Glossary

*One definition per term, with the API names and labels exactly as the package ships them. The version installed in your org can be read in **Setup → Installed Packages**.*

Managed package namespace: `orgguardian`. In a subscriber org, every API name below is prefixed with `orgguardian__` (e.g. `orgguardian__Finding__c`).

What the package puts in your org:

Component	Count	Section
Lightning app	1	§1
Tabs (8 of them in the app's navigation bar)	43	§1
Custom objects	20	§2.1
Custom settings	9	§2.2
Big Objects (trend history)	7	§3
Platform events	2	§4
Custom Metadata Types (configuration)	37	§5
Permission sets	5	§6
Permission set groups (personas)	4	§6
Custom permissions	3	§6
Named Credential	1	§6
Lightning web components	174	—

Alongside these, the package ships, among other components, the code and Lightning pages behind its screens, and the reports and dashboards listed in the Operations Guide §4.

---

1. App and navigation

Label	API name	Definition
OrgGuardian (app)	`OrgGuardian`	The package's only Lightning app. Its navigation bar carries **8** tabs; the other 35 are reached through the App Launcher or through the "category" tabs.
Home	`OrgGuardian` (tab)	The product's entry point. Opening it for the first time, as an authorised user, **triggers the bootstrap** (scheduled jobs + reference scan).
Executive	`OrgGuardian_Tab_Executive`	Summary screen for executive management.

Label	API name	Definition
Findings	`OrgGuardian_Tab_Findings`	Main screen for working on findings.
Monitoring	`OrgGuardian_Tab_Monitoring`	Screen for monitoring the org's health.
Connectivity	`OrgGuardian_Tab_Connectivity`	Screen for tracking outbound integrations and their health.
Alerts	`OrgGuardian_Tab_Alerts`	Screen listing the alerts sent.
Drift & audit	`OrgGuardian_Tab_Drift`	Screen for tracking configuration changes and the audit trail.
Administration	`OrgGuardian_Cat_Admin`	"Category" tab: groups the package's administration screens.

Four "category" tabs serve as indexes to families of screens: `OrgGuardian_Cat_Admin` (**Administration**, the only one of the four pinned in the navigation), `OrgGuardian_Cat_Ciso` (**All executive screens**), `OrgGuardian_Cat_Security` (**All security screens**) and `OrgGuardian_Cat_Monitoring` (**All monitoring screens**).

> **No object tabs.** The 43 tabs are Lightning page or Lightning component tabs; none of them is an object tab. Records are therefore viewed through the product's screens, not through a standard object tab.

The other 35 tabs (API name → label): `OrgGuardian_Agentforce` → Agentforce · `OrgGuardian_AlertAnalytics` → Alert analytics · `OrgGuardian_AlertRouting` → Alert routing · `OrgGuardian_Ask` → Ask the org · `OrgGuardian_Baseline` → Approved baseline · `OrgGuardian_BlastRadius` → Config impact · `OrgGuardian_Callouts` → Outbound callouts · `OrgGuardian_Certs` → Certificates · `OrgGuardian_ChangeFeed` → Change feed · `OrgGuardian_ChannelSettings` → Alert channels · `OrgGuardian_Charts` → Charts · `OrgGuardian_Ciso` → Executive summary · `OrgGuardian_Compliance` → Compliance · `OrgGuardian_Config` → Settings · `OrgGuardian_Connections` → Connections · `OrgGuardian_Dashboard` → Dashboard · `OrgGuardian_Drift` → Configuration drift · `OrgGuardian_Enablement` → Enablement · `OrgGuardian_Endpoints` → Endpoints · `OrgGuardian_Fleet` → Fleet · `OrgGuardian_IncidentDrill` → Incident drill · `OrgGuardian_Incidents` → Incidents · `OrgGuardian_Insights` → Insights · `OrgGuardian_Limits` → Limits · `OrgGuardian_Parity` → Environment parity · `OrgGuardian_PerfGrid` → Weekly health grid · `OrgGuardian_Plans` → Subscription · `OrgGuardian_Remediation` → Remediation plan · `OrgGuardian_RiskHeatmap` → Risk heat map · `OrgGuardian_Security` → Security · `OrgGuardian_Trends` → Trends · `OrgGuardian_ValueLedger` → Value ledger · plus the three unpinned "category" tabs `OrgGuardian_Cat_Ciso`, `OrgGuardian_Cat_Security` and `OrgGuardian_Cat_Monitoring`. That makes 32 tabs named above + 3 = **35**.

