



NxSAM How NxSAM Works — Quick Guide

Quick reference guide for the How NxSAM Works module

Product release	**NxSAM 2026.09**
Verified against code on	**2026-09-10**
Document revision	**b441b08-dirty**
Audience	**End users**
Document owner	**eSec Forte Technologies — Product Documentation**

Screenshots are taken from a demonstration data set. Figures are illustrative.

eSec Forte Technologies

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Table of Contents

How NxSAM Works

- Overview
- Find a specific vulnerability
- The two status fields, and why findings disappear
- Workflow: a vulnerability from discovery to closure
- How NxSAM Works: find the right screen

Reference

- Glossary

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

How NxSAM Works

How NxSAM Works

NxSAM collects security findings from many different tools, keeps them in one place, and tracks each one until it is fixed and verified. Everything else in this manual is a variation on that idea, so it is worth understanding once.

The four things NxSAM keeps track of

	What it is
Asset	Something that can have a weakness — a web application, a mobile app, an API, a server, a container image.
Audit	One assessment of one or more assets. A scan you imported, or a manual test your team performed.
Vulnerability	A single weakness found by an audit, on a specific asset.
Deviation	A formal, time-limited decision to accept a vulnerability rather than fix it now.

An audit belongs to an asset. A vulnerability belongs to an audit. That chain is why the same finding appears in several places in the interface — you are looking at the same record from different directions.

The life of a vulnerability

1. **Discovered.** A scan is imported, or a tester records a finding by hand.
2. **Approved.** Someone reviews the finding and accepts it into the audit. Until this happens, it is not counted in dashboards.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

3. **Assigned an SLA.** Based on its severity, the vulnerability is given a date by which it should be resolved. More severe findings get less time.
4. **Worked on.** The owner investigates and fixes it, moving it through statuses such as *Patch in progress* and *Patched*.
5. **Validated.** Someone confirms the fix actually worked. **This is the step that closes it.** A finding marked as patched but not yet validated is still open.
6. **Or reopened.** If a later scan finds the same weakness again, it comes back as *Reopen* and continues to age.

> **Patched is not closed**
>
> The most common misunderstanding in NxSAM. Marking something *Patched* records that work was done; marking it *Validated* records that the fix was verified. Only the second one closes the finding and stops the SLA clock.

Severity and SLA

Every vulnerability carries a severity — Critical, High, Medium, Low or Informational. Severity drives two things: how it is ranked for attention, and how long you have to fix it before it is overdue.

The exact number of days per severity is configured per organisation, so check your own SLA configuration rather than assuming. What is universal is the shape: the more severe the finding, the shorter the window.

A vulnerability is **overdue** when today's date has passed its SLA date and it has not been closed. This is worked out fresh each time you look, which is why something can become overdue overnight without anyone changing it.

Deviations: accepting risk on purpose

Sometimes a finding cannot be fixed in the time available — the fix needs a vendor release, or a change freeze is in effect. A **deviation** records that decision formally: who asked, who approved, why, and until when.

Deviations expire. When one does, the vulnerability returns to normal tracking. The point is that risk acceptance is a dated, reviewable decision rather than a finding quietly disappearing.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Who sees what

NxSAM shows people only the assets they are responsible for. An asset owner sees their own applications; a security analyst sees the whole estate. If a colleague cannot see something you can, that is access control working, not a fault — the fix is a permission change, not a data change.

Where to go next

- To find your way around the interface, see Navigating NxSAM.
- Each security module — application, API, network, container, cloud — follows the same audit-and-vulnerability pattern described here, with its own specifics.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Find a specific vulnerability

Three ways, depending on what you already know.

If you know the asset

The quickest route when you are asking "what is wrong with *this* system".

