

CrackMapExec

github.com/byt3bl33d3r/CrackMapExec

Scanned 21 September 2026 at 16:29 · Report generated 29 October 2026 · Python · 9,161 stars

OVERALL ASSESSMENT

Critical

Combined score 510 · 240 findings after filtering

Summary

This repository has been flagged as having critical vulnerabilities that pose a severe threat to system safety, primarily because it uses dangerous coding patterns like executing commands through a shell and disabling SSL certificate checks. These issues are not necessarily signs of malicious intent but rather represent significant security risks where attackers could exploit the code to steal data or cause denial of service. While some findings point to intentional malicious behavior, the majority appear to be standard insecure practices that developers should fix immediately to protect against real-world attacks. The scan also identified several unsecured protocols such as FTP and weak hash algorithms, which further increase the potential for exploitation if these vulnerabilities are not addressed.

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 (506)

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

MALICIOUS PATTERN

Low (4)

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	12	61	67	140

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
semgrep	Insecure coding patterns across many languages	40	28	2	70
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	2	26	28
yara	Signatures of malicious behaviour in source files	1	0	1	2

99 of 240 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	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	/cme/helpers/powershell.py:48
HIGH	semgrep	certificate verification explicitly disabled, insecure connections possible	1	/cme/modules/ldap-checker.py:93
HIGH	bandit	Use of weak MD4 hash for security. Consider usedforsecurity=False	2	/cme/protocols/smb.py:356
HIGH	bandit	Call to requests with verify=False disabling SSL certificate checks, security issue.	6	/cme/modules/empire_exec.py:123
HIGH	bandit	Paramiko call with policy set to automatically trust the unknown host key.	1	/cme/protocols/ssh.py:53
HIGH	bandit	FTP-related functions are being called. FTP is considered insecure. Use SSH/SFTP/SCP or some other encrypted protocol.	1	/cme/protocols/ftp.py:46

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	The native Python `xml` library is vulnerable to XML External Entity (XXE) attacks. These attacks can leak confidential data and "XML bombs" can cause denial of service. Do not use this library to parse untrusted input. Instead the Python documentation recommends using `defusedxml`.	1	/cme/modules/keepass_trigger.py:392
HIGH	bandit	A FTP-related module is being imported. FTP is considered insecure. Use SSH/SFTP/SCP or some other encrypted protocol.	1	/cme/protocols/ftp.py:6
HIGH	semgrep	Certificate verification has been explicitly disabled. This permits insecure connections to insecure servers. Re-enable certification validation.	6	/cme/cmedb.py:440
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	29	/cme/modules/handlekatz.py:85
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	/cme/data/default.pem:1
HIGH	bandit	subprocess call with shell=True identified, security issue.	1	/cme/helpers/powershell.py:48
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	/Dockerfile:22
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	1	/Dockerfile

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected a python logger call with a potential hardcoded secret 'Cannot unpack msLAPS-EncryptedPassword blob due to error %s' % str(e) being logged. This may lead to secret credentials being exposed. Make sure that the logger is not logging sensitive information.	1	/cme/ protocols/ ldap/ laps.py:216
MEDIUM	semgrep	Detected use of an insecure MD4 or MD5 hash function. These functions have known vulnerabilities and are considered deprecated. Consider using 'SHA256' or a similar function instead.	2	/cme/ protocols/ winrm.py:179
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.	2	/cme/ servers/ http.py:74
MEDIUM	semgrep	Detected DES cipher or Triple DES algorithm which is considered insecure. This algorithm is not cryptographically secure and can be reversed easily. Use a secure symmetric cipher from the cryptodome package instead. Use secure stream ciphers such as ChaCha20, XChaCha20 and Salsa20, or a block cipher	2	/cme/ protocols/ smb/ firefox.py:224
MEDIUM	semgrep	Detected MD4 hash algorithm which is considered insecure. MD4 is not collision resistant and is therefore not suitable as a cryptographic signature. Use a modern hash algorithm from the SHA-2, SHA-3, or BLAKE2 family instead.	3	/cme/ protocols/ ldap.py:1372
MEDIUM	bandit	Use of insecure MD2, MD4, MD5, or SHA1 hash function.	3	/cme/ protocols/ smb.py:1712
MEDIUM	bandit	Possible shell injection via Paramiko call, check inputs are properly sanitized.	5	/cme/ protocols/ ssh.py:46