The labels quoted, in this section as in §2 to §6, are the shipped labels, in English: the package ships no translation of them.

---

2. Product data objects (`__c`)

29 in total: **20 custom objects** and **9 custom settings**.

2.1 Objects that carry results

API name	Label	Definition	Where it appears
`Finding__c`	**Finding**	The product's atomic unit: a condition detected on the org, with its severity, its category, its target and its deduplication key.	**Findings** (the list is its **Findings** view)
`Finding_Activity__c`	**Finding Activity**	Immutable log line on a finding: acknowledgement, snooze, risk acceptance, resurfacing.	**Findings** (its **Findings** view: a finding's **Activity**), **Change feed**
`Incident__c`	**Incident**	Correlated grouping of several findings into a single event; SLA, MTTA and MTTR live **at incident level**, never at finding level.	**Incidents**; the **Incidents** view of **Findings**
`Scan_Run__c`	**Scan Run**	One run of the detection chain: timestamp, scope, outcome.	**Home** (**Setup tour**), **Insights** (**Posture Change Timeline**), the start-up banner
`Org_Metric_Current__c`	**Current Org Metric**	The current value of a posture or health metric.	**Home**, **Monitoring**, **Dashboard**, **Executive**, **Insights** and many other screens
`Insight__c`	**Insight**	An editorial statement derived from findings and metrics — the explanation, not the detection.	**Insights**
`Attestation__c`	**Attestation**	Traced compliance decision on an article of a framework: decision, justification, author (opaque identifier), server timestamp, review due date.	**Compliance**
`Posture_Report__c`	**Posture Report**	Periodic posture report produced by the `OrgGuardian Board Report` scheduled job.	**Executive summary** (board report), **Value ledger**
`Org_Sla_Report__c`	**Org SLA Report**	Report of the service-level commitments measured over the period.	**Endpoints**
`Discovered_Endpoint__c`	**Discovered Endpoint**	An outbound destination **discovered** by static analysis of the org's code, before any monitoring decision.	**Endpoints**

API name	Label	Definition	Where it appears
`Endpoint_Status__c`	**Monitored Endpoint**	An endpoint **selected** for monitoring, with its current health status.	**Endpoints**
`Config_Baseline__c`	**Approved Configuration Baseline**	The security configuration state validated by your committee, frozen at a date, signed by the person who promoted it and backed by a written justification. Append-only register: promoting a new baseline replaces the previous one without modifying it. The record contains the approved fingerprints themselves, so the retention purge of snapshots does not empty it.	**Approved baseline**
`Config_Baseline_Item__c`	**Approved Baseline Component**	The setting-level detail of a signed baseline: one row per approved component, written once at approval, checked against the seal of its baseline before it is shown. It is read with its baseline and deleted with it.	**Approved baseline**
`Config_Dependency__c`	**Configuration Dependency**	An edge of the configuration dependency graph, the raw material for the config impact calculation.	**Config impact**
`Callout_Log_c`	**Callout Log**	Trace of an instrumented outbound callout. Written **only** if callout instrumentation is enabled; otherwise zero rows. It also carries the Id of the user whose transaction made the call (`Caller_Id__c`: a user Id read inside the org only, to count the distinct callers of an incident's failed calls; not readable by `OrgGuardian_Viewer`).	**Outbound callouts**
`Itsm_Ticket_c`	**ITSM Ticket**	Local reference to a ticket created in an external ITSM tool. Paid-plan feature, off by default. A ticket comes from a click on a finding, a click on an incident (the line is then attached to its most severe open finding, the only one the ticket's closure acknowledges) or the automatic creation policy; a finding that already carries a line that is not closed gets no second one (a refusal line blocks for 7 days only). An **Open** line with **no external reference** is a ticket whose reference OrgGuardian did not receive — the tool did not answer in time, or its answer did not carry it: check it in the tool. A **Refused** line keeps a payload the tool or the personal-data guard refused: no other ticket is created for that finding for 7 days, then it is a candidate again. An administrator can lift that refusal sooner, from the finding or the incident (**Allow a new ticket**): the line then turns **Closed**, with no reference, and the automatic creation no longer retries that finding — create its ticket by hand.	**Alert channels** (**ITSM tickets**); created from a finding, from an incident or by the automatic creation policy
`Linked_Org__c`	**Linked Org**	Another org attached for fleet monitoring. Paid-plan feature, off by default.	**Fleet**