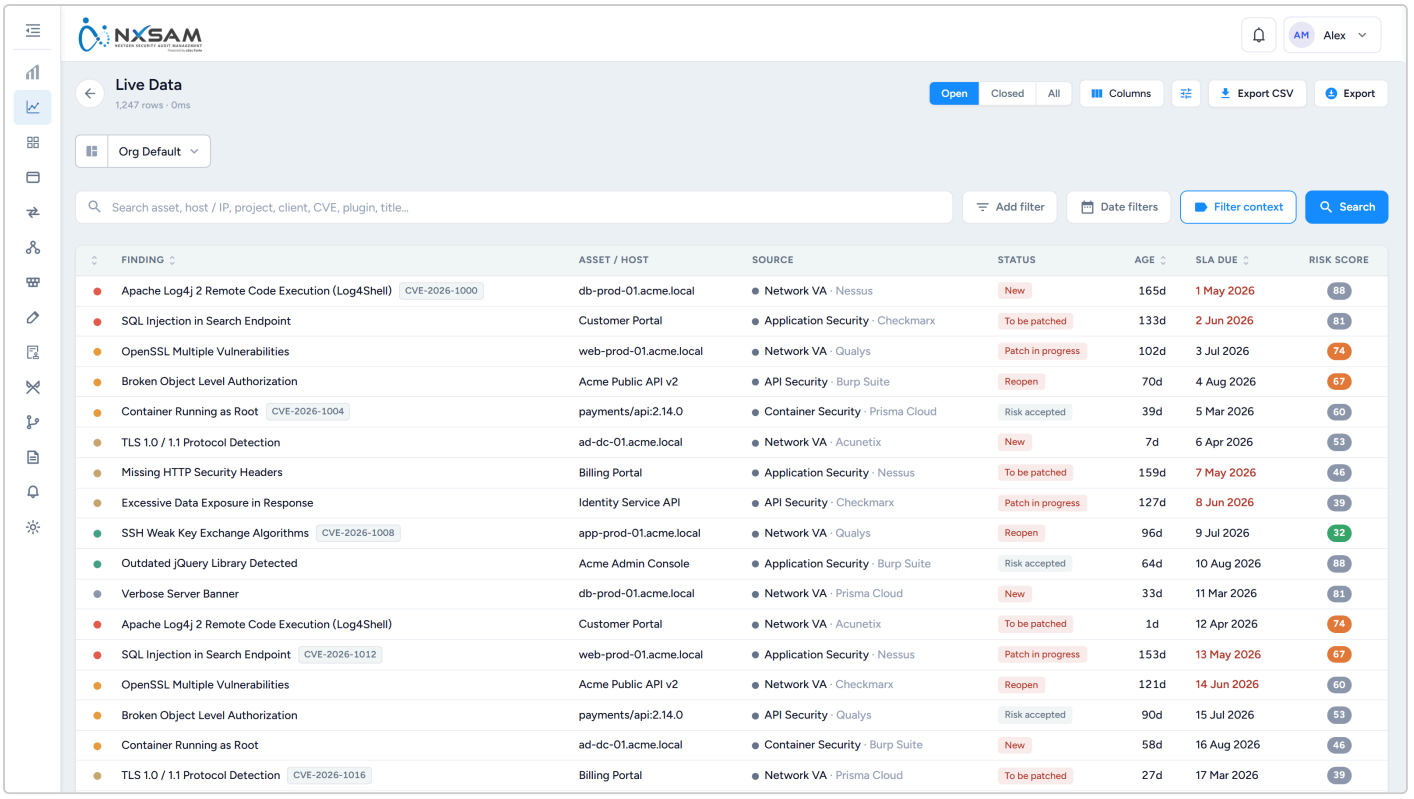
1. Go to the module the asset belongs to — Application Security, Network Security, and so on.
2. Open the asset, then its audits.
3. Open the audit to list its findings.

If you know the vulnerability

When you are asking "where do we have *this* problem" — after a CVE is announced, for instance.

1. Open **Analytics Dashboard** → **Live Data**.
2. Filter by the vulnerability name or identifier.
3. The result lists every asset affected.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.



Analytics Dashboard → Live Data — findings from every module in one list

This is the view to use for "are we exposed to X", because it crosses all modules at once rather than making you check each in turn.

