Authorized Security Testing & Scanning Standard

BRSG Compliance Standard · v1.0 · effective 2026-06-25 · standards.billricestrategy.com

Authorized Security Testing & Scanning Standard

Owner: Bill Rice (BRC LLC d/b/a Bill Rice Strategy Group) **Version:** v1.0 — 2026-06-25 **Scope:** Security testing, vulnerability scanning, and related probing activity performed by BRSG or its contractors. Authorizes such activity **only** against BRSG-owned properties and BRSG-controlled accounts and infrastructure. Does not authorize, and expressly excludes, testing of any client system or any third-party platform beyond what that platform's terms permit. **Status:** Deployable. This standard is the written authorization of record for testing BRSG-owned assets and the written prohibition of record for everything else.

1. Why this standard exists

BRSG owns and operates a fleet of content and lead-generation websites and the supporting accounts that run them (hosting, DNS, CRM, CMS, email, analytics). Keeping those properties secure is a legitimate and ongoing operational need: dependency vulnerabilities, leaked secrets, misconfigured headers, exposed endpoints, and weak TLS are real risks to BRSG, its readers, and the leads it collects.

Security testing is also legally sensitive. Under the federal Computer Fraud and Abuse Act (18 U.S.C. § 1030) and analogous state computer-crime statutes, the line between lawful security work and a criminal offense is **authorization**. Scanning, probing, or testing a system you are authorized to test is legitimate security work. Doing the same thing to a system you are *not* authorized to test — a client's system, a vendor's infrastructure, or a third party's platform — can be a violation regardless of intent.

This standard draws that line explicitly. It grants standing authorization to test BRSG's own assets, defines the rules of engagement for doing so, and makes unmistakable that this authorization stops at the boundary of BRSG ownership and control. It exists so that BRSG can run a real security program on its own properties while never having a contractor, tool, or automation drift across that boundary.

2. Definitions

Security testing. Any activity intended to discover, confirm, or measure a security weakness in a system. Includes vulnerability scanning, dependency and secret scanning, configuration and header review, TLS/certificate inspection, port and service enumeration, authentication and access-control testing, and dynamic application scanning (DAST).

BRSG-owned asset. Any system, domain, account, repository, or infrastructure component owned, operated, or controlled by BRC LLC d/b/a Bill Rice Strategy Group. This includes:

- **Owned web properties** and their staging/preview deployments. The current roster of owned domains is maintained outside this standard (in the owned-properties reference and the property audit); it is not enumerated here, so this standard does not require revision when the roster changes.
- **BRSG-controlled accounts and infrastructure** under the @billricestrategy.com identity, including the Vercel team, GitHub organization/repositories, Sanity projects, domain registrar/DNS, email sending domains, the GoHighLevel sub-account, and analytics properties that BRSG owns.

@billricestrategy.com account. Any service account, login, API credential, or tenant that is provisioned under, billed to, or administratively controlled by BRSG's @billricestrategy.com identity. The presence of an @billricestrategy.com login does **not** by itself extend authorization to infrastructure that the account merely *touches* but does not own (see § 4.3).

Client system. Any system, application, environment, account, or infrastructure owned or operated by a BRSG consulting client — currently including, without limitation, ZoomCasa, Figure, Boldin, ProPair, and Infinity IPS — or by any future client. Client systems are **out of scope** under this standard. (See § 4.2.)

Third-party platform. Any system operated by a vendor or service provider that BRSG uses but does not own — e.g., Vercel, GitHub, Google, Meta, Sanity, GoHighLevel, GoDaddy, Resend, Cloudflare. BRSG's ownership of an *account* on these platforms does not extend to the platform's own infrastructure. (See § 4.3.)

Owner self-authorization. Authorization to test that arises from BRSG's ownership and control of an asset. Bill Rice, as owner of BRC LLC, may authorize testing of BRSG-owned assets under this standard without further sign-off.

Written authorization. A separate, signed, scoped engagement document — a penetration-testing agreement, a statement of work with an explicit security-testing clause, or an equivalent written grant from the system owner — required before any testing of a system BRSG does not own.

