SECURITY WHITE PAPER · WP-2026-01 REV. C (v8.2)

A Local Heuristic Warning Layer That Complements Cloud Protection

Technical architecture, two-stage verification and TLS reporting in RITS Browser v4.2.7

Author: K.A.M. Rashedul Mazid — CEO, RITS Browser; Managing Director, Raise IT Solutions LLC
Publisher: Raise IT Solutions LLC (affiliated with RAISE IT Solutions Ltd.) · 16 November 2026
Target build: RITS Browser v4.2.7 (Build 255, package acr.browser.raisebrowserfull)
Base APK SHA-256: 271a9aced1d5df4aefaf75470c716c043cb3acf77eb99f837094e77952eea9f6

Disclosure. First-party internal engineering evaluation by Raise IT Solutions LLC. It is not an independent audit, certification or academic peer-reviewed publication. External reviewers examined page images of earlier revisions only and did not independently check the APK, source code or test logs. The raw data behind every figure in this revision is publicly downloadable (Section 8).

1. Executive Summary

RITS Browser v4.2.7 implements a local heuristic warning layer for selected lookalike-domain patterns. On the documented navigation paths, the check runs before the explicit WebView loading call, subject to the lifecycle behaviour described in Section 3. 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 centralised cloud threat-intelligence feeds such as Google Safe Browsing, complemented by varying client-side models and heuristics. Cloud feeds provide broad coverage of established campaigns, but newly registered lookalike domains may not yet be listed during their first hours of operation. A local heuristic layer is intended to complement, not replace, that cloud protection.

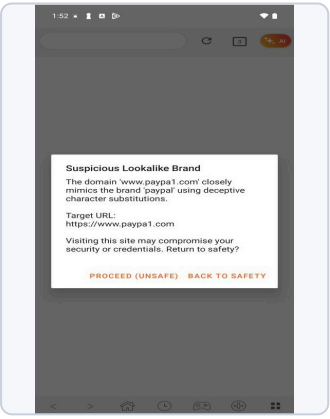
2. Two-Stage Pipeline and Threat Taxonomy

Stage 1 — Pre-navigation hook (loadUrl / shouldOverrideUrlLoading): domain resemblance (Levenshtein distance d = 1 and brand-stem rules), IDN homoglyphs after xn-- Punycode decoding, userinfo traps and sensitive HTTP paths.
Stage 2 — Post-response download callback (onDownloadStart): server-provided Content-Type and Content-Disposition filenames, to halt disguised executables such as .pdf.apk.
Network scope: no relevant TCP socket was observed in recorded snapshots. Absence of other pre-warning traffic remains unverified.

Threat category	Stage and hook	Rule and scope
Brand lookalikes (typosquatting)	Stage 1 · loadUrl / shouldOverrideUrlLoading	Levenshtein d = 1 against 100+ protected seeds, plus brand-stem rules (e.g. *book) and digraph normalisation.
IDN homoglyphs (multilingual IDN sample)	Stage 1	xn-- decoding starts script inspection; mixed Cyrillic/Greek and Latin is flagged. Single-script IDNs pass. Punycode alone is not treated as a threat.
Userinfo credential obfuscation	Stage 1	Flags userinfo structures (e.g. https://paypal.com@evil.com) before host extraction.
Insecure sensitive auth paths	Stage 1	Flags HTTP requests with sensitive path segments (e.g. http://example.test/login).
Disguised executable downloads	Stage 2 · onDownloadStart	Inspects server MIME type and Content-Disposition for double extensions (.pdf.apk).

3. Live Evidence and Interception Logic

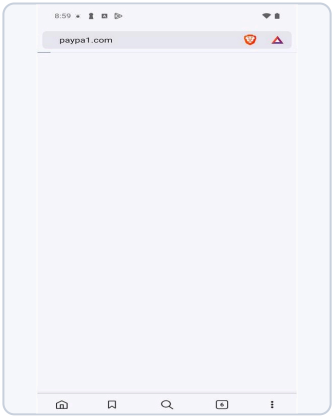
Android 16 (API 36, arm64-v8a) emulator captures, each loading https://www.paypa1.com under the same recorded conditions.



