

psiphon-tunnel-core

github.com/Psiphon-Labs/psiphon-tunnel-core

Scanned 24 September 2026 at 07:32 · Report generated 29 October 2026 · Go · 1,158 stars

OVERALL ASSESSMENT

High

Combined score 278 · 132 findings after filtering

Summary

The repository has two distinct security issues that need immediate attention before it can be used in production. The most critical problem is that the code contains hardcoded private keys, which are extremely dangerous because anyone with access to this file can steal sensitive credentials and compromise the entire system. Additionally, there are several high-risk vulnerabilities related to unsafe coding practices, such as missing mutex locks that could cause crashes and untrusted input that might lead to cross-site scripting attacks. While some of these issues appear to be intentional obfuscation or malware evasion techniques rather than simple mistakes, they still represent significant security risks that should not be ignored. The developer should remove the hardcoded keys immediately and fix the unsafe patterns before deploying this code anywhere.

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

High (258)

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

MALICIOUS PATTERN

Medium (19)

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	15	87	0	102

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
yara	Signatures of malicious behaviour in source files	7	0	15	22
guarddog	Malicious package and install-time behaviour	0	5	3	8

76 of 132 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	Missing mutex unlock (<code>`manager.ctxMutex`</code> variable) before returning from a function. This could result in panics resulting from double lock operations	1	/psiphon/common/refraction/refraction.go:919
HIGH	semgrep	Untrusted input could be used to tamper with a web page rendering, which can lead to a Cross-site scripting (XSS) vulnerability. XSS vulnerabilities occur when untrusted input executes malicious JavaScript code, leading to issues such as account compromise and sensitive information leakage. To preve	3	/psiphon/internal/testutils/dsl.go:343
HIGH	semgrep	Calling <code>`waitGroup.Add`</code> inside of an anonymous goroutine may result in <code>`waitGroup.Wait`</code> waiting for more or less calls to <code>`waitGroup.Done()`</code> than expected	1	/psiphon/common/light/proxy.go:753
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.	9	/MobileLibrary/go-mobile/cmd/gomobile/init.go:182

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	/ MobileLibrary/go-mobile/cmd/gomobile/build_androidapp.go:330
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	1	/vendor/github.com/syndtr/gocapability/capability/capability_linux.go
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	2	/vendor/github.com/bits-and-blooms/bitset/azure-pipelines.yml
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	/vendor/golang.org/x/sys/unix/syscall_zos_s390x.go
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	2	/vendor/github.com/ryanuber/go-glob/README.md
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	8	/ MobileLibrary/iOS/PsiphonTunnel/PsiphonTunnel/Info.plist:3

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected DES cipher algorithm which is insecure. The algorithm is considered weak and has been deprecated. Use AES instead.	2	/psiphon/upstreamproxy/go-ntlm/ntlm/crypto.go:76
MEDIUM	semgrep	This code uses a 224-bit hash function, which is deprecated or disallowed in some security policies. Consider updating to a stronger hash function such as SHA-384 or higher to ensure compliance and security.	1	/replace/dtls/pkg/crypto/hash/hash.go:67
MEDIUM	semgrep	Detected RC4 cipher algorithm which is insecure. The algorithm has many known vulnerabilities. Use AES instead.	5	/psiphon/common/obfuscator/obfuscator.go:322
MEDIUM	semgrep	Iteration over a possibly empty map `set`. This is likely a bug or redundant code	1	/MobileLibrary/go-mobile/internal/importers/java/java.go:655
MEDIUM	semgrep	Detected use of the functions `Math.random()` or `java.util.Random()`. These are both not cryptographically strong random number generators (RNGs). If you are using these RNGs to create passwords or secret tokens, use `java.security.SecureRandom` instead.	1	/MobileLibrary/go-mobile/bind/java/SeqTest.java:171
MEDIUM	semgrep	The application's App Transport Security (ATS) configuration allows for insecure HTTP connections. The `NSAllowsArbitraryLoads` and `NSTemporaryExceptionAllowsInsecureHTTPLoads` property keys can be used to allow HTTP for all domains or a particular domain respectively. HTTPs should be used in all c	2	/MobileLibrary/iOS/SampleApps/TunneledWebRequest/TunneledWebRequest/Info.plist:27