If you only know roughly what you are after

Use the filters on any module's vulnerability list. They combine, so you can narrow by several at once:

- **Severity** — Critical through Informational
- **Status** — open, in progress, closed
- **SLA state** — overdue, approaching
- **Asset**, **owner**, **project**, **business unit**

If the list looks empty, check the filters first

Filters persist. A filter set earlier — especially a pinned one on the Analytics Dashboard — remains applied, and an unexpectedly empty list is much more often a filter than an absence of data.

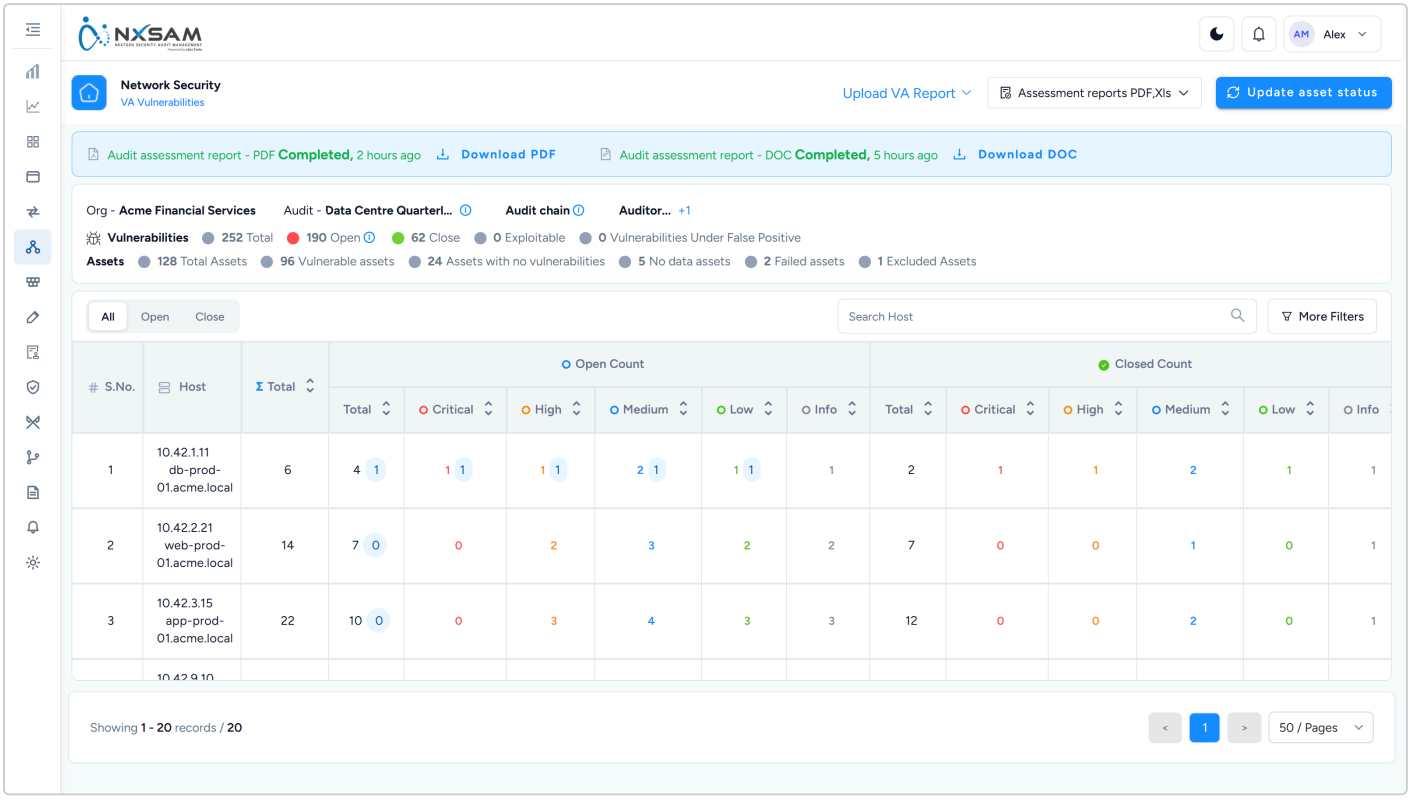
Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Reading the result

Each finding shows:

Severity	How serious the weakness is
Status	Where it is in its lifecycle
SLA date	When it should be resolved by
Discovered	When it was first seen — not when it was imported
Asset	What it affects

Open the finding for the full description, evidence, and history of what has happened to it.



