BizBlasts Security Policy

Version 1.0 - Effective 2025-12-11

Purpose
This policy defines the security principles and controls used to protect the BizBlasts application, its tenants (businesses), and their customers. It focuses on application-layer security, data protection, operational safeguards, and third-party integrations.

1. Scope

- Applies to the BizBlasts Rails application, supporting infrastructure, and production deployments.
- Covers authentication/authorization, multi-tenant isolation, data protection, logging, uploads, and external service integrations.
- Does not replace vendor (Stripe/Twilio/AWS/etc.) policies; it describes how BizBlasts uses them securely.

2. Architecture and Hosting

- Rails app server (Puma) behind a TLS-terminating edge (Render) and trusted proxy handling (Cloudflare-aware configuration).
- Primary datastore is PostgreSQL; production can use dedicated logical connections for primary, cache, queue, and cable.
- Background processing uses ActiveJob with SolidQueue (database-backed), and caching uses SolidCache (database-backed).

3. Identity, Authentication, and Authorization

- Authentication: Devise with support for passwordless/magic-link flows and JWT-based API authentication where applicable.
- Authorization: Pundit policies enforce role- and resource-based access controls.
- Multi-tenant isolation: acts_as_tenant provides tenant scoping; host-based routing supports subdomains and verified custom domains.
- Cross-domain authentication continuity uses short-lived, single-use token bridging for transitions between platform and custom domains.

4. Data Protection

- Encryption at rest: sensitive phone fields are protected using Active Record Encryption with deterministic encryption for safe equality queries.
- Secrets management: sensitive credentials are sourced from Rails credentials and environment variables; production avoids logging secret presence.
- Data minimization: logs and operational telemetry avoid storing customer PII in clear text.

5. Transport Security and Sessions

- Production traffic is intended to use HTTPS; the application is configured to assume SSL and force SSL in production deployments.
- Session cookies are configured with secure defaults (HttpOnly, SameSite) and

environment-appropriate domain scoping.
- HSTS is recommended at the edge for platform and custom domains.

6. Web Security Controls

- Content Security Policy (CSP) is hardened to reduce XSS risk (nonce-based scripts; unsafe-inline/unsafe-eval removed where feasible).
- CSRF protection is enabled for browser sessions. Frontend AJAX requests include the CSRF token header for state-changing requests.
- Rate limiting is enforced with rack-attack to mitigate brute force and abusive traffic patterns.

7. Webhook and Integration Security

- External webhooks (e.g., Stripe, Twilio) are authenticated using signature verification. Requests are verified before reaching controllers via middleware.
- Stripe webhooks are queued for asynchronous processing to isolate failures and improve reliability.
- OAuth flows (Google/Microsoft calendar integrations) use state/redirect validation to prevent CSRF and token substitution attacks.

8. Logging, Monitoring, and Privacy

- Security events are logged using a dedicated security logging approach (authentication, authorization failures, sensitive actions).
- PII redaction: logs that may include email/phone/payment identifiers are sanitized via SecureLogger-style redaction rules.
- Health endpoints avoid returning sensitive environment or infrastructure details.

9. File Upload Security

- Uploads are handled by ActiveStorage with server-side validations and centralized allowlists/limits.
- Production storage uses S3; large upload handling includes timeouts and security logging for anomalous uploads.
- Future hardening may include malware scanning and image metadata (EXIF) stripping as defense-in-depth.

10. Dependency and Vulnerability Management

- Static analysis and security scanning are part of the development process (e.g., Brakeman; repository-integrated scanning such as CodeQL/Dependabot where configured).
- Security-relevant changes should include tests and documentation updates when applicable.

11. Incident Response

- Report suspected vulnerabilities or security incidents to the BizBlasts team via the configured support/security contact channel.
- Incidents are triaged, contained, and remediated. Post-incident reviews capture root

cause, corrective actions, and prevention measures.
- Where required, affected tenants/users are notified in a timely manner consistent with contractual and legal obligations.

12. Third-Party Services

- Payments: Stripe (payments, checkout, connect, webhooks).
- Email: Resend. SMS: Twilio.
- Storage: AWS S3. Edge/hosting: Render; trusted proxy awareness: Cloudflare.
- Calendar/video: Google Calendar/Meet APIs; Microsoft Graph APIs.

Document Control
Owner: BizBlasts Engineering
Review cadence: at least annually or upon major architecture/security changes
Last generated: 2025-12-11