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Using xml.etree.ElementTree.parse to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.parse with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	1	/cme/modules/keepass_trigger.py:392
MEDIUM	dep_checker	Unpinned dependency: 'oscrypto' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	dep_checker	Unpinned dependency: 'impacket' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	semgrep	Detected SHA1 hash algorithm which is considered insecure. SHA1 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	3	/cme/protocols/smb/firefox.py:236
MEDIUM	semgrep	Detected possible formatted SQL query. Use parameterized queries instead.	7	/cme/modules/hash_spider.py:245
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/crackmapexec-test.yml:20
MEDIUM	bandit	Using xml.etree.ElementTree.fromstring to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.fromstring with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	3	/cme/modules/gpp_password.py:50
MEDIUM	bandit	Use of weak SHA1 hash for security. Consider usedforsecurity=False	3	/cme/protocols/smb/firefox.py:215
MEDIUM	bandit	Function call with shell=True parameter identified, possible security issue.	2	/cme/protocols/ssh.py:154

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Possible SQL injection vector through string-based query construction.	14	/cme/protocols/wmi/wmiexec_event.py:164
MEDIUM	semgrep	Found non static data as an index to 'globals()'. This is extremely dangerous because it allows an attacker to execute arbitrary code on the system. Refactor your code not to use 'globals()'.	2	/cme/protocols/smb/database.py:960
MEDIUM	bandit	Possible binding to all interfaces.	2	/cme/servers/smb.py:16
MEDIUM	bandit	Probable insecure usage of temp file/directory.	23	/cme/modules/nanodump.py:68
MEDIUM	bandit	Call to requests without timeout	5	/cme/modules/empire_exec.py:96
LOW	semgrep	The 'FTP' class sends information unencrypted. Consider using the 'FTP_TLS' class instead.	1	/cme/protocols/ftp.py:46
LOW	bandit	ssl.wrap_socket call with no SSL/TLS protocol version specified, the default SSLv23 could be insecure, possible security issue.	2	/cme/servers/http.py:74
LOW	bandit	Possible hardcoded password: 'neo4j'	1	/cme/modules/bh_owned.py:40
LOW	bandit	Possible hardcoded password: 'C:\'	2	/cme/protocols/smb/wmiexec.py:46
LOW	bandit	Using ElementTree to parse untrusted XML data is known to be vulnerable to XML attacks. Replace ElementTree with the equivalent defusedxml package, or make sure defusedxml.defuse_stdlib() is called.	1	/cme/modules/keepass_trigger.py:9

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'pass'	1	/cme/ protocols/ rdp.py:116
LOW	bandit	Using xml.etree.ElementTree to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree with the equivalent defusedxml package, or make sure defusedxml.defuse_stdlib() is called.	2	/cme/ modules/ gpp_autologin.py:4
LOW	bandit	Possible hardcoded password: ''	49	/cme/ protocols/ ldap.py:316
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.	1	/cme/ protocols/ winrm.py:208
LOW	bandit	Consider possible security implications associated with the subprocess module.	2	/cme/ helpers/ powershell.py:8
LOW	dep_checker	Loose version constraint: 'python-libnmap^0.7.3' — attacker could push malicious update	1	pyproject. toml
LOW	bandit	subprocess call - check for execution of untrusted input.	1	/ build_collector.py:37
LOW	dep_checker	Loose version constraint: 'neo4j^4.1.1' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'xmldict^0.12.0' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'terminaltables^3.1.0' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'sqlalchemy^2.0.4' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'aardwolf^0.2.7' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'aioconsole^0.3.3' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'aiosqlite^0.18.0' — attacker could push malicious update	1	pyproject. toml

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'beautifulsoup4>=4.11,<5' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'bloodhound^1.6.1' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'dploot^2.2.1' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'dsinternals^1.2.4' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'lsassy>=3.1.8' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'masky^0.2.0' — attacker could push malicious update	1	pyproject.toml

Showing the 60 most significant of 74 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 21 September 2026.