ablation, we isolate an 8.0 percentage-point improvement attributable to multi-pathway integration beyond augmentation-only baselines, establishing that the robustness is structural and cannot be recovered by training-data diversity alone.

II. BACKGROUND AND RELATED WORK

A. Traditional Machine Learning Approaches

Early LOTL detection relied on handcrafted features with classical classifiers. Hendler et al. [7] achieve 91–94% F1 on PowerShell detection using TF-IDF with Random Forest; however, any syntactic transformation defeats lexical feature extraction [8]. Bag-of-words and n-gram representations cannot model cross-command dependencies or recognize semantic equivalence between utilities serving identical attack purposes, such as `ntdsutil` and `vssadmin` for credential access. These approaches share a structural limitation: they operate on surface form rather than semantic intent, making them unsuitable for LOTL scenarios where defenders and adversaries share the same vocabulary of legitimate tools [2].

B. Deep Learning and Transformer Methods

LSTM-based models provide temporal sequence modeling but suffer gradient vanishing over long command chains [9], losing early reconnaissance context before credential-theft commands appear. Character-level CNNs [6], [10] handle obfuscation by operating below token boundaries but cannot distinguish malicious from benign commands with identical character distributions (e.g., `ntdsutil snapshot` for disaster recovery versus credential harvesting). BERT [4] advances semantic understanding but was designed for natural language: WordPiece tokenization fragments base64-encoded payloads into meaningless subwords, causing a 25–27 percentage point accuracy drop on adversarial inputs. Domain-adapted variants such as SecurityBERT [11] and LogBERT [12] improve security-event modeling but inherit the same tokenization vulnerability and do not provide the character-level obfuscation invariance that command-line LOTL detection requires.

C. Specialized LOTL Systems

The LOLBAS project [13] catalogs 347+ abusable Windows binaries, but signature matching breaks under any syntactic transformation. LOLWTC [14] reaches 93.83% validation accuracy via handcrafted features yet collapses to 52.0% on Volt Typhoon and 48% on obfuscated variants once encoding destroys its feature basis. PowerPeeler [15] robustly handles PowerShell-specific obfuscation but offers no coverage for WMI, `certutil`, or `ntdsutil` utilities central to Volt Typhoon activity. Ongun et al. [16] reduce annotation cost through active learning, and CmdCaliper [21] shows that embedding quality is critical for downstream detection, but both rely on command-structure features that degrade under obfuscation. Critically, no prior architecture jointly addresses obfuscation invariance, semantic discrimination, temporal sequence modeling, and anomaly detection in a single end-to-end trainable framework evaluated against documented state-sponsored campaigns.

III. SENTINEL ARCHITECTURE

A. Overview

SENTINEL detects LOTL attacks through four parallel pathways integrated via learned fusion. Fig. 1 illustrates the full architecture: input commands flow through dual encoding (WordPiece tokenization and raw character conversion) into four independent pathways whose outputs are concatenated into a unified 1856-dimensional representation before the classification head. The BERT pathway captures semantic intent through contextual embeddings, recognizing credential-access commands such as `reg save HKLM\SAM` despite syntactic variation. The CharCNN pathway provides obfuscation invariance by operating on raw character sequences, detecting base64 padding, operator manipulation, and bracket nesting independent of tokenization. The inter-command attention pathway models multi-stage attack progressions by learning that reconnaissance commands followed by credential access constitute coordinated campaigns. The anomaly detection pathway flags distributional shifts through autoencoder

