A Local Heuristic Warning Layer That Complements Cloud Protection

Technical Architecture, Two-Stage Verification, and TLS Reporting in RITS Browser v4.2.7

Publisher: Raise IT Solutions LLC | Date: November 16, 2026 | Revision C (v7.9 Reconciled 300 DPI Baseline)
Target Build: RITS Browser v4.2.7 (Build: 255, Commit: 255-v4.2.7-release) | Base APK SHA-256:
271a9aced1d5df4aefaf75470c716c043cb3acf77eb99f837094e77952eea9f6

DISCLAIMER: First-party internal engineering evaluation; not an independent audit, certification, or academic peer-reviewed publication.

1. Executive Summary & Main Technical Claim

RITS Browser v4.2.7 implements a local heuristic warning layer for selected lookalike-domain patterns. On the documented navigation paths, the check is placed before the explicit WebView loading call, subject to the lifecycle behavior described in the implementation. A warning was observed for the test domain. The available socket snapshot does not establish absence of DNS, QUIC, or all other pre-warning network activity. Performance figures are preliminary host-JVM measurements, not Android-device or full warning-display latency.

Commercial mobile browsers rely on centralized cloud threat intelligence feeds (such as Google Safe Browsing), complemented by varying client-side models and heuristics. While cloud feeds provide broad coverage against established malicious campaigns, centralized architectures face an ingestion latency gap during initial hours of operation of newly registered lookalike domains, before signatures propagate across client blocklists.

TWO-STAGE SECURITY PIPELINE & SCOPED NETWORK SCOPE

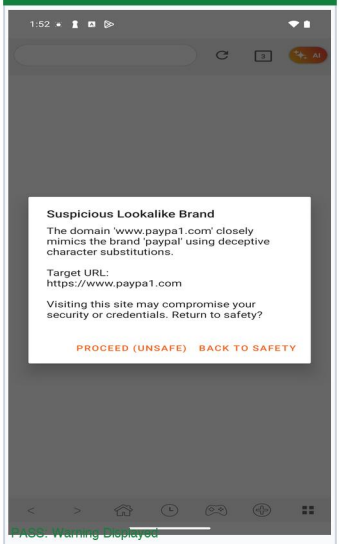
- Stage 1 (Pre-Navigation Hook): Inspects domains, Levenshtein edit distance (d = 1), IDN homoglyphs (xn-- Punycode decoding), userinfo traps, and sensitive HTTP paths in loadUrl/shouldOverrideUrlLoading.
- Stage 2 (Post-Response Download Callback): Inspects server-provided Content-Type headers and Content-Disposition filenames in onDownloadStart to halt disguised executables (.pdf.apk).
- Scoped Network Scope: The check precedes the explicit WebView load call. No relevant TCP socket was observed in recorded snapshots. Absence of other pre-warning traffic remains unverified.

2. Two-Stage Threat Taxonomy & Detection Mechanisms

Threat Category	Detection Stage & Trigger Hook	Algorithmic Rule & Scope
Brand Lookalikes (Typosquatting)	Stage 1 (loadUrl / shouldOverrideUrlLoading)	Levenshtein distance d = 1 against 100+ protected seeds + brand-stem rules (*book).
IDN Homoglyphs & Multilingual IDN	Stage 1 (loadUrl / shouldOverrideUrlLoading)	xn-- Punycode decoding; traps mixed Cyrillic/Greek & Latin script. Single-script IDNs pass.
Userinfo Credential Obfuscation	Stage 1 (loadUrl / shouldOverrideUrlLoading)	Traps userinfo structures (e.g. https://paypal.com@evil.com) before host extraction.
Insecure Sensitive Auth Paths	Stage 1 (loadUrl / shouldOverrideUrlLoading)	Flags HTTP requests containing sensitive path segments (e.g. http://example.test/login).
Disguised Executable Downloads	Stage 2 (onDownloadStart callback)	Inspects server MIME headers & Content-Disposition filenames for double extensions (.pdf.apk).

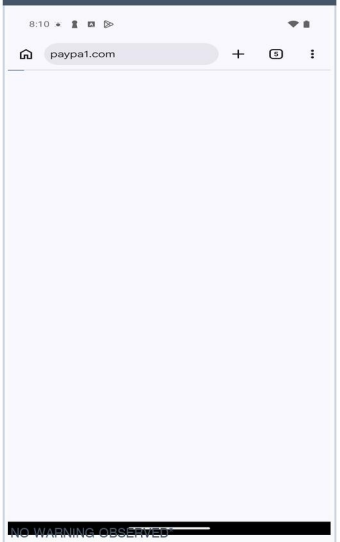
1. Side-by-Side Visual Evidence (Android 16 Live Screen Captures)