Network Security → VA Audits → the audit's findings — one module's list, and the shape they all share

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Discovered is not imported

A finding your scanner first saw months ago arrives already that old, and may already be past its SLA on the day it appears. That is correct — the weakness has genuinely existed that long.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

The two status fields, and why findings disappear

More tickets come from this one design point than from anything else in the product, because two different fields are both called "status" in conversation and they answer different questions.

They are not the same field

Every finding carries two independent statuses:

	Question it answers	Owned by
Validation status	Is this finding genuine and agreed?	The security team
Client status	What is being done about it?	The asset owner

A finding has both at all times. Neither is derived from the other, and changing one does not change the other.

Validation status moves through `Saved` → `Submitted` → `Approved`, with `Closed` and `False Positive` as terminal outcomes. Those five are the whole set. A finding is not visible outside its audit until it is approved.

Client status covers remediation and has fourteen values: `New`, `To be patched`, `Patch in progress`, `Patched`, `Not Patched`, `On hold`, `Reopen`, `Validated`, `Not Applicable`, `Risk accepted`, `Risk acceptance`, `Duplicate`, `Exception Handled`, `Expired`.

Only `Validated` and `Exception Handled` are flagged as closed in the product. Everything else — `Patched` included — is still counted as an open finding, which surprises people who have just marked a batch as patched and seen no movement in the totals.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

The consequence people hit

> ⚠ **A finding that exists is not necessarily counted**
>
> Dashboards, SLA clocks and posture counts consider **approved** findings. A finding that has been imported and is visible inside its audit but is still `Submitted` will not appear in any of them.
>
> So "the dashboard is missing findings I can see in the audit" is almost always approval status — not a data loss, a caching problem, or a filter.

This is deliberate. It stops an unreviewed scanner export from moving the numbers a board sees. But it is invisible from the dashboard end, which is why it generates tickets.

The three questions that resolve most of these

1. **Is the finding approved?** If not, it is working as designed. The remedy is review, not a bug report.
2. **Is the audit considered?** An audit can be excluded from counting entirely. Its findings then exist, are approved, and still do not appear in aggregate figures.
3. **Is the client status one that counts as open?** `Risk Accepted` and `Not Applicable` findings are not closed — they are open findings deliberately not being fixed, and different screens treat them differently.

Work through them in that order. The first two account for most cases.

Re-opening, and what it does to the SLA

A finding closed and then seen again by a later scan is re-opened rather than created fresh. It keeps its identity and its history, which is what you want for audit purposes.

The corner case worth knowing: **re-opening sets a new discovery date but does not always reset the SLA expiry**. A finding can therefore show as discovered today and overdue at the same time — an apparent contradiction that is really two fields updated on different schedules. Treat the discovery date as "last seen", not "first seen", and read the SLA from the expiry date rather than inferring it from the age.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Why a count differs between two screens

When a module screen and the analytics dashboard disagree, it is one of four things, in descending order of likelihood:

1. **Approval.** The module screen shows unapproved findings; aggregate views do not.
2. **Considered.** An excluded audit is visible in its module and absent from totals.
3. **Timing.** Analytics is served from a synchronised copy of the data. A gap between the two is a sync interval, not a discrepancy — the Warehouse Sync screen shows when the copy was last refreshed.
4. **Scope.** The two screens are filtered differently. Asset-owner accounts see only their own assets, so the same screen shows different totals to different people, both correct.

Checking them in that order is faster than comparing row by row.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Workflow: a vulnerability from discovery to closure

The end-to-end journey, with the decision points where things actually go wrong. This cuts across every security module — the specifics differ, the shape does not.

Stage 1 — Discovery

A finding enters NxSAM by scanner integration, by report upload, or by hand.