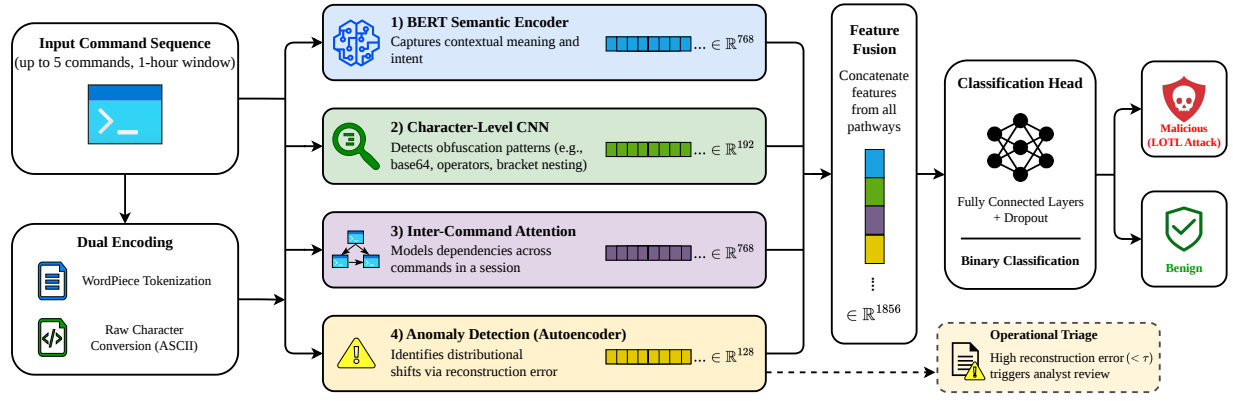


Fig. 1. SENTINEL architecture: SENTINEL encodes inputs using WordPiece and characters, processes them through BERT, sequence attention, CharCNN, and anomaly detection, and then fuses features into a 1856-D vector for malicious/benign classification.

reconstruction error, identifying novel command combinations absent from training data.

B. BERT Semantic Encoder

The BERT encoder produces contextual embeddings capturing command intent. Each token t_i embeds as:

$$\mathbf{e}_i = \mathbf{W}_{\text{token}}[t_i] + \mathbf{W}_{\text{pos}}[i] + \mathbf{W}_{\text{seg}}[s] \quad (1)$$

where $\mathbf{W}_{\text{token}} \in \mathbb{R}^{V \times 768}$ provides token embeddings, $\mathbf{W}_{\text{pos}}$ encodes absolute positions, and $\mathbf{W}_{\text{seg}}$ encodes segment identifiers. Twelve transformer layers [17] apply multi-head self-attention and feedforward transformations, producing output $\mathbf{H}^{(12)} = [\mathbf{h}_1, \dots, \mathbf{h}_n]$ with $\mathbf{h}_i \in \mathbb{R}^{768}$ flowing to three downstream pathways: inter-command attention, anomaly detection, and feature fusion.

C. Character-Level CNN

The CharCNN pathway detects obfuscation patterns invisible to token-based methods. Commands convert to fixed-length ASCII sequences $\mathbf{x}_{\text{char}} \in \mathbb{Z}_{[0,255]}^{512}$. Each character embeds through $\mathbf{W}_{\text{char}} \in \mathbb{R}^{256 \times 32}$. Three parallel convolutions with filter sizes $k \in \{3, 4, 5\}$ extract character n -grams:

$$\mathbf{f}_i^{(k)} = \text{ReLU} \left(\sum_{j=0}^{k-1} \mathbf{W}_j^{(k)} \mathbf{e}_{i+j} + b^{(k)} \right) \quad (2)$$

Global max-pooling identifies the strongest pattern per filter size: $\mathbf{g}^{(k)} = \max_i \mathbf{f}_i^{(k)} \in \mathbb{R}^{64}$. Concatenation yields $\mathbf{h}_{\text{char}} = [\mathbf{g}^{(3)}; \mathbf{g}^{(4)}; \mathbf{g}^{(5)}] \in \mathbb{R}^{192}$, capturing base64 padding (==), PowerShell concatenation operators, and bracket nesting regardless of position within the command string.

D. Inter-Command Attention

This module models dependencies across commands within a session. Each command c_i in sequence $S = [c_1, \dots, c_T]$ with $T \leq 5$, grouped in 1-hour windows, is encoded via BERT and mean-pooled to $\mathbf{h}_i \in \mathbb{R}^{768}$. Multi-head attention

with $H = 4$ heads and $d_k = 192$ computes inter-command dependencies:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \mathbf{V} \quad (3)$$

Training applies dual supervision balancing single-command and sequence-level classification:

$$\mathcal{L}_{\text{seq}} = \alpha \mathcal{L}_{\text{single}} + (1 - \alpha) \mathcal{L}_{\text{sequence}}, \quad \alpha = 0.6 \quad (4)$$

Training sequences ($n = 1,247$) include 892 benign administrative workflows and 355 malicious progressions labeled according to MITRE ATT&CK phases [18], with inter-rater agreement of Cohen's $\kappa = 0.87$.

