Automating Detection Workflows with OSINT YARA Rules

Abderrahim Mayaba
ammayaba@gmail.com

Abstract—Zero-day disclosures frequently outpace vendor-provided detection content, leaving organizations exposed during critical windows where manual triage and rule integration introduce delays. To reduce mean time to coverage and eliminate human bottlenecks, this report presents an automated workflow for extracting, combining, and executing open-source YARA threat detection rules on containerized environments. The pipeline dynamically fetches rules from Abuse.ch YARAify, filters out classified entries, exports target containers or paths using docker cp or filesystem export, and performs recursive scanning in a single execution step. This end-to-end automation eliminates manual intervention, improves responsiveness to emerging threats, and enables scalable, reproducible detection operations.

I. Context

In modern security operations, the threat landscape evolves faster than traditional vendor-driven detection mechanisms can respond. Zero-day vulnerabilities and rapidly weaponized exploits are often publicly disclosed or observed in the wild before commercial detection signatures become available. When detection workflows rely on manual rule collection, validation, and deployment, analysts face critical delays that increase exposure time and risk.

Open-source threat intelligence sources such as Abuse.ch provide timely YARA rules that can capture emerging malware patterns ahead of vendor coverage. However, without automation, integrating these rules into containerized environments for proactive scanning becomes labor-intensive and error-prone. This report addresses that operational gap by leveraging automation to continuously fetch, consolidate, and execute YARA rules at scale, enabling accelerated threat detection with minimal human intervention.

II. Driver

The primary driver for this work is the operational gap between threat disclosure and real-world detection coverage. Security teams often rely on vendor-delivered signatures or manual YARA rule deployment, creating a latency window in which adversaries can operate undetected. This delay is further amplified when detection content must be manually fetched, reviewed, and executed against multiple containerized environments.

To eliminate this bottleneck, I sought to develop an automated system that continuously pulls emerging open-source YARA rules, applies them directly to containerized workloads, and returns actionable matches with zero human intervention. The goal is to reduce mean time to detection (MTTD) and ensure immediate coverage for newly disclosed threats—before attackers exploit them in the wild.

III. Data Source

This workflow uses community-contributed YARA rules retrieved dynamically from the Abuse.ch YARAify platform, providing real-time coverage for emerging threats. The scan target is a containerized environment, where files are extracted using docker cp or filesystem export for inspection.

IV. Snippets & Repository

This workflow is implemented in Python and executed within a Docker environment using the YARA engine for malware detection. All source code and the report are available in the repository:

Repository:https://github.com/mayaba/automated-yara-container-scanner

V. What Could Be Enhanced

While the current workflow provides full automation from rule retrieval to container scanning, several enhancements could further improve its operational value:

- SIEM/SOAR Integration: Automatically forward scan results to detection platforms for alerting and correlation.
- Parallel Scanning: Enable concurrent scans across multiple containers or paths to improve scalability.
- Rule Versioning: Track changes in YARA rules over time to support differential analysis and rollback.
- Reporting Output: Generate structured JSON or HTML reports suitable for dashboards and threat intelligence feeds.

VI. Summary

This work demonstrates an automated approach to threat detection that bridges the gap between zero-day disclosures and vendor-delivered signatures. By dynamically retrieving community YARA rules and applying them directly to containerized environments, the workflow enables immediate coverage with minimal manual effort. This lightweight automation reduces exposure time, improves operational efficiency, and provides a scalable foundation for proactive threat hunting.