What is decided here: its **identity**. The platform must determine whether this is something it has seen before or something new. For network VA this is the vulnerability key; for application findings it is the tool's identifier plus asset context.

Where it goes wrong:

Symptom	Cause
Counts double after an import	Identity changed, so nothing matched the previous data
Findings reopen unexpectedly	Same cause, seen from the other side
Two distinct findings merged into one	Key parts coerced to the same value (blank port becomes `0`)
Findings attached to the wrong asset	Asset identifier in the file differs from the one held

Detail in the integrations reference.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Stage 2 — Approval

A discovered finding is not yet counted. Someone reviews it and approves it into the audit.

Why it exists: scanners report things that are not real. Approval is where a human accepts the finding as genuine before it starts consuming anyone's attention.

The support consequence: unapproved findings are visible inside the audit but absent from dashboards and posture counts. "The dashboard is missing findings I can see in the audit" is this stage, essentially every time.

At import, some findings may be auto-classified as false positives and never enter the open population at all. That is configuration, not data loss.

Stage 3 — SLA assignment

On approval the finding is given an SLA date derived from its severity.

What to know:

- The date is **stored**; the overdue *state* is **computed live** against today's date.
- Changing SLA configuration does not re-date existing findings.
- Shortening an SLA can push a large number of findings overdue immediately, because the comparison is against today.
- A finding with no SLA date is never overdue — and never compliant either. It is simply outside SLA reporting.

Stage 4 — Ownership and remediation

The finding becomes visible to whoever owns the asset, and moves through working statuses: *New*, *Patch in progress*, *Patched*.

The critical misunderstanding lives here. *Patched* does **not** close a finding. It records that work was done. The finding remains open, keeps ageing, and keeps counting.

Where it goes wrong:

Symptom	Cause
"We fixed it but it still shows open"	Marked *Patched*, not *Validated*

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Symptom	Cause
"The owner cannot see it"	Asset ownership, not role
"It is assigned to someone who left"	Ownership follows the asset, not the person

Stage 5 — Validation and closure

Someone confirms the fix actually worked and sets the finding to **Validated**. Only this — or *Exception Handled* — stops the clock.

Statuses that look like closure and are **not**: *Risk accepted*, *Not Applicable*, *Duplicate*, *Expired*. All four continue to count as open. See the counting rules.

Stage 6 — Reopening

A validated finding seen again by a later scan returns as **Reopen**.

What it means: the weakness is present again. Usually a rolled-back patch, a host rebuilt from an old image, or configuration drift.

What to check when it looks wrong: whether identity changed. If the vulnerability key was reconfigured, findings can appear to reopen *en masse* when nothing on the systems changed at all. A sudden reopen spike with no corresponding infrastructure event is an identity question, not a security one.

The alternative path: accepting the risk

At any point after approval, a **deviation** may be raised to accept the risk for a fixed period. It is a dated decision with an approver, not a way of removing the finding.

The finding stays open throughout. When the deviation expires the finding continues exactly as before — it was never closed. See the deviations reference.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Diagnostic summary

Given a symptom, the stage to examine first:

Reported symptom	Stage
Counts jumped or doubled	1 — identity
Findings visible in audit, missing from dashboard	2 — approval
Overdue figures changed with no data edit	3 — SLA computed live
"We fixed it but it is still open"	4/5 — Patched is not Validated
Sudden mass reopen	6, then back to 1 — identity changed
Accepted risk still counting as open	Alternative path — by design

