

Empire

github.com/BC-SECURITY/Empire

Scanned 18 September 2026 at 19:47 · Report generated 29 October 2026 · PowerShell · 5,305 stars

OVERALL ASSESSMENT

Critical

Combined score 834 · 444 findings after filtering

Summary

This repository has been flagged as critical because it contains both severe insecure coding patterns and clear signs of malicious intent, resulting in a vulnerability score of 593.0 and a malicious-pattern score of 241.0. The most alarming issues involve the use of curl piped directly into shell commands like bash, which allows attackers to easily install malware if the code is compromised or the URL is hijacked. Additionally, the code contains multiple instances of ClamAV detections for known malicious agents and tools, such as LinPEAS and DotNetToJScript, indicating that the repository likely hosts active exploits or malware rather than just standard bugs. While some findings like using Jinja2 templates with autoescape disabled could lead to cross-site scripting (XSS), these are treated as ordinary insecure coding practices rather than intentional attacks.

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 (593)

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

MALICIOUS PATTERN

Critical (241)

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	37	130	0	167

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	9	48	77	134
yara	Signatures of malicious behaviour in source files	76	0	21	97
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	2	28	30
clamav	Known malware signatures	16	0	0	16

159 of 444 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	A `run:` step pipes the output of `curl` or `wget` directly into a shell interpreter. This is the "curl bash" install pattern — if the remote server is compromised or the URL is hijacked, an attacker can execute arbitrary code in your CI runner. Consider downloading the file first, verifying its c	5	/.github/workflows/release-private-start.yml:41
HIGH	clamav	ClamAV detected: Win.Trojan.Generic-10024536-0	4	/empire/server/data/misc/Reflectiv ePick_x86_ orig.dll
HIGH	clamav	ClamAV detected: Osx.Malware.Agent-6494195-0	1	/empire/server/data/misc/apptempla teResourc es/x86/ launcher. app/ Contents/ MacOS/ launcher
HIGH	clamav	ClamAV detected: Osx.Malware.Agent-6509323-0	1	/empire/server/data/misc/machotemp late

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	clamav	ClamAV detected: Osx.Malware.Agent-6537989-0	1	/empire/ server/ data/misc/ apptempla teResourc es/empty/ macho
HIGH	clamav	ClamAV detected: Osx.Malware.Agent-7367901-0	1	/empire/ server/ data/misc/ apptempla teResourc es/x64/ launcher. app/ Contents/ MacOS/ launcher
HIGH	clamav	ClamAV detected: Unix.Tool.LinPEASS-10004077-0	1	/empire/ server/ data/ module_so urce/ python/ situation al_aware ness/ LinPEAS.py
HIGH	clamav	ClamAV detected: Win.Countermeasure.DotNetToJScript-9804890-0	1	/empire/ server/ data/ module_so urce/ privesc/ PowerUp.p s1
HIGH	clamav	ClamAV detected: Win.Exploit.CVE_2017_0144-7404604-0	1	/empire/ server/ data/ module_so urce/ lateral_m ovement/ Invoke- SMBExec.p s1

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	clamav	ClamAV detected: Win.Exploit.PowerSploit-6983656-2	1	/empire/ server/ data/ module_source/ management /Invoke-ReflectivePEInjection.ps1
HIGH	clamav	ClamAV detected: Win.Trojan.BabySharkPS1-3-7404875-2	1	/empire/ server/ data/ module_source/ collection /Get-Keystrokes.ps1
HIGH	clamav	ClamAV detected: Win.Trojan.Powercat-9840812-0	1	/empire/ server/ data/ module_source/ management / powercat.ps1
HIGH	clamav	ClamAV detected: Win.Trojan.PSempireInj-7013548-0	2	/empire/ server/ data/ module_source/ credentials/Invoke-CredentialInjection.ps1
HIGH	bandit	Using jinja2 templates with autoescape=False is dangerous and can lead to XSS. Use autoescape=True or use the select_autoescape function to mitigate XSS vulnerabilities.	1	/empire/ server/ core/ go.py:79

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Using variable interpolation `\${{...}}` with `github` context data in a `run:` step could allow an attacker to inject their own code into the runner. This would allow them to steal secrets and code. `github` context data can have arbitrary user input and should be treated as untrusted. Instead, use	11	/.github/actions/clean-merge/action.yml:19
HIGH	semgrep	Found 'subprocess' function 'run' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to execute commands. Use 'shell=False' instead.	1	/empire/server/core/obfuscation_service.py:133
HIGH	semgrep	sqlalchemy.text passes the constructed SQL statement to the database mostly unchanged. This means that the usual SQL injection protections are not applied and this function is vulnerable to SQL injection if user input can reach here. Use normal SQLAlchemy operators (such as `or_()`, `and_()`, etc.)	8	/empire/server/core/db/base.py:70
HIGH	semgrep	Found 'subprocess' function 'Popen' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to execute commands. Use 'shell=False' instead.	3	/empire/server/data/agent/agent.py:865
HIGH	semgrep	Found 'subprocess' function 'check_output' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to execute commands. Use 'shell=False' instead.	2	/empire/server/stagers/windows/cmd_exec.py:146
HIGH	semgrep	Possible path traversal through `tarfile.open(\$PATH).extractall()` if the source tar is controlled by an attacker	3	/empire/server/core/plugin_service.py:329

