

pupy

github.com/n1nj4sec/pupy

Scanned 03 October 2026 at 16:34 · Report generated 29 October 2026 · Python · 8,999 stars

OVERALL ASSESSMENT

Critical

Combined score 1269 · 521 findings after filtering

Summary

This repository has been flagged as extremely dangerous because it contains multiple critical vulnerabilities that could allow attackers to steal data, execute code, or destroy files on the server. The scan identified specific issues such as using shell commands with 'shell=True' which lets attackers run arbitrary system commands, and a deprecated library called pyCrypto that is no longer secure. These findings indicate that the code likely contains malicious intent rather than just poor security practices, as it uses suspicious techniques like downloading and executing remote files to hide its true purpose. While some issues are related to outdated libraries, the presence of these high-confidence threats means the project should not be used in any production environment without a complete rewrite by a security expert.

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 (1043)

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

MALICIOUS PATTERN

Critical (226)

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
bandit	Insecure coding patterns in Python source	67	99	141	307

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
semgrep	Insecure coding patterns across many languages	18	94	1	113
dep_checker	Vulnerable, unpinned or typosquatted dependencies	1	42	1	44
yara	Signatures of malicious behaviour in source files	18	0	10	28
guarddog	Malicious package and install-time behaviour	6	16	5	27
clamav	Known malware signatures	2	0	0	2

247 of 521 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	The application builds a file path from potentially untrusted data, which can lead to a path traversal vulnerability. An attacker can manipulate the path which the application uses to access files. If the application does not validate user input and sanitize file paths, sensitive files such as confi	1	/client/sources-linux-py3/main_so.c:122
HIGH	semgrep	Found 'subprocess' function 'call' 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	/pupy/packages/linux/all/linux_stealth.py:21
HIGH	semgrep	Found 'subprocess' function 'check_call' 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	/pupy/cli/pupygen.py:1040
HIGH	guarddog	Detects suspicious autostart persistence mechanisms [threat-filesystem-autostart]	2	/pupy/network/lib/proxies.py:237

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	clamav	ClamAV detected: Win.Trojan.Powershell-7007230-1	1	/pupy/ modules/lib/ windows/ powershell. py
HIGH	clamav	ClamAV detected: Win.Trojan.PSempireInj-7013548-0	1	/pupy/ external/ PowerSploit/ CodeExecuti on/Invoke- ReflectiveP EInjection. ps1
HIGH	bandit	The pyCrypto library and its module AES are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	4	/pupy/ network/lib/ transports/ ecm.py:20
HIGH	bandit	The pyCrypto library and its module ARC4 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	5	/pupy/ external/ creddump7/ framework/ win32/ domcachedump .py:31
HIGH	bandit	The pyCrypto library and its module HMAC are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	3	/pupy/ network/lib/ transports/ cryptoutils/ __init__.py: 30
HIGH	bandit	The pyCrypto library and its module Cipher are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/ packages/ linux/all/ secretstora ge/ util.py:46
HIGH	bandit	The pyCrypto library and its module Crypto.Cipher are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/conf/ imports_done .py:23
HIGH	bandit	The pyCrypto library and its module Counter are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/conf/ imports_done .py:8

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	bandit	The pyCrypto library and its module pycryptodome_filename are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/library_patches_py2/Crypto/Util/_raw_api.py:41
HIGH	bandit	The pyCrypto library and its module get_random_bytes are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/network/lib/transports/cryptoutils/__init__.py:28
HIGH	bandit	The pyCrypto library and its module byte_string are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/library_patches_py2/Crypto/Util/_raw_api.py:40
HIGH	bandit	The pyCrypto library and its module SHA256 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/conf/imports_done.py:5
HIGH	bandit	The pyCrypto library and its module PBKDF2 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/conf/imports_done.py:7
HIGH	bandit	The pyCrypto library and its module Crypto.Cipher.AES are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/conf/imports_done.py:24
HIGH	bandit	The pyCrypto library and its module Crypto.Hash.HMAC are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/conf/imports_done.py:25
HIGH	bandit	The pyCrypto library and its module Crypto.Hash.SHA256 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/conf/imports_done.py:26

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	bandit	The pyCrypto library and its module Crypto.Util.Counter are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/conf/imports_done.py:27
HIGH	bandit	The pyCrypto library and its module SHA384 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/network/lib/ transports/ ecm.py:22
HIGH	bandit	The pyCrypto library and its module SHA3_224 are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/network/lib/ transports/ ecm.py:21
HIGH	bandit	Use of insecure cipher Crypto.Cipher.ARC4.new. Replace with a known secure cipher such as AES.	8	/pupy/external/creddump7/framework/win32/domcachedump.py:45
HIGH	bandit	Use of insecure cipher Crypto.Cipher.DES.new. Replace with a known secure cipher such as AES.	5	/pupy/external/creddump7/framework/win32/hashdump.py:179
HIGH	bandit	The pyCrypto library and its module DES are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	1	/pupy/packages/all/pupyutils/psexec.py:49
HIGH	bandit	The pyCrypto library and its module Random are no longer actively maintained and have been deprecated. Consider using pyca/cryptography library.	2	/pupy/pupylib/PupyCredentials.py:37
HIGH	bandit	Paramiko call with policy set to automatically trust the unknown host key.	1	/pupy/packages/all/ssh.py:864