E. Anomaly Detection

The anomaly branch identifies out-of-distribution commands through autoencoder reconstruction error [5]. The autoencoder is pretrained solely on benign training data and frozen during joint SENTINEL training, preventing data leakage. The encoder compresses $\mathbf{h}_{\text{BERT}} \in \mathbb{R}^{768}$ to a 128-dimensional latent representation $\mathbf{z}_{\text{anomaly}}$ through two linear layers with ReLU activations. A symmetric decoder reconstructs the BERT embedding $\hat{\mathbf{h}}_{\text{BERT}}$. Reconstruction error $\mathcal{L}_{\text{recon}} = \|\mathbf{h}_{\text{BERT}} - \hat{\mathbf{h}}_{\text{BERT}}\|_2^2$ exceeding threshold $\tau = 0.11$ (95th percentile on benign validation data) triggers analyst triage. Ablation confirms 0.4–0.6% classification impact, positioning this pathway as operational triage rather than primary detection.

F. Feature Fusion and Training

Feature concatenation forms a unified representation:

$$\mathbf{h}_{\text{concat}} = [\mathbf{h}_{\text{BERT}}; \mathbf{h}_{\text{char}}; \mathbf{h}_{\text{temporal}}; \mathbf{z}_{\text{anomaly}}] \in \mathbb{R}^{1856} \quad (5)$$

where $1856 = 768 + 192 + 768 + 128$. A fusion layer applies LayerNorm, ReLU, and dropout ($p = 0.3$), followed by two dense layers for final binary classification. SENTINEL optimizes a combined objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{recon}}, \quad \lambda = 0.1 \quad (6)$$

where $\mathcal{L}_{CE}$ uses class weights ($w_{\text{malicious}} = 8.08$, $w_{\text{benign}} = 0.53$) to address the 15:1 training imbalance. Training uses the AdamW optimizer with learning rate 2×10^{-5} , batch size 32, and early stopping with patience 3. Performance is stable across 5 random seeds: Volt Typhoon 92.0% $\pm$ 1.2%, obfuscated 91.2% $\pm$ 0.8%.

IV. EXPERIMENTAL EVALUATION

A. Dataset and Setup

Our dataset contains 7,049 commands: 6,613 benign and 436 malicious. Benign samples originate from enterprise Sysmon logs (3,847 commands from 50 workstations over 6 months), GitHub administrative scripts (1,523), and automated system tasks (1,243). Malicious samples include commands from documented APT29, APT32, Lazarus, and Turla campaigns (187) [22], red-team exercises using Empire and Cobalt Strike (134), and LOLBAS abuse cases (115). Data was split 70/15/15 (training: 4,934; validation: 1,058; testing: 1,057), with SHA-256 deduplication removing 312 exact duplicates and Levenshtein filtering ($>90\%$ overlap threshold) removing 89 near-duplicates. All Volt Typhoon samples were held out completely from training.

Volt Typhoon Evaluation Set ($n = 100$, balanced). Fifty malicious commands were manually extracted from Microsoft Threat Intelligence documentation [3] and CISA advisories [1] spanning eight attack phases: SSH key enumeration (6), browser profile discovery (4), remote access tool queries (12), credential harvesting (8), network reconnaissance (7), persistence (5), lateral movement (5), and exfiltration (3). Fifty benign commands using the same utilities (`reg query`, `dir`, `ntdsutil`) in legitimate administrative contexts were sourced from enterprise Sysmon logs matched by utility type and argument count. Two security analysts achieved Cohen’s $\kappa = 0.91$ inter-rater agreement.

Obfuscated Dataset ($n = 250$). Generated from test-split commands using eight deterministic transformation techniques: base64 encoding (40), string concatenation (35), backtick injection (30), alias substitution (35), case randomization (30), character encoding (25), format operators (30), and whitespace insertion (25).