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

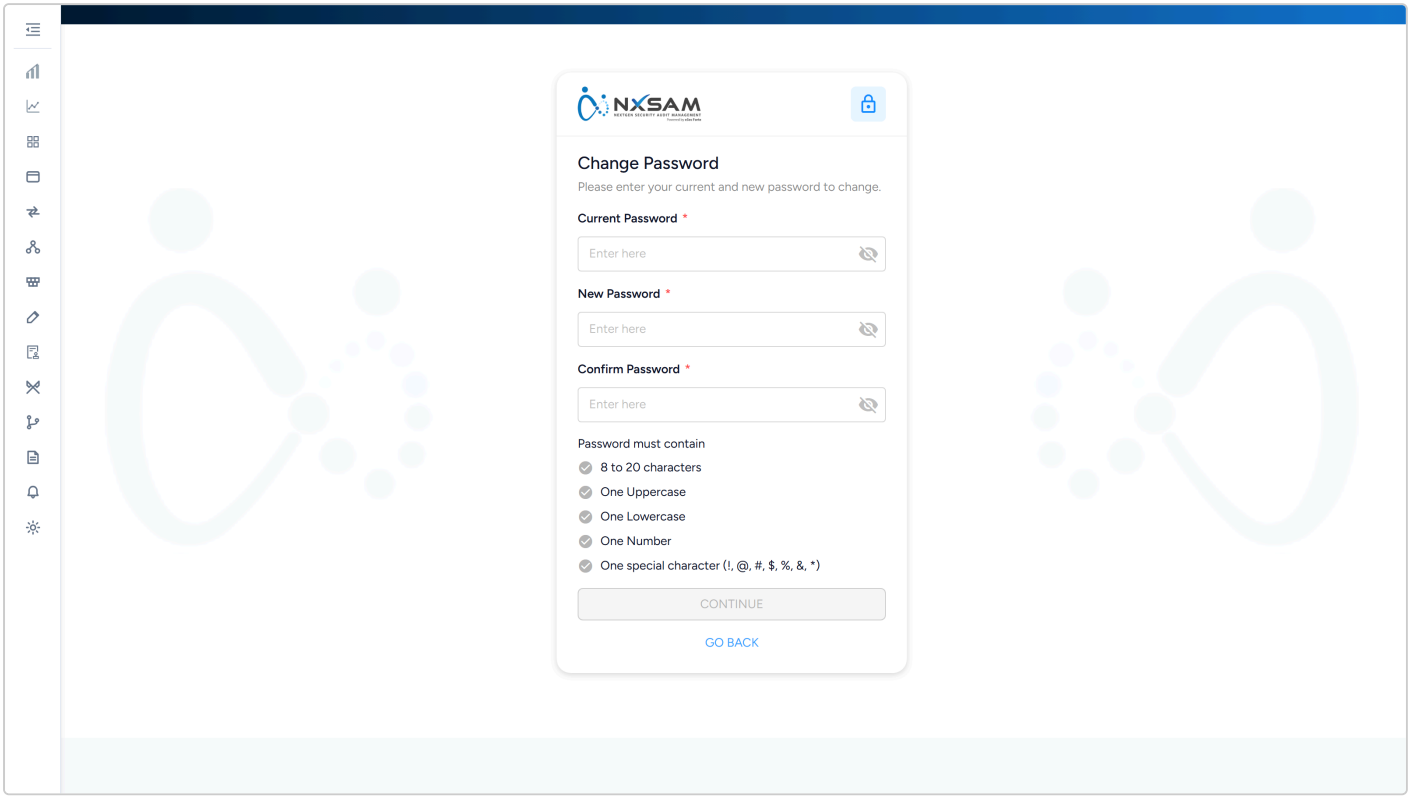
How NxSAM Works: find the right screen

Every screen in this module, what each one is for, and where it lives. Use it to find the right screen quickly, and to confirm you are on the one a colleague means — several screens in this product have similar names and different jobs.

Screenshots come from a demonstration data set, so the figures are illustrative. Click any screenshot to open it at full resolution.