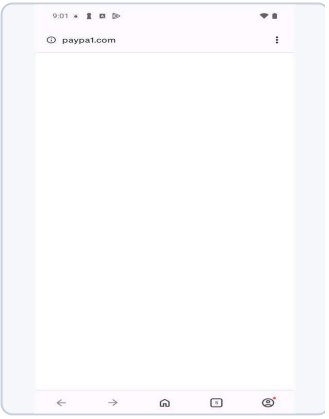
RITS Browser v4.2.7
Warning displayed



Google Chrome v155
No warning observed



Brave v1.99
No warning observed



Opera v103
No warning observed

```
// WebClient.kt — Stage 1 lifecycle exception and bypass scope
val threat = ExtensiveSecurityEngine.checkUrlThreat(url)
if (threat != null && !userBypassedThreatUrls.contains(url)) {
    if (activity.isFinishing) {
        // Activity is being destroyed: abort navigation (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 a deceptive domain name. It is not an active credential-harvesting kit or confirmed phishing site; paypa1.com is a typosquatting domain defensively held by the brand owner. No competitor warnings were observed under these recorded conditions; this does not imply the competitor browsers are generally less secure.

Bypass scope. userBypassedThreatUrls.add(url) is called inside showThreatWarningDialog (the "Proceed (unsafe)" button) before onProceed(). The key is the exact upgraded URL string, held in volatile memory for the active app session only.

4. Automated Test Suites (Gradle run, 16 November 2026)

Suite (module)	Tests	Time	Scope
ExtensiveSecurityEngineTest (:core)	25 / 25	0.078 s	Typosquatting, digraphs, IDN homoglyphs, userinfo, downloads
ExtensiveSecurityEngineBenchmarkTest (:core)	2 / 2	18.537 s*	Host microbenchmark (30,000 timed evaluations) and safe-corpus check
AdBlockerRuntimeSecurityTest (:adblock)	6 / 6	1.413 s	Malicious scripts, allowlists, safe assets
SecurityFiltersUnitTest (:adblock)	2 / 2	0.003 s	Filter syntax parsing and compiled-rule integrity
VpnStatusTrackerTest (:legacy)	1 / 1	<0.001 s	VPN connection-state transitions
VpnFullAuditTest (:legacy)	1 / 1	62.770 s	End-to-end VPN lifecycle audit (network-bound)
Total, listed security suites	**37 / 37**	**82.801 s**	**All listed methods passed within their defined scope**

* Includes the 150,000-evaluation JIT warm-up and safe-corpus test; the timed loop itself took 1,329.06 ms. Three template ExampleUnitTest methods (one per module) also passed and are excluded from the table. Per-class counts are read directly from the Gradle HTML reports in the audit package.

5. Host Latency Metrics (Reported Preliminary Timing)

Metric	µs	ms	Context
Minimum	0.625	0.000625	Observed minimum across 30,000 host-JVM evaluations
Median (p50)	34.583	0.034583	0-based index 15,000 of the sorted samples
Mean	44.302	0.044302	Arithmetic mean of all 30,000 evaluations
p95	111.916	0.111916	0-based index int(N × 0.95) = 28,500
p99	202.583	0.202583	0-based index int(N × 0.99) = 29,700
Maximum	5,034.292	5.034292	Single outlier; cause not established (no profiler attached)

Method. 5,000 unmeasured passes (150,000 evaluations) warm up the JIT compiler; 1,000 timed passes × 30 URLs (30,000 evaluations) are sampled with System.nanoTime(), stored in a LongArray and sorted ascending. Timed loop: 1,329.06 ms (mean × N = 1,329.06 ms). Environment: OpenJDK 64-Bit Server VM 21.0.9 (Homebrew) on Darwin ARM64 workstation, run 2026-11-16T02:39:56Z via ./gradlew :core:testDebugUnitTest.

> **Independent recalculation.** All 30,000 raw nanosecond values in `04_benchmarks/benchmark_30000_raw_nanoseconds.csv` were recalculated and reproduce every figure above exactly. **Why these differ from v8.1:** earlier revisions reported a different run (median 7.458 µs) whose raw samples were not retained. That run is withdrawn; only this fully reproducible run is reported. Host-JVM timing varies with machine load and is not Android-device latency.

6. Tranco Flag Rate, OpenPhish Sample and TLS/HSTS