Experiments used PyTorch 2.0.1, `bert-base-uncased`, and an NVIDIA Tesla T4 GPU. Baselines span classical ML (BoW, TF-IDF, Random Forest, XGBoost), deep learning (Char-CNN [6], Token-Char Fusion [7], BERT [4], BERT with obfuscation-aware augmentation), and specialized systems (PowerPeeler [15], LOLWTC [14]).

B. Comparative Results

Table I presents accuracy across all three evaluation sets. On imbalanced validation data, most models exceed 98% accuracy. The critical finding emerges on adversarial datasets: SENTINEL achieves 92.0% on Volt Typhoon and 91.2% on obfuscated inputs, while BERT drops to 74.0% and 72.0%, and LOLWTC collapses to 52.0% and 48.0%. The 8.0 percentage point gap over BERT with augmentation (84.0%) on Volt Typhoon confirms that multi-pathway architectural

integration provides robustness beyond data augmentation alone. Augmentation improves over vanilla BERT by 10.0 points, while the architecture contributes an additional 8.0 points that data diversity cannot recover. Fig. 2 visualizes accuracy across all three evaluation sets; the dramatic drop in baseline bars from the Validation to Volt Typhoon and Obfuscated columns confirms that the performance degradation is structural, not a tuning artifact.

TABLE I
ACCURACY ACROSS ALL EVALUATION DATASETS.

Model	Validation	Volt Typhoon	Obfuscated
TF-IDF	98.72%	70.0%	66.0%
Char-CNN [6]	96.2%	78.0%	82.4%
Token-Char Fusion [7]	97.4%	82.0%	85.2%
BERT [4]	99.29%	74.0%	72.0%
BERT + Augmentation	99.15%	84.0%	86.0%
PowerPeeler [15]	97.8%	81.0%	85.6%
LOLWTC [14]	93.83%	52.0%	48.0%
SENTINEL	99.43%	92.0%	91.2%

Volt Typhoon ($n = 100$) is balanced (50 benign + 50 malicious); validation is imbalanced (15:1). Validation accuracy masks per-class failures, as revealed in Fig. 4.

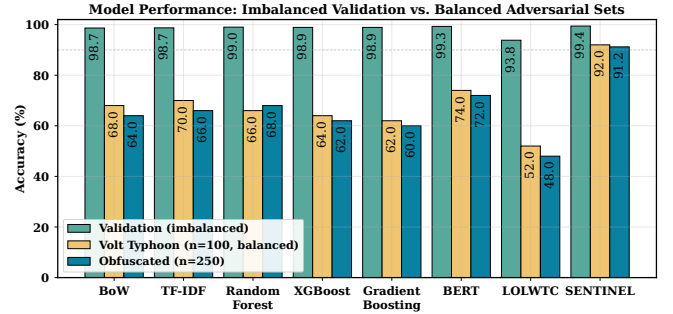


Fig. 2. SENTINEL shows strong and stable accuracy on validation, Volt Typhoon, and obfuscated tests, while baseline models drop sharply under adversarial conditions.

C. Structural Failure Analysis

Fig. 3 shows a radar-based evaluation across five architectural capability dimensions derived from the APT threat model, where each axis isolates a specific weakness and the overall area reflects structural robustness. These limitations arise from model design rather than data or tuning constraints.

Obfuscation Invariance: Lexical models (TF-IDF, LOLWTC) break under syntactic changes, while BERT’s subword tokenization fails on heavily obfuscated or encoded commands. In contrast, SENTINEL’s CharCNN operates at the character level, capturing robust patterns independent of tokenization.

Multi-Stage Recognition and Temporal Dispersion: Independent classifiers miss coordinated attack sequences, and LSTM struggles with long-range dependencies. SENTINEL’s inter-command attention models full sequence relationships, enabling stronger multi-stage understanding.

