

client

github.com/keybase/client

Scanned 24 September 2026 at 06:24 · Report generated 29 October 2026 · Go · 9,247 stars

OVERALL ASSESSMENT

Critical

Combined score 812 · 455 findings after filtering

Summary

This repository has two distinct security issues that require immediate attention. The most critical finding is a vulnerability score of 789, which indicates serious insecure coding patterns that an attacker could exploit to inject malicious code into the system. Additionally, there are several medium-risk findings related to potential malicious intent, such as hardcoded PGP keys and suspicious command execution patterns, which suggest the code might be intentionally harmful or obfuscated. While some of these issues appear to be standard security practices like avoiding unsafe file operations, they still represent significant risks that should not be ignored without careful review.

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 (789)

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

MALICIOUS PATTERN

Medium (24)

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	52	336	3	391
yara	Signatures of malicious behaviour in source files	18	0	9	27

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	5	3	14	22
guarddog	Malicious package and install-time behaviour	0	4	11	15

101 of 455 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	Detected non-static command inside Exec. Audit the input to 'syscall.Exec'. 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.	1	/go/libcmdline/fork_nix.go:41
HIGH	semgrep	Detected calls to child_process from a function argument `action`. This could lead to a command injection if the input is user controllable. Try to avoid calls to child_process, and if it is needed ensure user input is correctly sanitized or sandboxed.	1	/shared/desktop/app/ipc-handlers.desktop.tsx:191
HIGH	semgrep	Detected calls to child_process from a function argument `args`. This could lead to a command injection if the input is user controllable. Try to avoid calls to child_process, and if it is needed ensure user input is correctly sanitized or sandboxed.	1	/shared/desktop/app/exec.desktop.tsx:49
HIGH	semgrep	Something that looks like a PGP private key block is detected. This is a potential hardcoded secret that could be leaked if this code is committed. Instead, remove this code block from the commit.	4	/go/fuzz/pgp_key_read/corpus/1:53
HIGH	semgrep	Missing mutex unlock (`api.clientsMu` variable) before returning from a function. This could result in panics resulting from double lock operations	1	/go/libkb/api.go:143

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Missing mutex unlock (`lock.lock` variable) before returning from a function. This could result in panics resulting from double lock operations	1	/go/chat/utls/locktab.go:124
HIGH	semgrep	Facebook OAuth detected	2	/go/chat/unfurl/testcases/pinterest0.html:463
HIGH	semgrep	`Clean` is not intended to sanitize against path traversal attacks. This function is for finding the shortest path name equivalent to the given input. Using `Clean` to sanitize file reads may expose this application to path traversal attacks, where an attacker could access arbitrary files on the ser	1	/go/kbfs/libpages/server.go:372
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	/go/tools/airdrop-setup/airdrop-setup.py:71
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.	1	/go/tools/airdrop-setup/airdrop-setup.py:46
HIGH	semgrep	bcrypt hash detected	1	/go/kbfs/libpages/config/passwords.go:153
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.	26	/go/pinentry/pinentry.go:132

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	/go/kbfs/kbfs/crypto/root_certs.go:99
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	2	/go/tools/airdrop-setup/airdrop-setup.py:55
HIGH	bandit	subprocess call with shell=True identified, security issue.	3	/go/tools/airdrop-setup/airdrop-setup.py:71
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'.	9	/packaging/linux/docker/standard/Dockerfile:39
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	4	/shared/scripts/react-compiler-bailouts.mts
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	7	/packaging/linux/docker/Dockerfile
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	1	/shared/common-adapters/icon.tsx
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	6	/packaging/linux/rpm/postinst.template

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Variable `result` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	7	/go/updater/ keybase/ platform_linux.go:98
MEDIUM	semgrep	Variable `rres` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	2	/go/stellar/ stellar.go:867
MEDIUM	semgrep	Variable `maxDelH` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	2	/go/ protocol/ chat1/ extras.go:1660
MEDIUM	semgrep	Variable `obr` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/chat/ sender.go:1422
MEDIUM	semgrep	Variable `candidate` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/teams/ service_helper.go:473
MEDIUM	semgrep	Variable `spotlightResult` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/updater/ keybase/ platform_darwin.go:360
MEDIUM	semgrep	Variable `cmd` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/keybase/ main.go:196
MEDIUM	semgrep	Variable `md` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/kbfs/ libkbfs/ disk_block_cache.go:830
MEDIUM	semgrep	Variable `ret` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/libkb/ upak_loader.go:661
MEDIUM	semgrep	Variable `quarantineResult` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/updater/ keybase/ platform_darwin.go:367
MEDIUM	semgrep	Variable `res` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/teams/ service_helper.go:952