3. Authorized activities (BRSG-owned assets only)

The following activities are authorized against BRSG-owned assets as defined in § 2, subject to the rules of engagement in § 5:

1. **Vulnerability scanning** of BRSG-owned web properties and their endpoints (e.g., automated scanners against own production or, preferably, staging).
2. **Dependency and software-composition scanning** of BRSG repositories (e.g., `npm audit`, Dependabot, OSV/Snyk-style scans).
3. **Secret scanning** of BRSG repositories and build artifacts (e.g., gitleaks, trufflehog) to catch committed tokens or keys.
4. **Configuration review** of BRSG infrastructure: HTTP security headers, CORS policy, cookie flags, `robots` / `X-Robots-Tag`, redirect chains, exposure of `.env`/source maps/admin paths.
5. **TLS/certificate inspection** of BRSG domains (cipher suites, expiry, HSTS, chain validity).
6. **Port and service enumeration** of infrastructure BRSG owns or controls at the host level (note: serverless/managed hosts like Vercel are platform-operated — see § 4.3 before scanning at the network layer).
7. **Authentication and access-control testing** of BRSG-owned applications using test accounts BRSG controls — verifying that auth gates, deployment protection, and admin boundaries hold.

8. **Dynamic application security testing (DAST)** against BRSG-owned **staging/preview** environments by default; against production only under the throttling and stop conditions in § 5.
9. **Form and API abuse testing** of BRSG's own lead-capture endpoints (e.g., anti-spam origin enforcement, rate limiting, honeypot behavior) using non-destructive, clearly-marked test submissions.

4. Scope boundaries

4.1 In scope

BRSG-owned assets, and only BRSG-owned assets, as enumerated in § 2. Testing of these assets is self-authorized by ownership under this standard.

4.2 Out of scope — client systems

No client system may be scanned, probed, or tested under this standard. This includes any system owned or operated by ZoomCasa, Figure, Boldin, ProPair, Infinity IPS, or any future BRSG client — production, staging, marketing sites, APIs, or accounts — regardless of whether BRSG has been given credentials or access for consulting work.

Consulting access (e.g., a login granted to review analytics, advise on architecture, or deliver work product) is **not** authorization to perform security testing. Security testing of a client system requires a separate, signed engagement (penetration-test agreement or SOW security clause) from that client that names the systems, the window, and the permitted activities. Absent that document, treat all client systems as off-limits to scanning of any kind.

4.3 Out of scope — third-party platform infrastructure

BRSG owns accounts on third-party platforms (Vercel, GitHub, Google, Meta, Sanity, GoHighLevel, GoDaddy, Resend, and others). This standard authorizes testing of **BRSG's own content and configuration** on those platforms — it does **not** authorize testing the platform's underlying infrastructure.

Concretely: BRSG may scan its own deployed site, review its own headers, and audit its own repository. BRSG may **not** run network-layer attacks, load/stress testing, or infrastructure probing against the platform's shared systems. Each platform's acceptable-use policy governs what is permitted; where a platform requires advance notice or has a published testing policy (e.g., a bug-bounty or pentest-authorization process), that policy controls and must be followed before any activity that reaches the platform's infrastructure rather than BRSG's own content.

4.4 Always prohibited

Regardless of asset ownership, the following are never authorized under this standard:

- **Denial-of-service or resource-exhaustion attacks** (volumetric floods, intentional load testing to failure) against any production system.
- **Destructive payloads** — anything that deletes, corrupts, encrypts, or persistently alters data or configuration.
- **Exfiltration of real user or lead data**, or of third-party data, beyond the minimum needed to confirm a finding.
- **Social engineering, phishing, or pretexting** of any person, including BRSG's own contractors or vendor support staff.
- **Pivoting** from a BRSG-owned asset into any adjacent system that BRSG does not own.
- **Use of another party's credentials** without that party's specific written authorization.