API name	Label	Definition	Where it appears
`Fleet_Finding__c`	**Fleet Finding**	The fleet inbox: an open finding that a linked org publishes, kept in the managing org for cross-org triage — severity, category, source, occurrences, dates and triage status. Never its target field, its message or the record Id of the linked org, which survives only as a SHA-256 hash that no permission set other than `OrgGuardian_Admin` can read; the source, for its part, names the finding's target for some families (a host, a metric, an audited setting), never a user — a source that names one is not kept. The triage status stays local, is never written to the linked org, and its history says who triaged. Purged on the findings retention window of the managing org. Paid-plan feature, off by default.	**Fleet** (fleet inbox card)
`System_Log__c`	**System Log**	Technical log **specific to the package**: errors, warnings, the history of sent and suppressed alerts, egress refusals (category `OUTBOUND_BLOCKED`). It is not a log of customer data.	No dedicated screen (the alert volumes of **Alerts** and **Alert analytics** are computed from it); the Security FAQ §3 gives a query
`OrgGuardian_PurgeCursor__c`	**OrgGuardian Purge Cursor**	Resume position of the retention purge between two runs. An ordinary custom object, not a custom setting.	None (retention job only)

2.2 Custom settings

API name	Label	Definition
`OrgGuardian_Channel_Settings__c`	**OrgGuardian Notification Channels**	**Protected** hierarchy setting for the outbound channels, ITSM and SIEM. It holds four secrets (details below).
`OrgGuardian_Heroku__c`	**OrgGuardian Heroku External Service**	**Protected** hierarchy setting for the Heroku option. It holds three secrets (details below).
`OrgGuardian_Feature_Mute__c`	**OrgGuardian Feature Mute**	**Protected** list setting: mutes a feature to cut the noise, one record per feature. A mute can only switch a feature off, never on.
`OrgGuardian_Install__c`	**OrgGuardian Installation**	Installation marker: timestamp, version, upgrade flag, previous version. An **ownerless** store, the only write possible under the package's system user.
`OrgGuardian_Demo__c`	**OrgGuardian Demo**	**Protected**; used by the publisher's demo orgs only; no effect in a customer org.
`OrgGuardian_Plan_Trial__c`	**OrgGuardian Plan Trial**	**Protected** setting: the start date of the 30-day trial of the full plan, armed at the first screen opening by an administrator. A trial once started stays started for good, expired or not. It also records the log retention window the nightly purge last applied, and when.