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/ MobileLibrary/go-mobile/internal/binres/arsc.go:9
MEDIUM	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	1	/vendor/github.com/refraction-networking/conjure/pkg/client/assets/assets.go:112
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	3	/ MobileLibrary/go-mobile/vendor/golang.org/x/tools/internal/stdlib/manifest.go:8666
MEDIUM	semgrep	A session cookie was detected without setting the 'HttpOnly' flag. The 'HttpOnly' flag for cookies instructs the browser to forbid client-side scripts from reading the cookie which mitigates XSS attacks. Set the 'HttpOnly' flag by setting 'HttpOnly' to 'true' in the Cookie.	2	/psiphon/meekConn.go:2012

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	/ MobileLibrary/ Android/ SampleApps/ / TunneledWebView/ app/src/ main/ AndroidManifest.xml :11
MEDIUM	semgrep	Downcasting or changing sign of an integer with `uint32` method	2	/ MobileLibrary/go-mobile/ internal/ binres/ binres.go: 419
MEDIUM	semgrep	A session cookie was detected without setting the 'Secure' flag. The 'secure' flag for cookies prevents the client from transmitting the cookie over insecure channels such as HTTP. Set the 'Secure' flag by setting 'Secure' to 'true' in the Options struct.	2	/psiphon/ meekConn. go:2012
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.	8	/ MobileLibrary/go-mobile/ cmd/ gomobile/ cert.go:44
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.	8	/psiphon/ upstreamproxy/go-ntlm/ntlm/ crypto.go: 22
MEDIUM	semgrep	`MinVersion` is missing from this TLS configuration. By default, as of Go 1.22, TLS 1.2 is currently used as the minimum. General purpose web applications should default to TLS 1.3 with all other protocols disabled. Only where it is known that a web server must support legacy clients with unsupport	16	/psiphon/ common/ quic/ gquic.go:5 8

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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. <code>`uses: ac</code>	2	<code>/.github/workflows/tests.yml:40</code>
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	2	<code>/MobileLibrary/go-mobile/cmd/gomobile/build_apple.go:17</code>
MEDIUM	semgrep	Deserializing into <code>`interface{}`</code> allows arbitrary data structures and types, which can lead to security vulnerabilities (CWE-502). Use a concrete struct type instead.	3	<code>/psiphon/dataStore.go:2088</code>
MEDIUM	semgrep	Checks for disabling of TLS/SSL certificate verification. This should only be used for debugging purposes because it leads to vulnerability to MTM attacks.	5	<code>/psiphon/common/inproxy/webrtc.go:2399</code>
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.	3	<code>/MobileLibrary/go-mobile/internal/binres/table.go:116</code>
MEDIUM	semgrep	Found an HTTP server without TLS. Use 'http.ListenAndServeTLS' instead. See https://golang.org/pkg/net/http/#ListenAndServeTLS for more information.	1	<code>/psiphon/pprof.go:37</code>
MEDIUM	semgrep	Detected directly writing or similar in 'http.ResponseWriter.write()'. This bypasses HTML escaping that prevents cross-site scripting vulnerabilities. Instead, use the 'html/template' package and render data using 'template.Execute()'.	4	<code>/psiphon/server/meek.go:2275</code>

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Do not use <code>`math/rand`</code> . Use <code>`crypto/rand`</code> instead.	8	/psiphon/common/regen/regen.go:121
LOW	guarddog	Detects heavy obfuscation techniques commonly used by malware [threat-runtime-obfuscation]	1	/MobileLibrary/go-mobile/event/key/code_string.go:9
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	2	/psiphon/common/protocol/signer/main.go:59
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	15	/vendor/golang.org/x/sys/unix/ztypes_openbsd_mips64.go

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.