Settings → Extensive Security Protection

Tranco Top-100k flag rate

164 unique domains flagged (0.164% raw flag rate), mutually exclusive by rule: 9 IDN homograph; 151 Levenshtein/digraph substitution (including 14 character-pair substitutions such as arnazon.com); 4 brand-stem (e.g. macbook.com). Tranco inclusion is not a benignness certificate, so this is a flag rate, not a verified false-positive rate; flags require manual adjudication. List ID 92X2W.

OpenPhish snapshot (300 URLs, 2026-11-10)

1 detected (0.33%: www.favebook.vercel.app, "favebook" vs "facebook", d = 1); 296 undetected (98.67%, generic credential harvesting without brand-lookalike names); 3 unreachable or malformed IP hosts (1.00%). This measures detection on that sample, not recall on a dedicated lookalike dataset.

TLS / HSTS current behaviour

In Build 255, onReceivedSslError suspends the load and shows a warning. Choosing to continue calls handler.proceed(); going back calls handler.cancel(). Users can currently override certificate warnings, including on HSTS-preloaded domains. This is a known limitation; the planned offline HSTS preload list has not yet been validated.

7. Reconciliation Notes and Known Limitations

Digraph normalisation

normalizeSubstitutions(token) runs in memory on candidate tokens only, against PROTECTED_TARGETS, in this order: 0→o, 1→l, !→i, @→a, vv→w, rn→m, cl→d. Example: https://www.arnazon.com → token "arnazon" → "amazon" (d = 0 after normalisation) → ThreatCategory.TYPOSQUATTING.

Server-side redirects

Stage 1 is hooked in loadUrl / shouldOverrideUrlLoading. HTTP 301/302 redirects handled inside the network stack may bypass the lookalike warning.

Pre-connection network activity

The Stage 1 check precedes the explicit WebView load call. Absence of other pre-warning traffic (for example Chromium speculative DNS prefetch or UDP/QUIC) remains unverified pending packet-capture instrumentation.

8. Reproducibility and Public Evidence

Both packages below contain a SHA-256 manifest; every listed file was verified against it (audit package 39/39, evidence package 69/69).

Package	Public download and SHA-256 of the ZIP
Security audit package (raw 30,000 timing samples, benchmark output, Gradle HTML reports, test sources, AUDIT_MANIFEST.json)	ritsbrowser.com/__l5e/assets-v1/819d9749-3e67-4965-96ca-d9eace3562bf/rits-security-audit-package-2026-11-16.zip `95c0658b23a4e9a2ad6df434410bc0ad442627f77aa6347e35093b01dd2ce892`
Evidence package (Tranco and OpenPhish corpora, flagged lists, screen captures, EVIDENCE_MANIFEST.json)	ritsbrowser.com/__l5e/assets-v1/a66c7a83-9ed8-4bdf-b516-b53e7965d4d3/rits-security-evidence-package-2026-11-16.zip `cf81623d320eed9fc0d981f25e6ed2f828149c5c2f87c17939ea6be2b23b0539`

Both packages are also linked from ritsbrowser.com/browser-security. Questions: info@raiseit-bd.com.

9. Roadmap (Q1–Q2 2027)

Ship an offline HSTS preload list that blocks certificate overrides on key financial domains; add packet-capture traces to the network evidence; publish the raw benchmark samples for each release; and engage an independent laboratory for a formal OWASP MASVS assessment.

Published by Raise IT Solutions LLC, affiliated with RAISE IT Solutions Ltd. · Document WP-2026-01 Rev. C (v8.2) · 16 November 2026 · Supersedes v8.1. Changes from v8.1: latency figures replaced with the 16 November run reproducible from public raw samples; test-suite table updated to the 16 November Gradle reports (37 security tests); APK SHA-256 corrected to match the audit manifest; public evidence links and hashes added.