API name	Label	Definition
`OrgGuardian_Pseudonym__c`	**OrgGuardian Subject Pseudonym**	**Protected** hierarchy setting: the key, specific to your org, that replaces a user's Id with a pseudonym in the deduplication key of per-user findings (HMAC-SHA256). It holds one secret, generated in the org and never sent, logged, displayed or exported; no permission set grants it. It is created automatically the first time a user holding Customize Application opens an OrgGuardian screen; until then, per-user findings are reported as not measured.
`OrgGuardian_Alert_Preference__c`	**OrgGuardian Alert Preference**	**Protected** hierarchy setting: each user's alert delivery preferences — minimum severity for email and for the in-app notification, muted categories, quiet hours — set by an administrator holding Customize Application. No secret.
`OrgGuardian_Compliance_Map__c`	**OrgGuardian Compliance Map**	**Protected** hierarchy setting: the Id of the version of the imported compliance mapping in force for the whole org (`Override_Version_Id__c`). Written only by the import and by the return to the shipped mapping, which require Customize Application; empty, the shipped mapping applies (the Configuration Guide, "Compliance frameworks"). An opaque record Id, nothing personal, no secret.

What the Notification Channels and Heroku settings hold. Both are invisible and not queryable from your org: only the package's code reads them.

- `OrgGuardian_Channel_Settings__c` carries the Named Credential references for the channels (Slack, Teams, PagerDuty, webhook), for ITSM and for SIEM, the sender address for alerts, the Chatter group identifier, the ITSM settings and the ITSM and SIEM consents, the webhook consent, the automatic ITSM ticket creation policy (`Itsm_Auto_Min_Severity__c`, the threshold, `Itsm_Auto_Consent__c`, the date of the separate consent, no identity, and `Itsm_Auto_Breaker__c`, the automatic creation's circuit breaker), and Slack reactions (`Slack_Channel_Id__c`, the channel ID; `Chat_Reactions_Consent__c`, the consent date, no identity; `Slack_Bot_Named_Credential__c`, the name of the bot's Named Credential, separate from the incoming webhook's; `Slack_Allowed_Members__c`, the optional list of Slack members whose reactions count; the bot token stays in the External Credential, never here). **It also contains four secrets**, empty at installation and entered by your administrator: the HMAC signing secrets for the webhook, ITSM and SIEM (`Webhook_HMAC_Secret__c`, `Itsm_Hmac_Secret__c`, `Siem_Hmac_Secret__c`) and the PagerDuty integration key (`PagerDuty_Routing_Key__c`): four of its **26 fields**.
- `OrgGuardian_Heroku__c` carries the consent (`Consent_Granted__c`), the target collector (Named Credential name or HTTPS address, `Endpoint__c`), the declared region (`Region__c`), the start of the 30-day trial (`Trial_Start_At__c`), the date and author of the last activation request (`Activation_Requested_At__c`, `Activation_Requested_By__c`: a user Id that the activation request does not send), and the collector that issued the viewer token (`Issuing_Collector__c`: the viewer link is composed for that collector only). **It also contains three secrets**, empty at installation: the tenant HMAC key, which signs what is sent (`Hmac_Key__c`), and the viewer token, which opens the Heroku dashboard read-only to anyone holding the full link (`Viewer_Token__c`), both retrieved from the collector once the activation is approved, or entered by your administrator; and the claim secret (`Activation_Claim_Secret__c`), created by your org at the activation request (the collector holds only its SHA-256 fingerprint), presented to retrieve the pair, then erased: three of its **10 fields**.

3. History objects (`__b`, Big Objects) — 7

API name	Label	Definition
`Compliance_Snapshot__b`	**Compliance Snapshot**	Compliance evidence frozen at a date (job `OrgGuardian Compliance Snapshot`).
`Config_Snapshot__b`	**Configuration Snapshot**	History of configuration states (one capture every N hours, adjustable) and drift reference (last known state, one row per component).
`Em_Metric_Snapshot__b`	**Event Monitoring Snapshot**	Metrics derived from Event Monitoring, kept as history.
`Endpoint_Health_Snapshot__b`	**Endpoint Health Snapshot**	Availability and latency history per monitored endpoint.
`Login_Baseline__b`	**Login Baseline**	Baseline of login behaviour, the reference for anomaly detection.
`Org_Limit_Snapshot__b`	**Org Limit Snapshot**	Historical consumption of the platform limits.
`Prediction_Outcome__b`	**Prediction Outcome**	Observed outcome of a forecast, the raw material for replay and for measuring accuracy.