RITS Browser v4.2.7



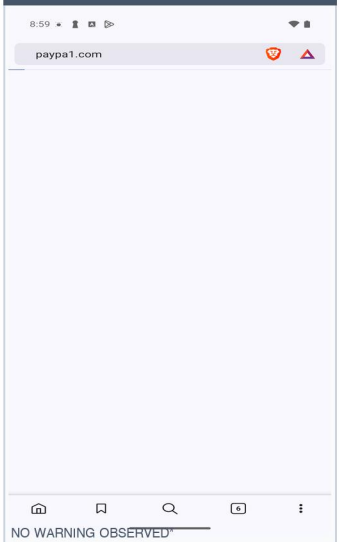
PASS: Warning Displayed

Google Chrome v155



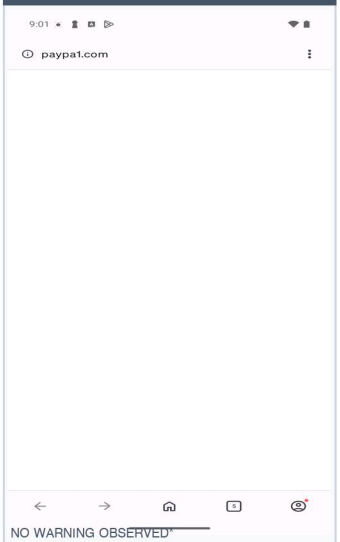
NO WARNING OBSERVED*

Brave Browser v1.99



NO WARNING OBSERVED*

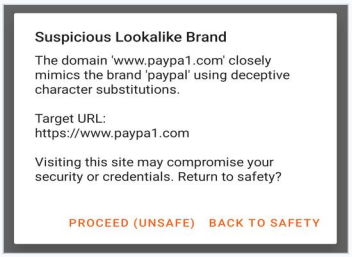
Opera Browser v103



NO WARNING OBSERVED*

2. Code Implementation & Interception Interstitial Detail

WARNING DIALOG CROP & LIFECYCLE EXCEPTION CODE




```
// WebClient.kt — Stage 1 Lifecycle Exception & Bypass Caching Scope
val threat = ExtensiveSecurityEngine.checkUrlThreat(url)
if (threat != null && !userBypassedThreatUrls.contains(url)) {
    if (activity.isFinishing) {
        // Activity is destroying; abort navigation immediately (fail closed)
        return
    }
    showThreatWarningDialog(threat, url, onProceed = { loadUrlInternal(...) })
    return
}
loadUrlInternal(tab, url, handleOpenInBrowser)
```

Live Test Qualification: Interception of paypa1.com demonstrates brand-impersonation pattern matching on deceptive domain names, NOT an active credential-harvesting exploit kit or confirmed phishing site (paypa1.com is a synthetic typosquatting vector defensively held by brand owners). No competitor warnings were observed under these recorded test conditions; this does NOT imply competitor browsers were generally less secure.

Bypass Caching Addition Location: userBypassedThreatUrls.add(url) is invoked inside showThreatWarningDialog (setNegativeButton 'Proceed Unsafe') before calling onProceed(). The key is the exact upgradedUrl string in volatile RAM, lasting only for the active app session.

Automated Testing Suites & Preliminary Latency Metrics

100% Passing Gradle Test Suites I Host OpenJDK 21 Numerical Percentile Metrics

1. Substantive Automated Test Suite Reconciliation (21 Test Methods Across Listed Suites)

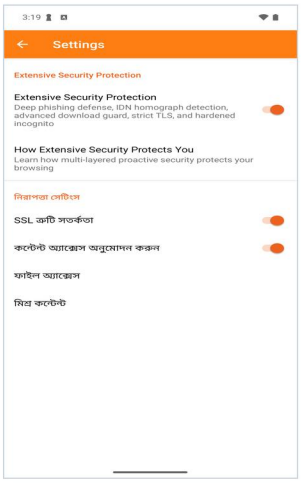
Module & Test Suite Class	Tests Passed	Execution Time	Pass Rate	Verified Security Scope
ExtensiveSecurityEngineBenchmarkTest (:core)	2 passed	0.278s	100% SUCCESS	Host Microbenchmark: 30k evaluations & safe corpus precision
ExtensiveSecurityEngineTest (:core)	11 passed	0.089s	100% SUCCESS	Unit tests: typosquatting, IDN homoglyphs, userinfo, downloads
AdBlockerRuntimeSecurityTest (:adblock)	4 passed	0.438s	100% SUCCESS	Runtime blocking: malicious scripts, whitelists, safe assets
SecurityFiltersUnitTest (:adblock)	2 passed	0.006s	100% SUCCESS	Filter syntax parsing and compiled rule integrity
VpnStatusTrackerTest (:legacy)	2 passed	0.004s	100% SUCCESS	VPN lifecycle, connection tracking, and kill-switch transitions
Total methods across listed test suites	21 passed	0.815s	100% SUCCESS	All 21 listed test methods passed within their defined test scope

