Graph-Based Detection of Malicious Windows Scheduled Tasks for Advanced Persistent Threats

Khaled Rahal [1 2] Ahmed Bokri [1] Arbia Riahi [1] Thibault Debatty [1] DRICOT Jean Michel[2]

[1]Royal Military Academy, Brussels, Belgium , [2]Université Libre de Bruxelles, Brussels, Belgium

Motivation

Problem statement:

- Scheduled Tasks are abused for persistence: they are native, reliable, and blend into legitimate administrative activity.
- Static rules and flat ML lose execution context; generic provenance graphs don't treat tasks as first-class targets.

Goal: Detect malicious tasks from their typed neighbourhood: processes, files, registry keys, accounts, lifecycle events.

Contributions: Task-centric graph construction · Deterministic Sysmon-Task Scheduler correlation · HTGNN with temporal/relation modelling · Zero-shot CYRST evaluation · Neo4j triage.

GRAST Overview

GRAST (Graph-based Representation for Anomalous Scheduled Tasks) models Windows Scheduled Task abuse as a node-classification problem on a heterogeneous graph:

- Task nodes are classification targets.
- Sysmon & Task Scheduler logs are correlated.
- Typed edges encode semantic relations.
- HTGNN predicts maliciousness at task-node level.
- Predictions are projected into Neo4j for triage.