4. Platform events (`__e`) — 2

API name	Label	Definition
`Org_Health_Alert__e`	**Org Health Alert**	Platform event published for health alerts. ⚠ A platform event is delivered **without re-applying the sender's field-level security**: a Pub/Sub subscriber receives the entire payload. That is why including the advisory prose in the payload is a separate switch, off by default.
`Callout_Log_Event__e`	**Callout Log Event**	Carries the observation of a call instrumented by `OrgGuardianCallout.execute` (entry point, host only, status, duration, error type) to the package's trigger, which saves it as a `Callout_Log__c` row outside the caller's transaction. Published immediately, only if callout instrumentation is enabled. Publishing requires create access on this event: it is the only right that an integration user calling `execute` needs (the Configuration Guide §3).

5. Configuration types (`__mdt`) — 37

These are **Custom Metadata Types**: versionable configuration, not data. Only one is shipped with **Protected** visibility (invisible to and not editable by the subscriber): `License__mdt`.

API name	Label	Role in one line
`Acceptance_Policy__mdt`	Risk Acceptance Policy	Rules for accepting a risk, and its duration

API name	Label	Role in one line
`Alert_Channel_Config__mdt`	Alert Channel Configuration	Settings of an alert output channel
`Alert_Routing_Rule__mdt`	Alert Routing Rule	Which alert goes to whom
`Attack_Path_Rule__mdt`	Attack Path Rule	Sequence of steps which, when observed on the same principal, make up a named attack path; creates no finding
`Benchmark_Cohort__mdt`	Benchmark Cohort	Comparison group for the indicators
`Branding__mdt`	Branding	Brand applied to the outputs
`Compliance_Framework_Setting__mdt`	Compliance Framework Setting	Enabling and tuning of a framework
`Criticality_Scope__mdt`	Criticality Scope	What is deemed critical in this org *(written by the **Business criticality scopes** pane; the business unit dimension can only raise a severity)*
`Data_Jurisdiction__mdt`	Data Jurisdiction	Mapping between a domain or an instance and its jurisdiction (country, GDPR adequacy), computed without an outbound callout
`Detection_Rule__mdt`	Detection Rule	A declarative detection rule
`Detector_Setting__mdt`	Detector Setting	Fine-tuning of a detector
`Digest_Config__mdt`	Digest Configuration	Cadence and content of the posture digest
`Email_Alert_Template__mdt`	Email Alert Template	Template of an alert message
`Endpoint_Tag_Rule__mdt`	Endpoint Tagging Rule	Automatic classification of endpoints
`Escalation_Chain__mdt`	Escalation Chain	Order of the recipients of an escalation
`Escalation_Policy__mdt`	Escalation Policy	Conditions that trigger an escalation
`Feature_Gate__mdt`	Feature Gate	Turns a feature on or off; the subscriber can override the default value shipped with the package
`Fleet_Policy__mdt`	Fleet Policy	A declared fleet policy — minimum score, maximum open findings, age of the last sync, differences from the golden org… —, rated in the managing org against what each linked org published; no record ships
`Gate_Label__mdt`	Feature Label	The displayed label of a feature governed by a switch
`License__mdt`	License	The org's licence plan. **Protected**: neither readable nor editable from the subscriber org