Change Password

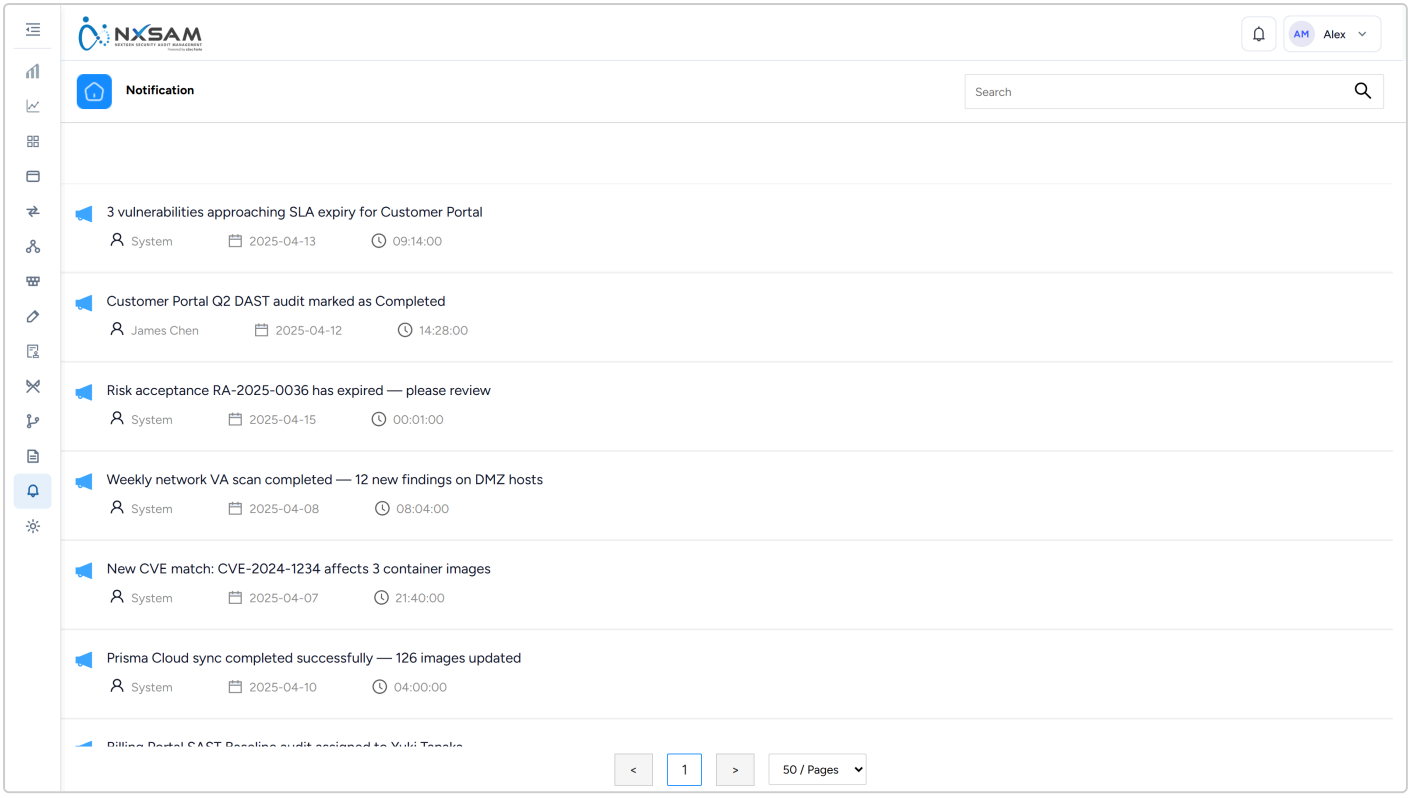
Where a signed-in user changes their own password.



Change Password

Notifications

In-platform alerts — SLA warnings, completed audits, finished imports.



Notification Feed → Notifications

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Reference

Glossary

Terms as NxSAM uses them. Several are ordinary security words with a narrower meaning here, and those are the ones worth reading even if the term is familiar.

Assets and scope

Asset — anything that can hold a vulnerability: a web or mobile application, an API, a network host, a thick client, or a container image. Each type is a separate inventory with its own columns and its own licensed limit.

Asset property — an attribute recorded against an asset, such as criticality or whether it is internet-facing. Properties are more than description: they narrow SLA durations and feed the contextual risk score.

Bifurcation — splitting one application into separately assessed parts, so a large system can be tested module by module rather than as one unit.

Collection — a grouping of container images that comes from the scanning platform rather than from NxSAM, and is not editable here.

Scope — what an assessment covers. For applications this is the application; for compliance it is a host list, which is why compliance scope disputes are settled with an exported list of addresses.