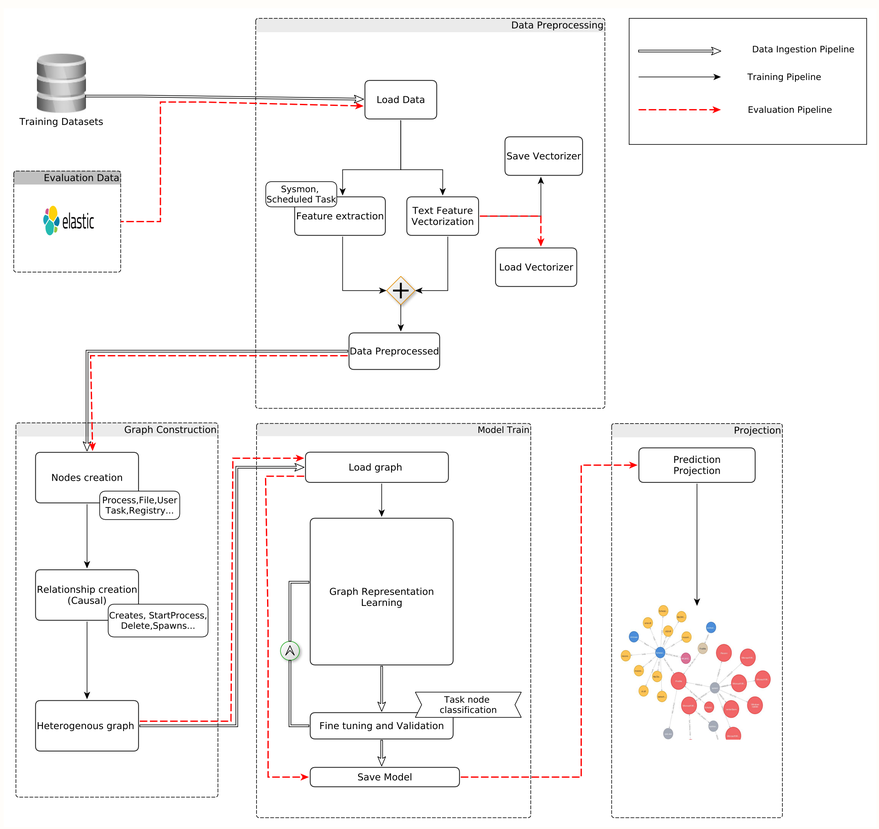


Figure 3: Overall architecture

Graph Model

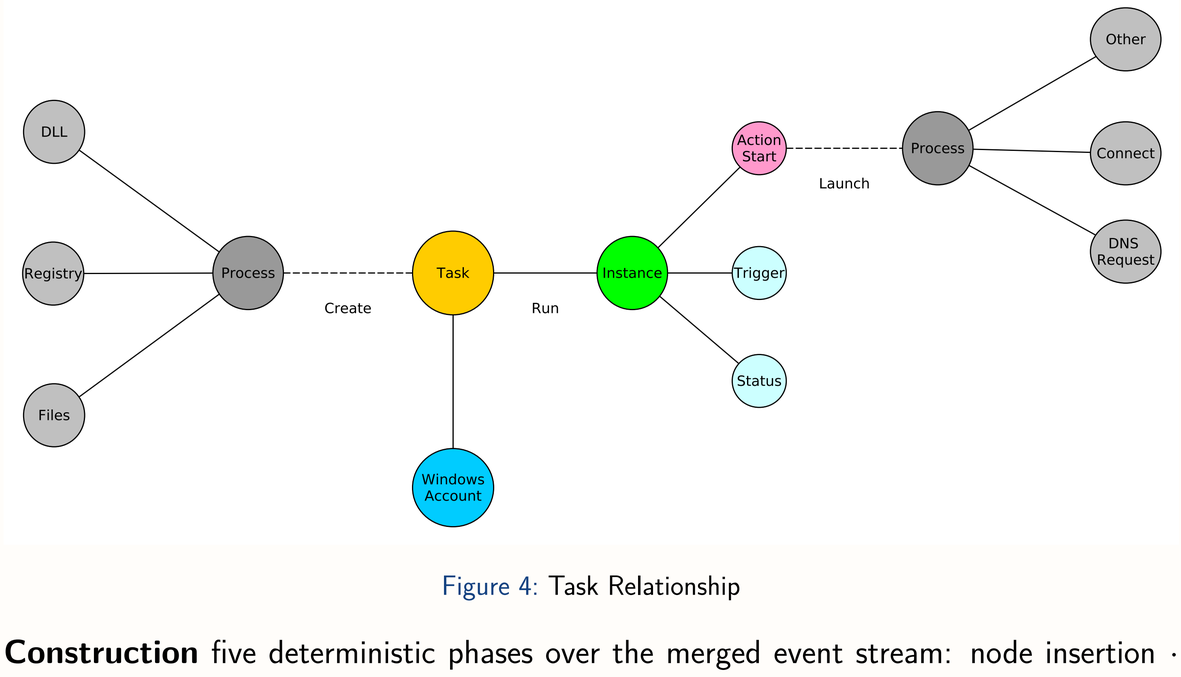


Figure 4: Task Relationship

Construction five deterministic phases over the merged event stream: node insertion · lifecycle enrichment · execution wiring · Sysmon correlation · artifact linking.

Experimental

PTD (Persistence Techniques Datasets).
120 Windows VMs (8/10/11/Server), APT-style WST persistence per MITRE T1053.005.

CYRST (CyberRange Datasets).
KYPO CyberRange, real Malware samples incl. the Tarrask/HAFNIUM implant. Zero-shot, no CYRST data used in training or tuning.

