

byob

github.com/malwaredlc/byob

Scanned 18 September 2026 at 19:36 · Report generated 29 October 2026 · Python · 9,504 stars

OVERALL ASSESSMENT

Critical

Combined score 517 · 369 findings after filtering

Summary

This repository shows clear signs of malicious intent rather than just poor coding practices. The most serious issue is that the code contains active cryptocurrency mining tools and reverse shells, which are designed to steal money or control other computers, not just cause temporary errors. While there are some standard security warnings about using insecure file permissions and FTP protocols, these are common mistakes in many projects and do not indicate that the entire system is unsafe. The presence of known malware evasion techniques like checking for debuggers before running code further confirms that this is a botnet or mining farm rather than a legitimate application. You should not use this repository as it poses a high risk of financial loss and data theft if accessed by an attacker.

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 (448)

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

MALICIOUS PATTERN

High (70)

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	72	104	79	255

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
yara	Signatures of malicious behaviour in source files	88	0	25	113
clamav	Known malware signatures	1	0	0	1

28 of 369 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.Coinminer.Generic-7151250-0	1	/web-gui/ buildyourow nbotnet/ modules/ xmrig/ xmrig_win32
HIGH	bandit	FTP-related functions are being called. FTP is considered insecure. Use SSH/SFTP/SCP or some other encrypted protocol.	5	/web-gui/ buildyourow nbotnet/ modules/ util.py:544
HIGH	bandit	A FTP-related module is being imported. FTP is considered insecure. Use SSH/SFTP/SCP or some other encrypted protocol.	8	/byob/ modules/ util.py:529
HIGH	yara	YARA rule matched: detect_reverse_shell — Reverse shell command patterns	1	/web-gui/ docker- pyinstaller/ entrypoint- linux.sh
HIGH	yara	YARA rule matched: detect_crypto_miner — Cryptocurrency mining pool connection patterns	12	/.git/index
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	24	/web-gui/ buildyourow nbotnet/ modules/ process.py:1 28
HIGH	bandit	subprocess call with shell=True identified, security issue.	35	/byob/ modules/ persistence. py:115

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	/byob/core/stagers.py
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	9	/byob/client.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	22	/web-gui/buildyourownbotnet/core/stagers.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	31	/web-gui/buildyourownbotnet/assets/js/codemirror/keymap/vim.js
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	10	/web-gui/docker-pyinstaller/Dockerfile-py2-win32
MEDIUM	bandit	Deserialization with the marshal module is possibly dangerous.	6	/web-gui/buildyourownbotnet/core/loader.py:166
MEDIUM	bandit	Chmod setting a permissive mask 0o1411 on file (target).	5	/web-gui/buildyourownbotnet/modules/util.py:362
MEDIUM	bandit	Chmod setting a permissive mask 0o1363 on file (mas).	2	/web-gui/buildyourownbotnet/modules/icloud.py:41

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Use of weak MD5 hash for security. Consider usedforsecurity=False	4	/web-gui/buildyourownbotnet/core/dao.py:110
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	20	/web-gui/buildyourownbotnet/core/dummy_payload_for_testing.py:93
MEDIUM	bandit	Possible SQL injection vector through string-based query construction.	1	/byob/core/database.py:242
MEDIUM	bandit	Use of possibly insecure function - consider using safer ast.literal_eval.	5	/byob/core/payloads.py:330
MEDIUM	bandit	Use of exec detected.	16	/byob/core/util.py:450
MEDIUM	bandit	Possible binding to all interfaces.	8	/web-gui/buildyourownbotnet/server.py:123
MEDIUM	bandit	Probable insecure usage of temp file/directory.	31	/web-gui/buildyourownbotnet/modules/util.py:549
MEDIUM	bandit	Call to requests without timeout	6	/web-gui/buildyourownbotnet/modules/util.py:221
LOW	bandit	Possible hardcoded password: 'helloworld'	1	/web-gui/buildyourownbotnet/config.py:10
LOW	bandit	Starting a process without a shell.	4	/byob/core/payloads.py:599

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Starting a process with a shell: Seems safe, but may be changed in the future, consider rewriting without shell	5	/web-gui/buildyourownbotnet/core/dummy_payload_for_testing.py:684
LOW	bandit	Consider possible security implications associated with the subprocess module.	15	/web-gui/buildyourownbotnet/api/payload/routes.py:2
LOW	bandit	Starting a process with a partial executable path	8	/web-gui/buildyourownbotnet/api/payload/routes.py:37
LOW	bandit	subprocess call - check for execution of untrusted input.	15	/web-gui/buildyourownbotnet/server.py:140
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	25	/web-gui/buildyourownbotnet/modules/util.py
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	31	/web-gui/buildyourownbotnet/client.py:377

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.