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Detected non-static command inside Command. Audit the input to 'exec.Command'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.	2	/empire/ server/ data/ agent/ gopire/ agent/ agent.go:99
HIGH	bandit	By default, jinja2 sets autoescape to False. Consider using autoescape=True or use the select_autoescape function to mitigate XSS vulnerabilities.	1	/empire/ server/ common/ templating.py:20
HIGH	yara	YARA rule matched: detect_crypto_miner — Cryptocurrency mining pool connection patterns	1	/empire/ server/ modules/ csharp/ management/ StratumMiner.yaml
HIGH	yara	YARA rule matched: detect_process_injection — Windows process-injection API sequence	17	/empire/ server/ data/ module_source/ credentials/Invoke-Mimikatz.ps1
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	1	/empire/ server/ data/ module_source/ python/ privesc/ linuxprivchecker.py:31
HIGH	bandit	subprocess call with shell=True identified, security issue.	6	/empire/ server/ stagers/ windows/ cmd_exec.py:146

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.	2	/ Dockerfile :102
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	5	/empire/ server/ data/misc/ Reflectiv ePick_x64_ orig.dll
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	14	/empire/ server/ modules/ python/ persisten ce/osx/ LaunchAge nt.py
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	17	/ CHANGELOG. md
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	22	/empire/ server/ modules/ powershell / management / impair_hi story_log ging.yaml
MEDIUM	semgrep	Service 'ubuntu2404' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this serv	1	/.github/ install_t ests/ docker- compose- install- tests.yaml: 17

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Service 'debian12' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this servic	1	/.github/install_tests/docker-compose-install-tests.yml: 24
MEDIUM	semgrep	Service 'kalirolling' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this ser	1	/.github/install_tests/docker-compose-install-tests.yml: 38
MEDIUM	semgrep	Service 'test' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service to	1	/.github/docker-compose.yml:5
MEDIUM	semgrep	Service 'ubuntu2204' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this serv	1	/.github/install_tests/docker-compose-install-tests.yml: 10
MEDIUM	semgrep	Service 'debian13' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this servic	1	/.github/install_tests/docker-compose-install-tests.yml: 31

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Service 'parrotrolling' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this s	1	/.github/install_tests/docker-compose-install-tests.yml:45
MEDIUM	semgrep	Service 'db' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service to p	1	/.github/docker-compose.yml:27
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret "Failed to generate password hash. Verify that OpenSSL supports %s " "and that os.urandom is available." being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive in	1	/empire/server/api/jwt_auth.py:109
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret "credential parser %s raised on task output" being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/empire/server/core/agent_communication_service.py:115
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret "credential parser %s returned %s; expected list — skipping" being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/empire/server/core/agent_communication_service.py:122
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret "credential_service raised persisting %s row for %s — " "aborting remainder of this batch" being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive informat	1	/empire/server/core/agent_communication_service.py:140

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret "credential_service.create_credential rejected %s row: %s" being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/empire/ server/ core/ agent_communicatio n_service. py:148
MEDIUM	semgrep	The application's App Transport Security (ATS) configuration does not leverage the in-built public key pinning mechanisms. The application should consider leverage ATS public key pinning to ensure that the application only communicates to serves with an allow-listed certificate (and public key). By	2	/empire/ server/ data/misc/ apptempla teResourc es/x86/ launcher. app/ Contents/ Info.plist :3
MEDIUM	semgrep	Service 'parrotrolling' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/ install_t ests/ docker- compose- install- tests.yml: 45
MEDIUM	semgrep	'ssl.wrap_socket()' is deprecated. This function creates an insecure socket without server name indication or hostname matching. Instead, create an SSL context using 'ssl.SSLContext()' and use that to wrap a socket.	1	/empire/ server/ data/ module_so urce/ python/ management / socks.py:4 52
MEDIUM	semgrep	These permissions `0o0755` are widely permissive and grant access to more people than may be necessary. A good default is `0o644` which gives read and write access to yourself and read access to everyone else.	2	/empire/ server/ data/ module_so urce/ python/ privesc/ CVE-2021-4 034.py:47

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Service 'kalirolling' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/install_tests/docker-compose-install-tests.yml:38
MEDIUM	semgrep	Service 'ubuntu2204' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/install_tests/docker-compose-install-tests.yml:10
MEDIUM	semgrep	Service 'debian12' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/install_tests/docker-compose-install-tests.yml:24
MEDIUM	semgrep	Service 'debian13' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/install_tests/docker-compose-install-tests.yml:31
MEDIUM	semgrep	Service 'ubuntu2404' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/install_tests/docker-compose-install-tests.yml:17
MEDIUM	semgrep	Service 'test' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/docker-compose.yml:5
MEDIUM	semgrep	Service 'db' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/.github/docker-compose.yml:27

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Avoid using sudo in Dockerfiles. Running processes as a non-root user can help reduce the potential impact of configuration errors and security vulnerabilities.	1	/.github/install_tests/InstallTest.Dockerfile:18
MEDIUM	semgrep	This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a `cooldown` block with `default-days: 7` to each `package-ecosystem` entry under `updates` to wait 7 days before proposing updates to newly published package versions. Reference:	1	/.github/dependabot.yml:3
MEDIUM	bandit	Use of unsafe yaml load. Allows instantiation of arbitrary objects. Consider yaml.safe_load().	3	/empire/server/utils/listener_template_util.py:64
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (GCONV_PATH=./exploit).	1	/empire/server/data/module_source/python/privesc/CVE-2021-4034.py:65
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (payload.so).	1	/empire/server/data/module_source/python/privesc/CVE-2021-4034.py:47

Showing the 60 most significant of 115 distinct findings. The counts in the previous section cover all of them.

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.