API name	Label	Role in one line
`Maintenance_Window__mdt`	Maintenance Window	Period during which alerts are suspended
`Monitor_Threshold__mdt`	Monitoring Threshold	Trigger threshold of a metric
`Org_Insight__mdt`	Insight Editorial	Editorial text associated with an insight
`Org_Setting__mdt`	Org Setting	General settings, including the retention windows
`Parity_Rule__mdt`	Environment Parity Rule	Settings that must be identical between the UAT sandbox and production
`Posture_Narrative__mdt`	Posture Narrative Segment	Text fragment of the posture outputs
`Posture_Recommendation__mdt`	Posture Recommendation	Recommendation associated with a finding
`Posture_Sla__mdt`	Posture SLA	Time commitment for handling a finding
`Recipient_Delivery_Pref__mdt`	Delivery Preference	Delivery preferences per recipient
`Release_Calendar__mdt`	Salesforce Release Calendar	Dates of the seasonal releases, to prepare for upgrades without an outbound callout
`Remediation_Effort__mdt`	Remediation Effort	Declared estimate of the effort per family of findings, to order the remediation plan
`Remediation_Playbook__mdt`	Remediation Playbook	Recommended procedure — **advice, never execution**
`Report_Template__mdt`	Report Template	Template of a report *(written by the **Posture digest** pane; it also carries the recipient permission set, the frequency and the board pack of a report by recipient)*
`Risk_Weight__mdt`	Risk Weight	Weight of a factor in the risk score
`Root_Cause_Rule__mdt`	Root Cause Rule	Grouping of findings by common cause
`Scan_Window_Policy__mdt`	Scan Window Policy	When detection is allowed to run
`Sensitive_Principal__mdt`	Sensitive Principal	Profiles and permission sets (groups included) judged sensitive: assigning one to a user raises a priority alert

> **Installation note.** The package writes **26** of these 37 types in the subscriber org, from its no-code editors; it never writes to the other 11. That does not make them read-only for you: these types are public (except `License__mdt`, which is protected), and you can create your own records in them, with all of their fields — for example a `Data_Jurisdiction__mdt` row for a domain in your estate, which takes precedence over a less specific shipped row. On the records the package ships, most fields are controlled by the publisher and updated by package upgrades; a few, such as the active flag on several types, remain yours to change. The type or types that each editor writes are listed in the Configuration Guide §2.

---

6. Access rights

Label	API name	Definition
OrgGuardian: technical administration	`OrgGuardian_Admin`	Permission set. The only one in the package that carries the custom permissions `OrgGuardian_Manage_Features` and `OrgGuardian_Run_Incident_Drill`; it also grants create and edit on `Finding__c` and `Org_Metric_Current__c`, which a user needs to set up the scheduled jobs.
OrgGuardian: finding handling	`OrgGuardian_Analyst`	Permission set. Working on findings: acknowledging, assigning, snoozing, accepting a risk.
OrgGuardian: audit and compliance	`OrgGuardian_Auditor`	Permission set. Viewing oriented towards evidence and attestation.
OrgGuardian: read-only access	`OrgGuardian_Viewer`	Permission set. Read-only.
OrgGuardian: financial amounts	`OrgGuardian_Financials`	Additive permission set that unlocks the display of amounts, through the custom permission `OS_View_Financials`.
OrgGuardian: Salesforce admin persona	`OrgGuardian_Persona_AdminSF`	Permission set group: Viewer + Analyst + Admin.
OrgGuardian: CISO persona	`OrgGuardian_Persona_RSSI`	Permission set group: Viewer + Auditor + Financials. `RSSI` is the French abbreviation for CISO.
OrgGuardian: IT department persona	`OrgGuardian_Persona_DSI`	Permission set group: Viewer + Financials. `DSI` is the French abbreviation for IT department.
OrgGuardian: executive leadership persona	`OrgGuardian_Persona_Direction`	Permission set group: Viewer + Financials. `Direction` is the French word for executive leadership.

