

PayloadsAllTheThings

github.com/swisskyrepo/PayloadsAllTheThings

Scanned 18 September 2026 at 19:54 · Report generated 29 October 2026 · Python · 80,960 stars

OVERALL ASSESSMENT

Critical

Combined score 615 · 151 findings after filtering

Summary

This repository has been flagged as critical because it contains 151 specific security issues that could be exploited by attackers, with two separate scores indicating high severity: a vulnerability risk score of 465.0 and a malicious-pattern risk score of 151.0. The most serious problems involve command injection vulnerabilities where untrusted user input is used to run arbitrary shell commands, which allows an attacker to execute code on the server. Additionally, there are confirmed instances of reverse shells, credential harvesting attempts, and the use of dangerous tools like phpinfo that expose sensitive system information. While some findings appear to be standard insecure coding practices rather than intentional malicious behavior, the presence of these patterns suggests the codebase is not safe for production use without significant refactoring.

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

Critical (465)

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

MALICIOUS PATTERN

Critical (151)

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
semgrep	Insecure coding patterns across many languages	55	24	2	81

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
yara	Signatures of malicious behaviour in source files	39	0	13	52
bandit	Insecure coding patterns in Python source	0	3	6	9
clamav	Known malware signatures	9	0	0	9

63 of 151 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	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	4	/Upload Insecure Files/ Extension PHP/ shell.php:1
HIGH	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	2	/Upload Insecure Files/ Extension PHP/ shell.png^ shell.php:1
HIGH	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	1	/Upload Insecure Files/ Extension PHP/ shell.jpeg. php:1
HIGH	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	1	/Upload Insecure Files/ Extension PHP/ shell.jpg^ shell.php:1

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Executing non-constant commands. This can lead to command injection. You should use `escapeshellarg()` when using command.	8	/Upload Insecure Files/ Extension PHP/ shell.png.php:1
HIGH	semgrep	Semgrep found a bash reverse shell	1	/.github/hopla_config.json:1129
HIGH	semgrep	linux shadow file detected	8	/Command Injection/ README.md:287
HIGH	semgrep	Google OAuth Access Token detected	2	/Server Side Request Forgery/ SSRF-Cloud-Instances.md:210
HIGH	clamav	ClamAV detected: Img.Phishing.SvgJsPhishing-10044283-0	6	/XSS Injection/ README.md
HIGH	clamav	ClamAV detected: Img.Exploit.CVE_2021_22204-9911806-0	2	/Upload Insecure Files/ Picture Metadata/ CVE-2021-22204_exiftool_revshell.jpg
HIGH	clamav	ClamAV detected: Eicar-Signature	1	/Upload Insecure Files/ EICAR/ eicar.txt
HIGH	semgrep	Generic API Key detected	1	/Insecure Source Code Management/ Files/ github-dorks.txt:257

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Insecure WebSocket Detected. WebSocket Secure (wss) should be used for all WebSocket connections.	1	/Web Sockets/ README.md:110
HIGH	semgrep	The 'phpinfo' function may reveal sensitive information about your environment.	3	/Upload Insecure Files/ Extension PHP/ phpinfo.php tml:1
HIGH	semgrep	JWT token detected	15	/JSON Web Token/ README.md:308
HIGH	semgrep	Executing non-constant commands. This can lead to command injection.	8	/Upload Insecure Files/ Extension PHP/ shell.jpg.php:1
HIGH	yara	YARA rule matched: detect_reverse_shell — Reverse shell command patterns	13	/Upload Insecure Files/ Picture ImageMagick / imagetrage k2_ubuntu_shell2.jpg
HIGH	yara	YARA rule matched: detect_credential_harvester — References credential/secret material by name AND a deliberate outbound-send API in the same file - narrower than \"reads any env var\", which large legitimate codebases do constantly regardless of intent. Still not proof the two are connected (YARA c	1	/File Inclusion/ LFI-to-RCE.md
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	3	/File Inclusion/ Intruders/ List_Of_Fi le_To_Incl ude.txt

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	4	/File Inclusion/ Wrappers.md
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	4	/Command Injection/ Intruder/ command_exec.txt
HIGH	yara	YARA rule matched: detect_obfuscated_string_execution — Reconstructs a string character-by-character or from hex escapes, then immediately executes it - fromCharCode/chr() alone are common in ordinary text-processing code, so this requires both the reconstruction and an execution call together	10	/XSS Injection/ README.md
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	4	/File Inclusion/ Intruders/ Linux-files.txt
MEDIUM	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	2	/Upload Insecure Files/ Extension PHP/ shell.jpg^ shell.php:1
MEDIUM	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	2	/Upload Insecure Files/ Extension PHP/ shell.png.php:1

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	2	/Upload Insecure Files/ Extension PHP/ shell.jpg. php:1
MEDIUM	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	1	/Upload Insecure Files/ Extension PHP/ shell.phtml :1
MEDIUM	semgrep	Untrusted input might be injected into a command executed by the application, which can lead to a command injection vulnerability. An attacker can execute arbitrary commands, potentially gaining complete control of the system. To prevent this vulnerability, avoid executing OS commands with user input	1	/Upload Insecure Files/ Extension PHP/ shell.jpeg. php:1
MEDIUM	semgrep	User input is passed to a function that executes a shell command. This can lead to remote code execution.	8	/Upload Insecure Files/ Extension PHP/ shell.jpeg. php:1
MEDIUM	semgrep	GitHub Actions step uses a mutable tag or branch reference. Tags and branch names can be silently repointed by the action owner, enabling supply-chain attacks — as seen in the trivy-action and kics-github-action compromises. Pin the reference to a full 40-character commit SHA instead, e.g. `uses: ac	6	/.github/ workflows/ mkdocs- build.yml:1 6

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted	2	/.github/overrides/main.html:17
MEDIUM	bandit	Call to requests without timeout	3	/File Inclusion/Files/uploadlfi.py:17
LOW	semgrep	Detected a request using 'http://'. This request will be unencrypted, and attackers could listen into traffic on the network and be able to obtain sensitive information. Use 'https://' instead.	2	/File Inclusion/Files/uploadlfi.py:11
LOW	yara	YARA rule matched: detect_downloader — Downloads a remote resource - common and low-signal alone; see detect_download_and_execute for the composite version	13	/Server Side Request Forgery/README.md
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	6	/Upload Insecure Files/CVE FFmpeg HLS/gen_xbin_a vi.py:148

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.