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	/pupy/packages/linux/all/linux_stealth.py:47
HIGH	guarddog	Detects dynamic import chains used to obfuscate code execution [threat-runtime-obfuscation-import-exec]	1	/pupy/agent/__init__.py:346
HIGH	dep_checker	Typosquatting: 'scapy' is suspiciously similar to 'scrapy'	1	requirements.txt:15
HIGH	bandit	tarfile.extractall used without any validation. Please check and discard dangerous members.	1	/pupy/cli/pupygen.py:315
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	/pupy/packages/posix/all/sudo_alias.py:152
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.	3	/pupy/packages/windows/all/pupwinutils/bypassuac_registry.py:76
HIGH	semgrep	Avoiding SQL string concatenation: untrusted input concatenated with raw SQL query can result in SQL Injection. In order to execute raw query safely, prepared statement should be used. SQLAlchemy provides TextualSQL to easily used prepared statement with named parameters. For complex SQL composition	5	/pupy/pupylib/PupyServer.py:659
HIGH	semgrep	Insecure WebSocket Detected. WebSocket Secure (wss) should be used for all WebSocket connections.	1	/pupy/webstatic/rdesktop/rdesktop.js:11

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Possible path traversal through <code>`tarfile.open(\$PATH).extractall()`</code> if the source tar is controlled by an attacker	1	/pupy/cli/pupygen.py:313
HIGH	semgrep	Private Key detected. This is a sensitive credential and should not be hardcoded here. Instead, store this in a separate, private file.	1	/pupy/network/ transports/ __init__.py:71
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	9	/pupy/network/lib/ launchers/ dnscnc.py:86
HIGH	yara	YARA rule matched: detect_process_injection — Windows process-injection API sequence	4	/pupy/external/PowerSploit/CodeExecution/Invoke-ReflectivePEInjection.ps1
HIGH	bandit	subprocess call with shell=True identified, security issue.	8	/pupy/cli/pupygen.py:1040
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'.	1	/client/pyoxidizer-build/linux-builder.Dockerfile:16
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	/pupy/packages/windows/amd64/vidcap.pyd
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	1	/pupy/network/lib/picocmd/picocmd.py

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	/pupy/ modules/ exploit_sug gester.py
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	2	/pupy/ webstatic/ jquery-3.2.1 .min.js
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	4	/pupy/ network/lib/ proxies.py
MEDIUM	semgrep	Found user-controlled request data passed into '.write(...)'. This could be dangerous if a malicious actor is able to control data into sensitive files. For example, a malicious actor could force rolling of critical log files, or cause a denial-of-service by using up available disk space. Instead, e	1	/pupy/ packages/ windows/all/ powershell. py:231
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'SOCKS Proxy found via GIO: %s:%s user=%s password=%s' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/ network/lib/ proxies.py:3 45
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Force preconfigured user/ password for %s (realm=%s) -> user=%s' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/ network/lib/ tinyhttp.py: 93

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'HTTP Proxy found via GIO: %s:%s user=%s password=%s' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/network/lib/proxies.py:36
MEDIUM	semgrep	Avoid using `cPickle`, which is known to lead to code execution vulnerabilities. When unpickling, the serialized data could be manipulated to run arbitrary code. Instead, consider serializing the relevant data as JSON or a similar text-based serialization format.	5	/pupy/agent/utils.py:49
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret '(Realm: %s) Use preconfigured credentials: domain=%s user=%s' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/packages/all/ad.py:1210
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret "Using default credentials for %s" being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/pupylib/Pupycredentials.py:716
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Credential "%s" will expire in %d days' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/pupylib/Pupycredentials.py:593
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Credentials storage (%s) is empty' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/pupylib/Pupycredentials.py:661
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'secretstorage dbus initialization failed: %s' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/pupylib/Pupycredentials.py:77

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Credential "%s" is expired! ' 'All related credentials will be regenerated' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/ pupylib/ PupyCredentials.py:581
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Credential "%s" is missing and will be generated' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/ pupylib/ PupyCredentials.py:571
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Credential "%s" exists' being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/pupy/ pupylib/ PupyCredentials.py:597

Showing the 60 most significant of 174 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 03 October 2026.