LLM-Based Autonomous Agents for Dynamic Malware Analysis

RQ: To what extent can Qwen3-4B distinguish between benign and malicious Windows executables using reduced dynamic-analysis reports?

AUTHORS

Thomas Crull – BSc CSE TU Delft

AFFILIATIONS

Delft University of Technology
Prof. Przemysław Pawełczak – Embedded Systems Group
Prof. Soham Chakraborty – Programming Languages Group
Prof. Arie van Deursen – Software Engineering Group

INTRODUCTION

Dynamic malware analysis entails running a piece of software in a sandbox environment, and confirming whether it is malicious or not based on the behaviours it displays.

The main types of dynamic malware analysis:

- Process analysis – what processes the sample creates
- Memory analysis – what interactions the sample has with the registers and memory
- Network analysis – what network connections the sample tries to create with outside entities

To catch these behaviours, analysts typically employ a suite of monitoring tools. Most such tools create large and noisy logs, which can be difficult and time consuming to parse even for an expert in the field.

AIM

Evaluate whether dynamic analysis logs can be interpreted by an LLM agent to distinguish malicious from benign executable samples, and assess whether this could support existing malware-analysis workflows.

CONTRIBUTION

- A working sandbox-to-LLM analysis pipeline
- Evaluation of LLM malware classification
- Insight into whether LLM agents can reduce manual log interpretation
- Benign and malware dataset

RELATED LITERATURE

Existing work shows that LLMs can assist with malware-related tasks such as memory forensics, malicious network-traffic detection and static analysis. However, there are no studies on the performance of LLMs on dynamic malware analysis, especially in resource constrained environments.

METHODOLOGY

Malware and benign samples are executed in an isolated sandbox while malware analysis tools record their behavior. The resulting logs are extracted, filtered for dynamic information and reduced to fit a limited context window, then given to an LLM-based agent, which classifies each sample as malicious or benign. These predictions are then compared with the dataset labels to evaluate the agent's accuracy, precision, recall and F1-score.

Setup:

- *Host machine: Lenovo laptop, Intel® Core™ Ultra 7 265H, 64 GB RAM.*
- *VMware Linux VM: Ubuntu 24.04 with 32 GB RAM and 600 GB storage.*
- *Sandbox orchestrator: CAPEv2 running on the Ubuntu VM.*
- *Detonation VM: Windows 10 Pro 22H2 on KVM\QEMU with 8 GB RAM and 100 GB storage.*
- *LLM agent: Qwen3-4B run on a DelftBlue A100-small node.*

***Workflow**: CAPEv2 executes samples in the Windows VM, collects behavioural logs, and resets the VM after analysis. The logs are preprocessed and then passed to Qwen for classification.*

*The **malware** samples which were used for testing are from the MalwareBazaar dataset, numbering 1082.*

*The **benign** samples which were used for testing are from a dataset which we built ourselves, as there are no state-of-the-art datatsets for benign executables. The executables were collected from PortableApps and their installers, the Sysinternals Suite and from the Benign-NET dataset, totaling 762 samples.*

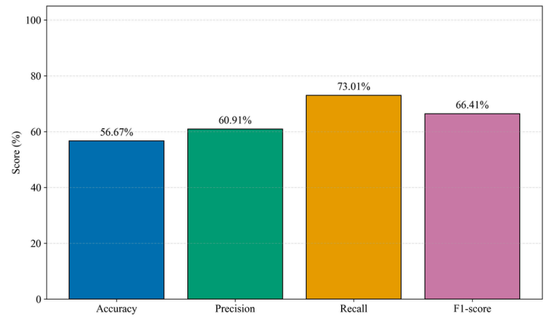
CONCLUSION

Overall, Qwen3-4B displays some potential for malware classification, but cannot be integrated in current cybersecurity systems without fine-tuning or a better setup, since it displays high false-positive rates. Qwen was inclined to err on the side of caution and classify benign executables as malicious whenever faced with inconclusive evidence, likely due to the protective features of the model. We tried to solve this issue with better prompt engineering and adding inconclusive as a possible verdict, but the former only marginally improved the results, where as the latter made them worst.

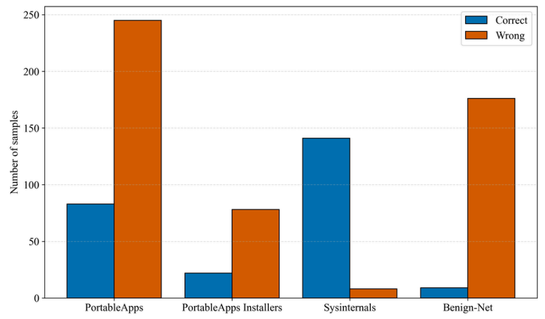
The model's user input classification pattern signifies a possible vulnerability of LLMs. Further research should test if other LLMs also tend to classify user input requests as clear benign evidence.

RESULTS

The accuracy outlined by the model is misleading, as it had drastically better results on the malware samples than the benign. This can be seen from the high recall score and low precision rate.



An unexpected find arose from the accuracy of the model on each benign dataset category. The model had overwhelmingly positive results on the Sysinternals Suite compared to the other categories. This was unexpected since this category contains powerful Windows administration tools which display behaviour closer to malware than the rest of the samples. Nevertheless, Qwen was inclined to classify them as benign since most requested user keyboard input, which it saw as clear benign evidence. This might mean that malware that also requests user input would bypass Qwen's detection.



TU Delft Delft University of Technology