Distributional Shift: Standard classifiers misclassify unseen attack types without uncertainty modeling. SENTINEL’s

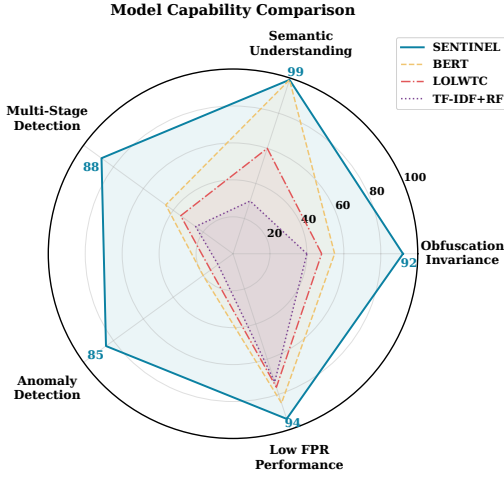


Fig. 3. Radar chart comparing SENTINEL against baselines across structural challenges.

anomaly detection module flags out-of-distribution inputs via reconstruction error, improving robustness to novel techniques.

D. Per-Class Analysis

Fig. 4 displays per-class precision and recall bars for each model on the balanced Volt Typhoon benchmark, directly exposing the fundamental detection failure hidden by overall accuracy. TF-IDF achieves 98.72% overall validation accuracy but only 50% malicious recall on the adversarial set, missing half of documented state-sponsored attack commands. BERT achieves 99.29% validation accuracy but 58% malicious recall. This failure arises because models trained on 15:1 imbalanced data converge to classify the dominant benign class accurately while sacrificing sensitivity to the rare malicious class. The consequence for defense operations is operationally unacceptable: a system that misses 42–56% of state-sponsored commands provides inadequate network protection regardless of its headline accuracy figure.

SENTINEL maintains balanced performance: 90% benign recall and 94% malicious recall on the Volt Typhoon set. For defense network monitoring, where missing a coordinated state-sponsored command sequence carries higher operational cost than a false alarm, malicious recall is the primary metric of operational consequence.

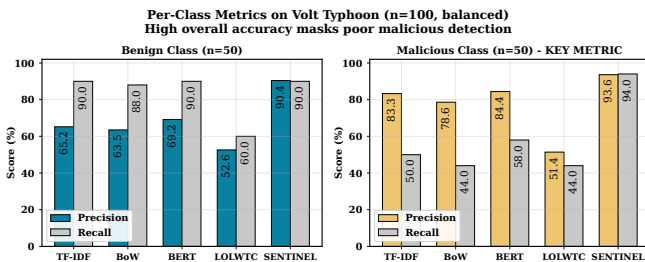


Fig. 4. Per-class performance comparison on Volt Typhoon, highlighting stronger malicious detection by SENTINEL.

E. Ablation Study

Fig. 5 shows grouped bars comparing full SENTINEL against three leave-one-out variants across all three datasets, quantifying each pathway’s independent contribution. Removing CharCNN degrades Volt Typhoon and obfuscated accuracy by 5.6 percentage points (91.6% to 86.0%), validating its role in syntactic invariance (McNemar’s $p < 0.001$, $n_{\text{eff}} = 50$). Disabling inter-command attention reduces Volt Typhoon accuracy by 2.0 percentage points, reflecting value for multi-stage pattern recognition despite training on only 355 malicious sequences. Removing anomaly detection has minimal classification impact (0.4–0.6%), confirming its role as operational triage for novel techniques rather than primary detection.

The CharCNN contribution is the largest single-pathway gain, confirming that character-level obfuscation invariance is the critical capability missing from all baseline architectures. The inter-command attention contribution is smaller in absolute accuracy terms but addresses a qualitatively distinct failure mode: recognizing coordinated attack progressions that no single-command classifier can detect by design.

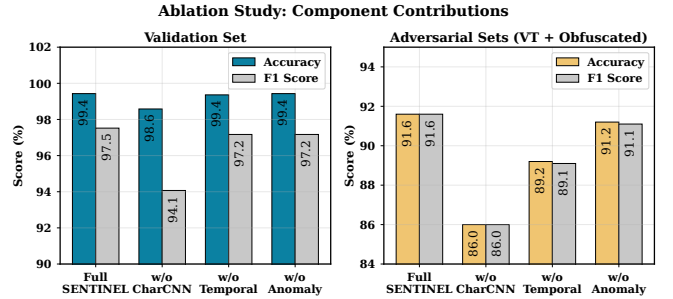


Fig. 5. Ablation study summarizing component contributions on the validation and adversarial evaluation sets. The CharCNN pathway produces the largest observed degradation under adversarial evaluation.

