

PowerSploit

github.com/PowerShellMafia/PowerSploit

Scanned 18 September 2026 at 19:44 · Report generated 29 October 2026 · PowerShell · 13,084 stars

OVERALL ASSESSMENT

Medium

Combined score 51 · 11 findings after filtering

Summary

This repository shows two major red flags that indicate it contains malicious code rather than just poor security practices. One finding was flagged by ClamAV with a high confidence score of 98%, which means the system found a known pattern used to execute malicious scripts on Windows machines. Another finding matched a YARA rule with a low confidence score of 36%, suggesting that while some suspicious code exists, it is not as certain as the first issue. These results are significant because they point directly to tools designed for privilege escalation and malware injection, which are dangerous for any developer working with this code. You should treat this repository as unsafe until you verify that all files have been removed or properly sanitized by a security expert who understands these specific threats.

Written by a language model running locally on the scanning machine, from the findings in this report. The risk levels above and below were decided by the scoring model, not by this summary.

Risk, by type

MalDet scores two different questions separately, because they call for different responses: code can be badly written without being hostile, and hostile code is not always badly written.

VULNERABILITY

Safe (0)

Insecure coding an attacker could exploit. Usually mistakes, and usually fixable in place.

MALICIOUS PATTERN

High (51)

Signs the code may be trying to do harm: hidden payloads, credential access, persistence, exfiltration.

What each detector found

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
yara	Signatures of malicious behaviour in source files	8	0	0	8

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
clamav	Known malware signatures	3	0	0	3

3 of 11 findings were rated more likely genuine than noise by the classifier.

Evidence

Identical findings are grouped, so a single rule firing many times takes one row. Ordered worst first. “Seen” is how many times that finding occurred.

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	clamav	ClamAV detected: Win.Countermeasure.DotNetToJScript-9804890-0	1	/Privesc/ PowerUp.ps1
HIGH	clamav	ClamAV detected: Win.Trojan.PSempireInj-7013548-0	2	/Exfiltration/ Invoke-CredentialInjection.ps1
HIGH	yara	YARA rule matched: detect_process_injection — Windows process-injection API sequence	6	/Exfiltration/ Invoke-Mimikatz.ps1
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	2	/Privesc/ PowerUp.ps1

How to read this report

- **This is a static scan.** The code was read, never executed. Behaviour that only appears at runtime is outside what any of these detectors can see.
- **Findings are not all equally real.** A classifier rates each finding on how likely it is to be genuine rather than noise, and that rating weights the scores. Measured against a hand-reviewed sample it reached precision 0.52 and recall 0.62 — useful for ranking, not a verdict on any single finding.
- **A high score is not proof of malice.** Security tools, penetration-testing frameworks and malware research repositories contain the same patterns as real malware by design. The two risk types above exist to keep that distinction visible.
- **Absence of findings is not proof of safety.** A detector that is not installed, or a language none of them parse, produces silence rather than a warning.
- **This scan is a snapshot.** The repository may have changed since 18 September 2026.