Results

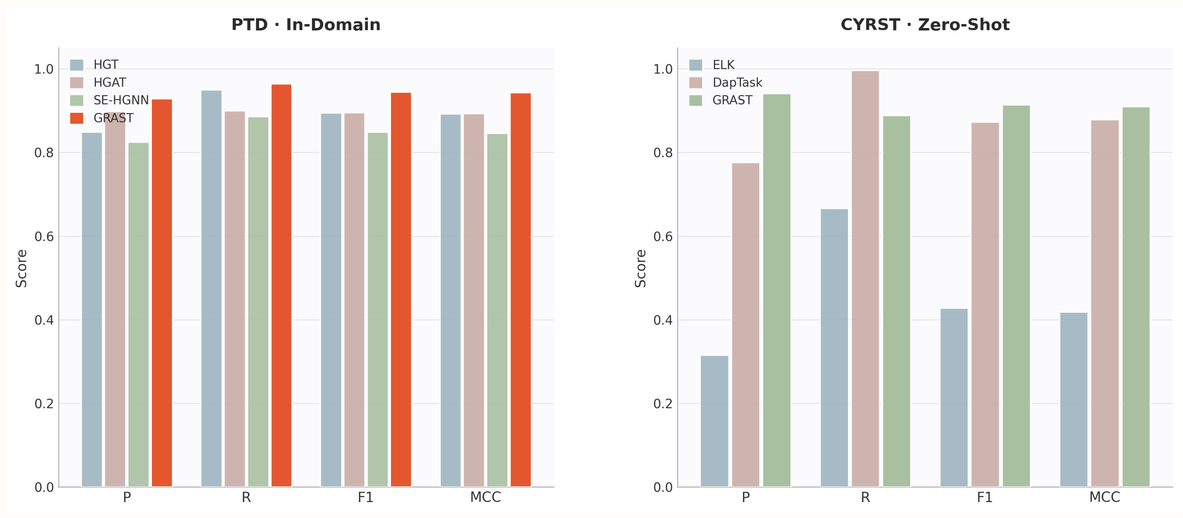


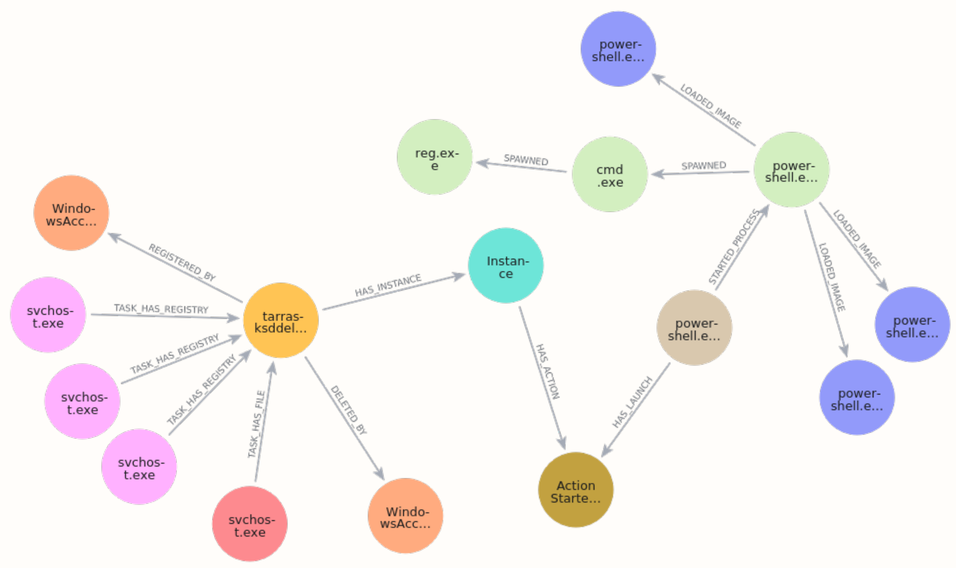
Figure 5: Performance Comparison: GRAST vs. Baselines

Neo4j Triage

Predictions are exported to Neo4j as inspectable subgraphs.

Recoverable evidence: registering account, execution instance, task-launched process, registry stealth artifacts, file/DLL/DNS/network activity, shared infrastructure.

Operational value: converts scores into graph evidence, supports short Cypher traversals, enables alert clustering, reduces manual SIEM correlation.



Conclusion and Future Work

- **Task-Centric Graph Modeling:** Captures the full system execution context.
- **HTGNN Performance:** Outperforms both heterogeneous GNN and flat baselines.
- **Analyst-in-the-Loop Triage:** Enabled by Neo4j projection, direct intervention in the triaging process.
- **Future Work:** Extend the task-centric graph to other persistence mechanisms (e.g., services, WMI subscriptions, registry autoruns).

References

[1] PTD Dataset: K. Rahal, A. Riahi and T. Debatty, "Dataset of APT Persistence Techniques on Windows Platforms Mapped to the MITRE ATTCK Framework, 2025, pp. 17-24, doi: 10.1109/ICIN64016.2025.10943025.

[2] Daptask: Khaled Rahal et al. Uncovering Malicious Persistence: Machine Learning-Based Detection of Windows Scheduled Tasks. Advances in Knowledge-Based Systems, Data Science, and Cybersecurity.Research 2025;2(3):16.

Cyber Defence Lab
Royal Military Academy
www.cylab.be