Assessments

Audit — one assessment of one asset. It has a type, a category, a cycle, a scope, and a status of its own, separate from any finding's status.

Audit cycle — which round of a recurring programme an audit belongs to.

Confirmatory audit — a re-test created against a parent audit to verify that fixes hold. It is evidence about the original rather than an independent assessment, so the two should not be counted as two rounds of testing.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Considered — the marker deciding whether an audit contributes to aggregate figures. Normally the latest audit per asset that reached a final state. An audit that is *not* considered still shows its findings on the module screen while contributing to no dashboard or report.

Initial audit — a first assessment, as opposed to a confirmatory one.

Findings

Vulnerability (also *finding*, *issue*) — one weakness on one asset, found by one assessment.

Vulnerability master — the catalogue entry (title, description, remediation) inherited by every occurrence of that vulnerability everywhere. Editing the master changes all of them.

Posture — Policy Compliance's unit: one control checked against one host. Every check produces a posture whether it passed or failed, which is why this module's volumes dwarf the others.

Artifact — the evidence attached to a finding: the steps, description and images that let someone else reproduce it.

Instance — which occurrence, where the same control or finding applies more than once to the same asset.

Status

Validation status — whether a finding has been confirmed. Nothing is counted anywhere until this reaches *Approved*, which is also when **Discovered On** is populated.

Client status (also *current status*) — what is being done about the finding. Of its values, only two close it.

Closed type — the flag deciding whether a status stops something counting as open. It is configuration rather than a fixed rule, so a deployment can change what "closed" means.

Posture status — what a compliance check returned: pass, fail or **error**. Error means the check could not run — a scanner or credentials problem rather than a compliance result.

Sub status — the qualifier beneath a main status: which kind of acceptance, or which stage of an approval.

Time and SLA

SLA duration — how long a finding of a given severity on a given asset type may remain open. Configurable, and narrowable by asset property.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

SLA expiry — the resulting deadline. Changing an SLA duration re-derives expiry dates, which can make a batch of findings overdue with no data change at all.

Discovered On — when a finding was first observed. Blank until the finding is approved.

Last observed — the most recent sighting, which is how a finding is known still to exist.

Overdue — past SLA expiry and not closed.

Decisions

Deviation (also *risk acceptance*, *RA*) — a recorded decision not to fix something, with an approval chain and an expiry. It does **not** close the finding: the exposure is still live and still counts as open.

Exception — a status marking an agreed exception rather than a resolution. Reporting that conflates the two overstates remediation.

Auto reject end date — when a temporary acceptance lapses on its own.

Integrations and processing

Integration — a connection to a scanning tool, either pulling from its API or parsing an uploaded report.

Vulnerability key — the columns deciding whether an incoming finding is one already seen. Changing it redefines "the same finding" and splits history.

Queue — asynchronous work: imports, syncs, report generation. A screen that seems not to have updated usually has a queue entry that explains it.

Data process — how far a tool-fed audit has got through the import pipeline; the difference between "the upload finished" and "the data is ready".

Beat — a scheduled job maintaining derived data, such as recalculating SLA states as dates pass. A stopped beat produces figures wrong by a fixed interval and raises no error.

Governance

RLCM — Red Line Compliance Management: assessment of controls against frameworks, rather than of systems for weaknesses.

TPRM — Third-Party Risk Management: the same control assessment applied to suppliers. It shares screens with RLCM and differs by the audit's type.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

UCF — Unified Control Framework: the mapping that lets one control satisfy several frameworks at once, so answering it once covers each of them.

Control — a requirement being assessed, as opposed to a weakness being found.

Standard — the framework a control belongs to, such as ISO/IEC 27001 or PCI DSS.

Reporting

Considered audit filter — the switch on Report Search that reproduces either the dashboard's numbers or the module screens'. Comparing the two settles most disagreements about counts.

Design template — the layout a report renders through, separate from the template that decides its content.

Share view against share copy — sharing a template as a view keeps one definition that your edits reach; sharing a copy creates an independent template that will drift.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.