

nishang

github.com/samratashok/nishang

Scanned 07 October 2026 at 03:30 · Report generated 29 October 2026 · PowerShell · 10,130 stars

OVERALL ASSESSMENT

Medium

Combined score 45 · 20 findings after filtering

Summary

The repository has two distinct security risks: a low vulnerability score of 5.0 indicates that the code contains some insecure patterns, but these are likely just standard mistakes rather than intentional attacks. However, the high malicious-pattern score of 40.0 is concerning because it suggests the code may be intentionally designed to bypass security checks or hide malicious behavior. The most serious findings involve PowerShell scripts that execute dangerous commands like remote shell access and process injection, which are strong indicators of malware rather than routine development errors. While some rules flagged as potentially normal were actually used in legitimate applications, the presence of these specific high-confidence detections means the code should not be trusted without further investigation.

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

Low (5)

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

MALICIOUS PATTERN

High (40)

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	16	0	0	16

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
clamav	Known malware signatures	3	0	0	3
semgrep	Insecure coding patterns across many languages	1	0	0	1

4 of 20 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.Exploit.Powershell-1	1	/ Execution /Execute-DNSTXT-Code.ps1
HIGH	clamav	ClamAV detected: Win.Trojan.PowershellReverseShellOneLine-9840826-0	1	/Shells/Invoke-PowerShellTcpOneLine.ps1
HIGH	clamav	ClamAV detected: Xml.Malware.Squiblydoo-6728833-0	1	/Client/Out-SCT.ps1
HIGH	semgrep	Generic API Key detected	1	/ Prasadhak /Invoke-Prasadhak.ps1:16
HIGH	yara	YARA rule matched: detect_process_injection — Windows process-injection API sequence	2	/Gather/Invoke-Mimikatz.ps1
HIGH	yara	YARA rule matched: detect_anti_analysis — Checks for a debugger, VM, or sandbox before continuing - a common malware evasion technique, essentially never present in ordinary application code	2	/ powerpreter/Powerpreter.psm1
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	12	/Gather/Invoke-SessionGopher.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 07 October 2026.