Label	API name	Definition
OrgGuardian: manage features	`OrgGuardian_Manage_Features`	Custom permission accepted by OrgGuardian's own check before an administration action: running the administrator commands (the Installation Guide §5.3), saving from the no-code editors, muting or re-enabling a feature. That check also accepts the platform privilege "Customize Application" or "Modify Metadata Through Metadata API Functions" in its place. The permission does not replace the platform privilege that the write itself requires ("Modify Metadata Through Metadata API Functions" for the Custom Metadata editors, "Customize Application" alone not being enough; "Customize Application" for a protected setting): on its own, it does not let a user save Custom Metadata from the no-code editors, nor write a protected setting such as a mute (the Configuration Guide §2, "Who is allowed to save").
View financial amounts	`OS_View_Financials`	Custom permission that governs the display of amounts.
OrgGuardian: incident drill	`OrgGuardian_Run_Incident_Drill`	Custom permission that allows launching an incident drill in **rehearsal** mode and reading the scenario catalogue, without granting administration authority; the **live** mode, which really notifies the on-call rota, still requires that authority. In the package, only `OrgGuardian_Admin` carries it.
OrgGuardian self-org connection	`OS_SelfOrg`	The **only** Named Credential shipped, used by the self-callouts: it is meant to point to your org's own My Domain and ships with a non-resolving placeholder address, which your administrator replaces before enabling a self-callout. An External Credential of the same name carries the authentication binding. Step-by-step wiring: the Configuration Guide §3, “Wiring the `OS_SelfOrg` self-callout”.

Details of the personas and of access: Permissions and Personas guide.

---

7. Cross-cutting concepts

Term	Definition
Posture	The security, configuration and health state of the org as the product measures it — never the business data it contains.
Detect + advise	What the product does: it detects and advises, and never remediates. It changes none of your org's own configuration, no permission and no business data; what it does write is listed in the Installation Guide §4.
Plan	What the org is **entitled** to use: the free plan or a paid plan. Technically `FREE`, or `PRO` / `ENTERPRISE`, two historical names of the same paid level; default value **`FREE`**, hence *fail-closed*. Provisioned by the publisher, never by the subscriber. The first screen opening by an administrator also starts a 30-day trial of the full plan, once per installation (`OrgGuardian_Plan_Trial__c`, §2.2); an org with no paid plan returns to the free plan when the trial ends. The **Subscription** screen shows the plan in force, for example **OrgGuardian Discovery, free plan**.