5. Rules of engagement

When performing authorized testing under § 3, the following rules apply:

1. **Prefer staging.** Run intrusive or dynamic tests against staging/preview deployments first. Test production only when the finding can't be reproduced in staging, and then only read-only / non-destructive.
2. **Throttle.** Rate-limit automated scanning to avoid degrading a live property. No concurrent high-volume scans against production lead-capture endpoints.
3. **Mark test traffic.** Use identifiable markers (test email patterns, a known user-agent string, dedicated test accounts) so test activity is distinguishable from real traffic in logs and analytics.

4. **Non-destructive only.** Confirm vulnerabilities to the minimum extent necessary. Do not weaponize, persist access, or alter data to "prove" a finding.
5. **Stop conditions.** Halt immediately and reassess if testing: (a) reaches a system you cannot confirm BRSG owns; (b) returns real user/lead/PII data; (c) degrades a production service; or (d) crosses into third-party-platform infrastructure. When in doubt about ownership or authorization, stop — do not continue and ask later.
6. **Data handling.** Any sensitive data incidentally surfaced during testing (secrets, PII, credentials) is handled under BRSG's data-handling practices: not copied beyond what's needed, not retained after remediation, and rotated/remediated immediately if exposed.
7. **Contractor obligations.** Any contractor performing security testing for BRSG is bound by this standard, must confine activity to BRSG-owned assets, and must obtain BRSG's written direction before testing anything outside the in-scope list. Contractors may not subcontract testing or use BRSG's authorization as cover for testing other parties' systems.

6. Remediation and reporting

1. **Triage by severity.** Findings are triaged on a standard severity scale (critical / high / medium / low). Critical and high findings — exposed secrets, authentication bypass, data exposure — are remediated on a priority basis; medium and low are scheduled.
2. **Secret exposure is immediate.** Any leaked credential or token is rotated immediately and the exposure root-caused (consistent with BRSG's "never carry forward issues — fix root cause" practice and local-token-safety rules).
3. **Record of testing.** Each testing pass records what was tested, when, by whom, the tools used, and the findings. This record is the evidence that BRSG runs an active, scoped security program — and the evidence that testing stayed within authorized boundaries.
4. **Cadence.** Lightweight automated scanning (dependency, secret, header/TLS) runs continuously or on a regular schedule across the owned fleet. Deeper manual review is performed on a periodic basis and after any significant infrastructure change.

7. Legal basis

- **Computer Fraud and Abuse Act, 18 U.S.C. § 1030.** Authorization is the operative line between lawful security testing and unlawful access. This standard documents BRSG's authorization over its own assets and its prohibition on testing assets it does not own.
- **State computer-crime statutes.** Most states have analogues to the CFAA (e.g., unauthorized-access and computer-trespass statutes). The same ownership/authorization boundary applies.
- **Third-party acceptable-use and testing policies.** Vendor platform terms (Vercel, GitHub, Google, and others) govern what testing, if any, is permitted against their infrastructure. Those policies control over this standard for any activity that reaches platform infrastructure rather than BRSG-owned content.
- **Client engagement terms.** Testing of a client system is governed exclusively by a separate signed agreement with that client — never by this standard.

Version History

VERSION	DATE	CHANGE
v1.0	2026-06-25	Initial standard. Authorizes testing of BRSG-owned assets only; expressly excludes client systems and third-party-platform infrastructure. Roster of owned domains kept in the owned-properties reference, not enumerated in this standard.

When this standard is cited as the authorization of record for a testing engagement, cite it **by version** (e.g., "Authorized Security Testing & Scanning Standard v1.0") so the authorization is pinned to a fixed text. The version and date in the header above are authoritative.

8. Decision rule (quick test before any scan)

Before running any security test, answer one question: **Does BRSG own and control this asset?**

- **Yes, BRSG owns it** → authorized under this standard; follow the rules of engagement in § 5.

- **No — it's a client system** → stop. Requires a separate signed security-testing agreement with that client (§ 4.2).
- **No — it's third-party platform infrastructure** → stop. Governed by that platform's acceptable-use/testing policy, not this standard (§ 4.3).
- **Not sure** → treat as "no." Do not test until ownership and authorization are confirmed in writing.