# Privileged Access Logging

School Administration Services Limited trading as EasyBus

**Version:** 1.1
**Date:** June 2026
**Owner:** Greig Neilson, Network Manager
**Next review:** April 2027

## 1. Purpose

This document defines what access and activity events are logged for privileged users of the EasyBus platform, where those logs are stored, how long they are retained, and the process for reviewing them.

Privileged users are those with super admin or network admin roles. Logging of privileged access supports the EasyBus Privacy Breach Notification Process and aligns with the logging controls in the ST4S assessment framework (§6.2.4 Security — Logging).

## 2. What is logged

The following events are logged for all privileged user sessions.

| Event | Log source | Detail captured |
|---|---|---|
| **Super admin login / logout** | Supabase Auth logs | Timestamp, IP address, user ID |
| **Super admin MFA challenge** | Supabase Auth logs | TOTP challenge events — success and failure. MFA is enforced for the super admin role. |
| **Reads of student PII tables (students, parents, student_parents)** | Postgres logs via pgaudit object audit | Timestamp, connection role, statement text. Parameter values (the specific student or parent ID queried) are captured pending a Supabase platform configuration request; until that lands, audit lines record that a read occurred and from which role, but not which specific row. |
| **Modifications to student and parent records (INSERT, UPDATE, DELETE)** | EasyBus platform audit log (activity_log table) | User, timestamp, full before and after row snapshots, list of changed columns |

| Event | Log source | Detail captured |
|---|---|---|
| **Modifications to ~85 other administered tables (routes, schools, operators, drivers, vehicles, schedules, service requests, committee records, documents, compliance records, and so on)** | EasyBus platform audit log (activity_log table) | Same as above. The complete list of audited tables is maintained in the source migration 20260512181304_activity_log_triggers.sql. |
| **Bulk data exports (CSV and PDF downloads from the admin portal)** | EasyBus platform audit log (activity_log table) | User, timestamp, entity exported, scope criteria (school, network, route, date range), row count. Captured at 14 export points across the admin portal. |
| **User account creation or deletion** | Supabase Auth logs | Timestamp, user ID, email |
| **Role changes** | EasyBus platform audit log (activity_log table) | Both changes to user_profiles.role and direct mutations of auth.users.raw_app_meta_data are captured. Direct mutations are captured by a dedicated trigger that logs only the role field, not the entire user record. |
| **Failed authorisation attempts (row-level security denials)** | Postgres logs | Timestamp, role, denial reason. Reviewed during incident response and as part of the annual log review. |
| **Service-role API access** | Supabase API logs | All API requests are logged by Supabase including those using the service role key. Distinguishing application from non-application sources is not implemented at the application layer; mitigation is service-role key custody, restriction to edge functions, and rotation on personnel change. |

The application audit log is append-only. Direct INSERT, UPDATE, DELETE and TRUNCATE on the activity_log table is revoked from all application roles (anon, authenticated, service_role). Writes are only possible through the SECURITY DEFINER trigger function. Even a compromised service-role key cannot rewrite or delete past audit entries.

## 3. Log storage and retention

| Log stream | Storage location | Retention |
|---|---|---|
| **EasyBus platform audit log — entries about students, parents, guardians** | activity_log table in the EasyBus database | 1 year after the related student is archived (purged with the underlying record) |
| **EasyBus platform audit log — entries about all other administered records** | activity_log table in the EasyBus database | 7 years |
| **Supabase Auth logs** | Supabase managed log store | Per Supabase plan retention (currently Pro plan default) |
| **Postgres logs (including pgaudit object audit and row-level security denials)** | Supabase managed log store | Per Supabase plan retention |
| **Supabase API logs** | Supabase managed log store | Per Supabase plan retention |

The Supabase project is hosted in ap-southeast-2 (Sydney). All log data therefore remains in Australia.

Logs are not accessible to end users. Access to raw Supabase logs requires super admin credentials and direct access to the Supabase dashboard. The application audit log is surfaced at /activity in the admin portal for super admins and network admins (network admins see only their network's rows).

## 4. Log review process

Logs are reviewed in the following circumstances:

- **Following any suspected or confirmed privacy breach —** reviewed immediately as part of the breach notification process.
- **Following any complaint or concern about data handling —** reviewed within five working days of the complaint.
- **Annually —** a routine review of privileged access logs is conducted as part of the annual data retention review.
- **Continuously —** application errors and row-level security denial spikes are surfaced by Sentry alerts to the Network Manager.

Findings from any log review are documented and retained for seven years alongside the privacy incident register.

## 5. Responsibility

Greig Neilson (Network Manager) is responsible for maintaining access to logs, conducting reviews, and acting on any anomalies identified. If an anomaly is identified that may constitute a privacy breach, the Privacy Breach Notification Process is followed immediately.

## 6. Review

This document is reviewed annually. Next review due April 2027.