F. Low False-Positive Rate Performance

Production security operations center deployment requires low false-positive rates to prevent alert fatigue from overwhelming analysts [19]. Fig. 6 compares TPR at three low-FPR operating points; the curves show that SENTINEL maintains a higher true positive rate at every low-FPR operating point on the validation set. At $\text{FPR} = 0.1\%$ (one false alarm per 1,000 benign commands), SENTINEL maintains 81.8% TPR versus 69.7% for BERT, a 12.1 percentage point improvement. This gain reflects CharCNN’s ability to maintain high-confidence classification scores on obfuscated variants that fragment BERT’s tokenization into diffuse probability distributions near the decision boundary.

G. Limitations

Platform Scope. This work addresses Windows command-line LOTL exclusively; Linux-based LOTL using `curl`, `python3`, and shell one-liners presents different challenges for which no annotated benchmark at LOLBAS fidelity yet exists, leaving cross-platform extension to future work.

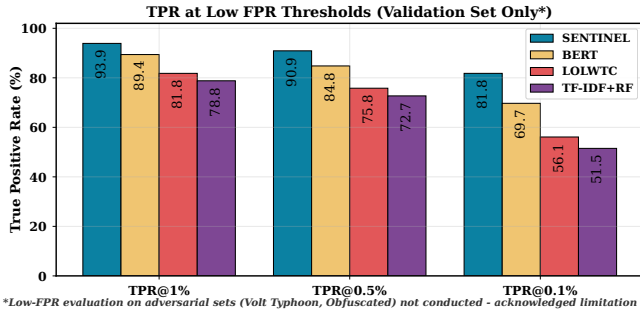


Fig. 6. True positive rate (TPR) versus false positive rate (FPR) curves on the validation set show that SENTINEL consistently outperforms baselines at low-FPR operating points, which are critical for SOC deployment. At 0.1% FPR, SENTINEL achieves 81.8% TPR, compared to 69.7% for BERT and 56.1% for LOLWTC.

Dataset Scale. With an effective Volt Typhoon test size of $n_{\text{eff}} \approx 50$ after accounting for template dependencies, point estimates carry wide 95% confidence intervals ([84.8%, 96.5%]), and 24% of malicious samples share semantic patterns with LOLBAS training data, potentially inflating generalization estimates. Performance on enterprise-scale telemetry remains unvalidated.

Temporal Scope. The inter-command attention module operates within fixed 1-hour windows, capturing tactical progressions only; strategic phases separated by days or weeks, as in Volt Typhoon, fall outside this scope and would require hierarchical attention or external memory.

Anomaly Calibration. The 95th-percentile threshold ($\tau = 0.11$) flags roughly 5% of benign commands, translating to thousands of daily triage alerts at enterprise scale without recalibration. Anomaly scoring should therefore be deployed as an optional analyst-triage tier with independent FPR/TPR calibration.

V. CONCLUSION

We present SENTINEL, an end-to-end framework for detecting Windows LOTL attacks that jointly addresses obfuscation, semantic discrimination, multi-stage temporal reasoning, and distribution shift. Evaluations on the Volt Typhoon dataset highlight that standard benchmark setups can mask critical weaknesses in baseline models, despite apparently strong overall accuracy. Ablation studies further show that each component targets a distinct failure mode, with the full architecture consistently outperforming all individual variants and ablated configurations. These results suggest that robust LOTL detection requires explicitly modeling both syntactic and temporal structure rather than relying on single-view representations. Future work includes extending the framework to Linux LOTL via GTFEBins-based behaviors, improving campaign-level temporal modeling for long-horizon attack tracking, and incorporating richer system and process context signals to support more reliable SOC deployment.

ACKNOWLEDGMENT

This work is supported by the U.S. Department of Energy (DOE), under Award number DE-NA0004189.

