



NxSAM Integrations — Quick Guide

Quick reference guide for the Integrations 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

Integrations

- Overview
- Connect a scanner, end to end
- How an import actually runs
- Setting up each integration
- Integrations: every screen, column and action
- Integrations: find the right screen

Reference

- Glossary

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Integrations

Integrations

NxSAM does not scan anything itself. It connects to the tools that do, collects what they find, and gives you one place to track all of it. This page explains what that connection involves and what to expect once it is running.

What an integration does

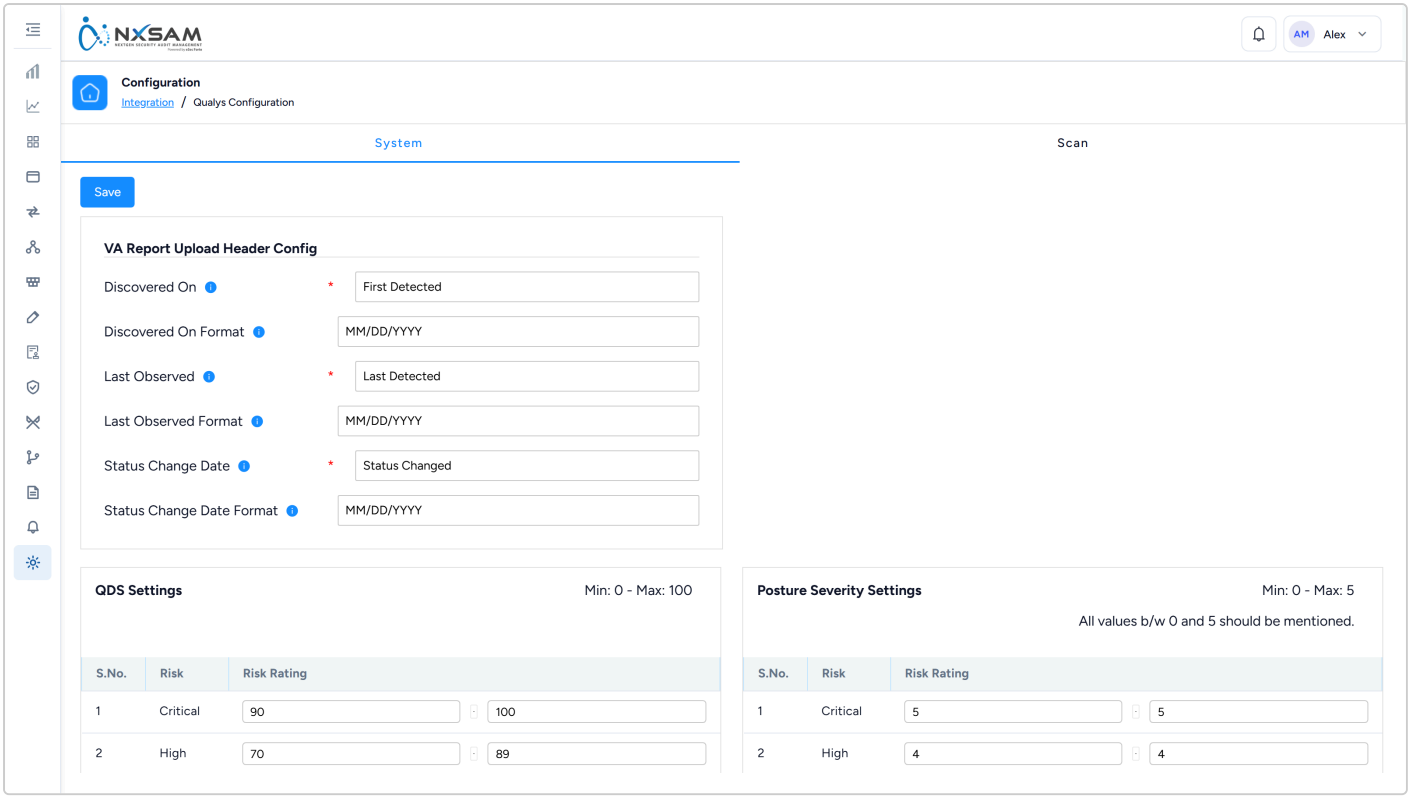
Once configured, an integration:

1. **Connects** to the tool using credentials you supply.
2. **Collects** findings from the scans that tool has run.
3. **Matches** each finding against what NxSAM already knows, so an issue found again is the same issue rather than a new one.
4. **Files** it against the right asset, in the right module.