Term	Definition
Feature switch	What actually **runs**, independently of the plan. Each switch has a default value shipped with the package, which your administrator can override in the **Feature activation** section (Administration ▸ Settings ▸ Feature activation; stored in `Feature_Gate__mdt`). **No outbound call is made by default**: the 22 outbound features — alert channels (5, reading Slack reactions included), exports and integrations (4, the automatic ITSM ticket creation included), external probes (7), self-callouts (6) — are all off at installation and are switched on one by one. 42 features are on by default, and none of them is an outbound feature. Among them: **Scan narrative**, the **Since the previous scan** card at the top of the **Dashboard** screen; the **in-org analyses**, which read the org and make no outbound call, each available only on the plan that includes it; and the **User access review** and the **Privilege matrix and consolidation plan** screen, both read-only and on the paid plan only. **Event Monitoring security events** (paid plan; it reads events only in an org that holds Event Monitoring) is off by default; an administrator turns it on in **Feature activation**.
Bootstrap	Placing the scheduled jobs and running the reference scan, **deferred** until an authorised user first opens a product screen (the Installation Guide §4.1). Until the product runs, a banner on its screens says where the bootstrap stands: **OrgGuardian is starting up**, **OrgGuardian is not bootstrapped yet**, **OrgGuardian bootstrap did not complete** or **Bootstrap state cannot be verified with your profile**.
Graceful degradation	A refused read does not cancel the scan: the scan skips the source and carries on. For the main setup sources, it **declares** the gap through a High-severity finding, "OrgGuardian could not read ...", of category `Errors`, whose deduplication key starts with `OrgGuardian:ReadFailure:`. Not every refused read produces that finding: the reads that degrade without it, such as the bootstrap state read from `CronTrigger`, are listed in the Troubleshooting Guide §4. At the start of each scan, OrgGuardian also checks the four read rights that the scan account needs (View Setup and Configuration, View Roles and Role Hierarchy, View All Users, View All Profiles) and names each missing one in a High-severity finding of category `Security`, whose deduplication key starts with `OrgGuardian:ScanRights:`. A source that does not exist in the org is skipped without a finding.
Query row budget	The platform's limit of 50,000 query rows per transaction, shared out between the reads that run in the same transaction and recomputed live against the actual limits, so that together they never cross it. When it is reached: the Troubleshooting Guide §8.
Deduplication key	`Finding__c.Dedupe_Key__c`: the stable identity of a finding between two runs.
Automatic resolution	A finding closes when the condition is no longer detected at the next run. There is **no** manual "Resolved" transition: this is a design decision by the publisher.
Regression	A finding that **came back** after its closure and is still open: `Finding__c.Regression_Count__c` above zero. Only the return of a closure **verified** by a scan increases that count, and the families whose closure is not a fix never count as regressions: product diagnostics, incident reconciliation notices, drift events, correlated incidents, attack paths, recounted guest sharing, deviations from the approved baseline and read notices (a read cut short or impossible, a detector that abstains, or a missing observation by the product itself). The posture digest says how many open findings are regressions ("resolved, then back"); the token `[REGRESSED]` opens the line of a regressed finding in the end-of-scan summary (its first ten lines: High and Critical findings, and Medium ones of Callouts and CodeQuality), and its immediate alert when it has one (a critical finding, escalation on the paid plan).

Term	Definition
Remediation reminder	The in-app bell sent to the owner (`Attestation__c.Remediation_Owner__c`) of a compliance gap whose due date has passed: one per owner and per run, at the night hour and at the frequency of the posture digest, at most once a day per control, and for at most 50 owners per run, in turn: those who can read the register, the least recently reminded first. **Remediation reminders** feature (`Remediation_Reminders` , paid plan, off by default), with the DORA / NIS2 pack. It names no one; the `REMEDIATION_REMINDER` lines of the system log count the owners left out, by reason.
Control	The unit of a pack score and of the org health score: a detector of the public detector catalogue, of severity Low or above, attached to the pack of its category (the controls of the Logins and Limits categories count in the health score only). All the findings of one detector form a single control. It **fails** as long as it has an open finding that no decision of your team covers, and, with no finding, it counts as **passed** only when its read was proven complete during the scan; otherwise it is **not evaluated** and does not count. A control your plan does not include is not evaluated, open finding included (the Configuration Guide, "How pack scores are computed" and "How the health score is computed").
Pack score	The 0 to 100 score of a pack (Security, Code quality, Callouts, Monitoring): the share of the controls evaluated by the last scan that passed, each weighted by its severity (Critical 10, High 5, Medium 2, Low 1). Controls passed by a decision of your team (acknowledged or snoozed, risk accepted, Ignored, active acceptance policy) are counted apart and cap the score at 99; "Not measured" when no control is evaluated, "Partial" when fewer than a quarter are.
Org health score	The 0 to 100 score of the org, the large figure of the Home page. The same rate as a pack score, over every control of the catalogue, those of the Logins and Limits categories included: the share of the controls evaluated by the last scan that passed, weighted by severity. Published when every pack your org is entitled to has published its score; otherwise it keeps its previous value, or reads "Not measured". Its word: At risk on an open Critical finding, "Partial" when the band would be green on a partial read, otherwise the word of the band (the Configuration Guide, "How the health score is computed").

See also: the Troubleshooting Guide (symptoms and degradations), the Security FAQ (posture, egress, retention), Permissions and Personas guide (personas and access).