REFERENCES

- [1] Cybersecurity and Infrastructure Security Agency, “PRC State-Sponsored Cyber Actor Living off the Land to Evade Detection,” CISA, Tech. Rep. AA23-144A, 2023.
- [2] F. Barr-Smith, X. Ugarte-Pedrero, M. Graziano, R. Spolaor, and I. Martinovic, “Survivalism: Systematic analysis of windows malware living-off-the-land,” in *Proc. IEEE Symp. Security and Privacy (SP)*, 2021, pp. 1557–1574.
- [3] Microsoft Threat Intelligence, “Volt Typhoon targets US critical infrastructure with living-off-the-land techniques,” Microsoft Security Blog, May 2023. [Online]. Available: <https://www.microsoft.com/en-us/security/blog/2023/05/24/volt-typhoon-targets-us-critical-infrastructure-with-living-off-the-land-techniques/>
- [4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North American Chapter Assoc. Computational Linguistics: Human Language Technologies (NAACL-HLT)*, vol. 1, 2019, pp. 4171–4186.
- [5] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for online network intrusion detection,” in *Proc. Network and Distributed System Security Symp. (NDSS)*, 2018.
- [6] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
- [7] D. Hendler, S. Kels, and A. Rubin, “Detecting malicious PowerShell commands using deep neural networks,” in *Proc. Asia Conf. Computer and Communications Security (AsiaCCS)*, 2018, pp. 187–197.
- [8] Z. Li, Q. A. Chen, C. Xiong, Y. Chen, T. Zhu, and H. Yang, “Effective and light-weight deobfuscation and semantic-aware attack detection for PowerShell scripts,” in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, 2019, pp. 1831–1847.
- [9] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly detection and diagnosis from system logs through deep learning,” in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, 2017, pp. 1285–1298.
- [10] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.
- [11] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, “SecureBERT: A domain-specific language model for cybersecurity,” in *Proc. Int. Conf. Security and Privacy in Communication Systems*, 2022, pp. 39–56.
- [12] H. Guo, S. Yuan, and X. Wu, “LogBERT: Log anomaly detection via BERT,” in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2021, pp. 1–8.
- [13] LOLBAS Project Contributors, “Living Off The Land Binaries and Scripts (LOLBAS),” 2024. [Online]. Available: <https://lolbas-project.github.io/>
- [14] K. Ding, S. Zhang, F. Yu, and G. Liu, “LOLWTC: A deep learning approach for detecting living off the land attacks,” in *Proc. IEEE 9th Int. Conf. Cloud Computing and Intelligent Systems (CCIS)*, 2023, pp. 176–181.
- [15] R. Li, C. Zhang, H. Chai, L. Ying, H. Duan, and J. Tao, “PowerPeeler: A precise and general dynamic deobfuscation method for PowerShell scripts,” in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, 2024, pp. 4539–4553.
- [16] T. Ongun, J. W. Stokes, J. Bar Or, K. Tian, F. Tajaddodianfar, J. Neil, C. Seifert, A. Oprea, and J. C. Platt, “Living-off-the-land command detection using active learning,” in *Proc. 24th Int. Symp. Research in Attacks, Intrusions and Defenses (RAID)*, 2021, pp. 442–455.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.
- [18] MITRE Corporation, “MITRE ATT&CK Framework,” 2024. [Online]. Available: <https://attack.mitre.org/>
- [19] Y. Liu, X. Shu, Y. Sun, J. Jang, and P. Mittal, “RAPID: Real-time alert investigation with context-aware prioritization for efficient threat discovery,” in *Proc. 38th Annu. Computer Security Applications Conf. (ACSAC)*, 2022, pp. 827–840.
- [20] Palo Alto Networks Unit 42, “Threat Brief: Attacks on Critical Infrastructure Attributed to Insidious Taurus (Volt Typhoon),” Palo Alto Networks, Tech. Rep., 2024. [Online]. Available: <https://unit42.paloaltonetworks.com/volt-typhoon-threat-brief/>

- [21] S.-Y. Huang, C.-L. Yang, C.-Y. Lin, and C.-Y. Huang, "CmdCaliper: A semantic-aware command-line embedding model and dataset for security research," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2024, pp. 20188–20206.
- [22] A. Saha, J. Mattei, J. Blasco, L. Cavallaro, D. Votipka, and M. Lindorfer, "Expert insights into advanced persistent threats: Analysis, attribution, and challenges," in *Proc. 34th USENIX Security Symp.*, 2025, pp. 2185–2204.