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	A request was found to be crafted from user-input `req`. This can lead to Server-Side Request Forgery (SSRF) vulnerabilities, potentially exposing sensitive data. It is recommend where possible to not allow user-input to craft the base request, but to be treated as part of the path or query paramete	1	/go/chat/attachment_httpsrv.go:393
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	17	/shared/ios/KeybaseShare/Info.plist:3
MEDIUM	semgrep	Cipher in ECB mode is detected. ECB mode produces the same output for the same input each time which allows an attacker to intercept and replay the data. Further, ECB mode does not provide any integrity checking. See https://find-sec-bugs.github.io/bugs.htm#CIPHER_INTEGRITY .	2	/shared/android/app/src/main/java/io/keybase/ossifrage/KeyStore.kt:193
MEDIUM	semgrep	Detected 'io.WriteString()' writing directly to 'http.ResponseWriter'. This bypasses HTML escaping that prevents cross-site scripting vulnerabilities. Instead, use the 'html/template' package to render data to users.	1	/go/chat/attachment_httpsrv.go:362
MEDIUM	semgrep	Variable `inbox` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/go/chat/helper.go:596
MEDIUM	semgrep	Detected an element with disabled HTML escaping. If external data can reach this, this is a cross-site scripting (XSS) vulnerability. Ensure no external data can reach here, or remove 'escape=false' from this element.	4	/go/chat/unfurl/testcases/pinterest0.html:463
MEDIUM	semgrep	Iteration over a possibly empty map `parentsToAddChainsFor`. This is likely a bug or redundant code	1	/go/kbfs/libkbfs/folder_branch_ops.go:6336

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected user input flowing into a manually constructed HTML string. You may be accidentally bypassing secure methods of rendering HTML by manually constructing HTML and this could create a cross-site scripting vulnerability, which could let attackers steal sensitive user data. Use the `html/templat	1	/go/chat/attachment_httpsrv.go:339
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/go/kbtime/long_duration.go:12
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	3	/go/libkb/env.go:733
MEDIUM	semgrep	The application's App Transport Security (ATS) configuration allows for local networking, which in some iOS versions is blocked by default in ATS. Consider whether or not the application requires this in release builds, or for development reasons only. This follows the principle of least privilege,	1	/shared/ios/Keybase/Info.plist:52
MEDIUM	semgrep	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks.	6	/go/chat/unfurl/testcases/twitter0.html:319
MEDIUM	semgrep	RegExp() called with a `filter` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controll	1	/shared/util/string.tsx:15
MEDIUM	semgrep	The application exports an activity. Any application on the device can launch the exported activity which may compromise the integrity of your application or its data. Ensure that any exported activities do not have privileged access to your application's control plane.	1	/shared/android/app/src/main/AndroidManifest.xml:27

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	/shared/tools/fonts/font_tool.py:157
MEDIUM	semgrep	Detected 'Fprintf' or similar writing to 'http.ResponseWriter'. This bypasses HTML escaping that prevents cross-site scripting vulnerabilities. Instead, use the 'html/template' package to render data to users.	2	/go/chat/attachment_httpsrv.go:483
MEDIUM	semgrep	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	5	/go/chat/attachments/s3.go:265
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.	1	/go/chat/storage/inbox.go:427
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	1	/.github/workflows/issues.yml:13
MEDIUM	semgrep	Checks for any usage of http servers instead of https servers. Encourages the usage of https protocol instead of http, which does not have TLS and is therefore unencrypted. Using http can lead to man-in-the-middle attacks in which the attacker is able to read sensitive information.	2	/shared/perf/run-desktop-cdp-profile.js:33

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	When working with web applications that involve rendering user-generated content, it's important to properly escape any HTML content to prevent Cross-Site Scripting (XSS) attacks. In Go, the <code>`text/template`</code> package does not automatically escape HTML content, which can leave your application vuln	1	/go/chat/emoji/main.go:14
MEDIUM	semgrep	The special variable IFS affects how splitting takes place when expanding unquoted variables. Don't set it globally. Prefer a dedicated utility such as 'cut' or 'awk' if you need to split input data. If you must use 'read', set IFS locally using e.g. 'IFS="," read -a my_array'.	1	/packaging/linux/docker/push.sh:59
MEDIUM	semgrep	Detected a possible denial-of-service via a zip bomb attack. By limiting the max bytes read, you can mitigate this attack. <code>`io.CopyN()`</code> can specify a size.	4	/go/updater/util/unzip.go:173
MEDIUM	semgrep	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.	31	/go/chat/unfurl/testcases/wikipedia0.html:163
MEDIUM	semgrep	Do not use <code>`math/rand`</code> . Use <code>`crypto/rand`</code> instead.	3	/go/libkb/backoff.go:7
MEDIUM	semgrep	Detected possible user input going into a <code>`path.join`</code> or <code>`path.resolve`</code> function. This could possibly lead to a path traversal vulnerability, where the attacker can access arbitrary files stored in the file system. Instead, be sure to sanitize or validate user input first.	11	/shared/perf/compare-perf.js:20
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	218	/go/chat/unfurl/testcases/giphy0.html:82

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Probable insecure usage of temp file/directory.	1	/go/tools/airdrop-setup/airdrop-setup.py:60
MEDIUM	bandit	Call to requests without timeout	1	/go/tools/airdrop-setup/airdrop-setup.py:73

Showing the 60 most significant of 75 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 24 September 2026.