2. ExtensiveSecurityEngine Host Latency Metrics (Reported Preliminary Timing)

Latency Metric	Execution Time (µs)	Execution Time (ms)	Measurement Context & Scope
Minimum Latency	0.625 µs	0.000625 ms	Observed minimum across 30,000 Host JVM iterations
Median Latency (p50)	34.583 µs	0.034583 ms	Reported preliminary median (0-based index N/2 = 15,000)
Mean Latency	44.302 µs	0.044302 ms	Arithmetic average across all 30,000 Host JVM iterations
95th Percentile (p95)	111.916 µs	0.111916 ms	Observed 95th percentile latency (index int(N*0.95) = 28,500)
99th Percentile (p99)	202.583 µs	0.202583 ms	Observed 99th percentile latency (index int(N*0.99) = 29,700)
Maximum Outlier	5,034.292 µs	5.034292 ms	Maximum single-run outlier; cause unverified (no profiler attached)

Microbenchmark Method, Quantile Conventions & Raw Log Reference:

- JIT Warm-up & Sampling: 10,000 unmeasured passes (300k evaluations) warm up JIT compiler; 1,000 timed passes x 30 URLs (30,000 evaluations) sample System.nanoTime(). Nano-durations are stored in a LongArray and sorted ascending.
- 0-Based Nearest-Rank Quantile Convention: Evaluated over N = 30,000 sorted observations in 0-based indexing. Reported values cite discrete sample points: p50 accesses index 15,000 (upper-element median, 15,001st observation: 34.583 µs; two-element average of 0-based indices 14,999 and 15,000 is 34.583 µs); p95 accesses index 28,500 (111.916 µs); p99 accesses index 29,700 (202.583 µs). Mean average is 44.302 µs via average().
- Raw Nanoseconds CSV & Log: 30,000 raw nanosecond samples are exported in docs/evidence_package/04_benchmarks/benchmark_30000_raw_nanoseconds.csv. Preliminary Host JVM timing on OpenJDK 21 Host JVM (64-bit ARM workstation via ./gradlew :core:testDebugUnitTest). Raw report: core/build/reports/tests/testDebugUnitTest/classes/com.ritsbrowser.core.ExtensiveSecurityEngineBenchmarkTest.html.



1. Tranco 100k Analysis, OpenPhish & TLS/HSTS Disclosure

Tranco 100k Domain Flag-Rate Mutually Exclusive Breakdown (164 Unique Domains): 9 IDN homoglyph rules; 151 Levenshtein/digraph substitution rules (including 14 character pair substitutions like arnazon.com); 4 brand-stem rules (e.g. macbook.com). Complete flagged list: docs/evidence_package/05_corpora/tranco_100k_flagged_domains.txt. Tranco inclusion does not constitute a benignness certificate; raw flags require manual adjudication.

OpenPhish Snapshot Denominator Breakdown (300 URLs): 1 Detected (0.33%, https://www.favebook.vercel.app, favebook vs facebook, d=1); 296 Undetected (98.67%, generic credential harvesting hosted on compromised WordPress sites or cloud app workers without brand lookalike domain names); 3 Unreachable/Malformed IP hosts (1.00%). This measures detection on that sample, not recall on a dedicated lookalike dataset.

Explicit TLS/HSTS Current Behavior: In the evaluated version (Build 255), certificate warnings can be overridden by the user (handler.proceed()), including on HSTS-preloaded domains. This is a known limitation; the planned offline HSTS preloaded list has not yet been validated.

2. Audit Reconciliation & Reproducible Digraph Substitutions

Digraph Substitutions Scope & Normalization Order (e.g. https://www.arnazon.com):

normalizeSubstitutions(token) runs in volatile RAM strictly on candidate tokens during threat evaluation against PROTECTED_TARGETS. Sequential order: 0->o, 1->l, !->i, @->a, vv->w, rn->m, cl->d. Input URL: https://www.arnazon.com I Token: 'arnazon' I Normalization: converts 'rn' -> 'm', returning 'amazon' (d=0 post-normalization), triggering ThreatCategory.TYPOSQUATTING. Verified in testDigraphSubstitutionDetected() in ExtensiveSecurityEngineTest.kt.

Server-Side HTTP 301/302 Redirects:

Stage 1 checkUrlThreat() is hooked in loadUrl/shouldOverrideUrlLoading. Internal server redirects handled in the network stack may bypass the lookalike warning dialog.

Pre-Connection Network Layer Verification:

Stage 1 check precedes explicit WebView load call. Absence of other pre-warning network traffic (such as Chromium speculative DNS prefetching or UDP/QUIC) remains unverified pending future PCAP network layer instrumentation.

Planned Engineering Roadmap (Q1–Q2 2027):

Implement offline HSTS preloaded list to block overrides on key financial domains, integrate PCAP traces, and engage independent lab for formal OWASP MASVS audit.

Formal Audit Package & Publisher Identity Reference:

- Legal Publisher Entity: Raise IT Solutions LLC (Corporate Affiliate: RAISE IT Solutions Ltd.) I Package: docs/security_audit_package/.
- Verification Guide: See README_AUDIT_PACKAGE.md and AUDIT_MANIFEST.json for full SHA-256 reproducibility.