From then on the finding behaves like any other in NxSAM: it has a severity, an SLA, an owner, and a lifecycle.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Which tools connect where



A scanner integration. Column mapping is what makes an import land in the right fields.

Area	Tools
Network vulnerability assessment	Qualys · Tenable.io · Tenable.sc · Nessus · Nexpose · Cortex XDR
Application security	Checkmarx · Checkmarx SCA · Burp Suite · Netsparker · Acunetix · Fortify · AppScan · Snyk Code · Gitleaks · Dependency Track · JFrog Xray
Container security	Prisma Cloud · Twistlock
Cloud posture	Prisma Cloud CSPM · Cortex CSPM · SentinelOne CSPM

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Findings from a network scanner appear under Network Security; findings from an application scanner appear under Application Security, and so on. The module a finding lands in is determined by the tool it came from, not by a choice you make at import time.

What you need before you start

Three things, in this order:

1. **Somewhere to reach.** The tool's API endpoint, reachable from the NxSAM server. If NxSAM sits behind a proxy, that is configured per integration.
2. **An account.** A service account with API access — not a personal login. Personal accounts break when the person leaves or changes their password.
3. **Scope.** The account must be able to see the assets you expect to import. This is the step most often missed, and it fails quietly.

> ⚠ **Valid credentials are not the same as useful access**
>
> An account with correct credentials but no visibility of your target scope connects successfully and imports nothing. The screen shows a working connection and an empty result, which looks identical to "there is nothing to report".
>
> If an integration connects but never brings anything in, check what the service account can see in the source tool before changing anything in NxSAM.

Setting one up

Every integration follows the same pattern:

1. Go to **Configurations** → **Integration** and choose the tool.
2. Enter the connection details — endpoint, port, credentials.
3. Select which client the imported data belongs to.
4. Save, then run a sync.

The exact fields differ per tool, but they are always some combination of *where*, *who*, and *for whom*.

After the first sync

Expect these, and do not treat them as faults:

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

- **The first sync imports history.** Your first import can bring in a large number of findings at once, including old ones. Subsequent syncs are much smaller.
- **Ages reflect discovery, not import.** A finding your scanner first saw six months ago arrives already six months old, and may already be past its SLA. This is correct — the weakness has genuinely existed that long.
- **Not every finding becomes an open item.** Some tools are configured to mark certain findings as false positives automatically. Your support team can explain the rules configured for your deployment.

When something looks wrong

What you see	Usually means
Connection works, nothing imported	The service account cannot see the target scope
Far fewer findings than the tool reports	Automatic classification rules are filtering them
Counts jumped sharply after a sync	How findings are matched has changed — raise with support
Findings you closed have come back	The scanner still sees them; closure needs re-validation

For the rules behind these, your support team has the detailed reference.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Connect a scanner, end to end

Connecting a tool is four decisions and one test. Getting the decisions right first saves the reconciliation afterwards.

Before you start

You need	From whom
The endpoint URL for **your** region	The tool's administrator
An account with **API access** enabled	The tool's administrator
Confirmation of what that account can *see*	The tool's administrator
Network reachability, including any allow-list	Your network team

> ⚠ **Ask what the API account can see, not what your account can see**
>
> This is the failure that wastes the most time. The connection succeeds, the account signs in, and it can see none of the scans — because scope in the scanning tool is granted per account and yours is not the one NxSAM uses.

1. Choose the right integration

Several vendors appear more than once, and the entries are not interchangeable:

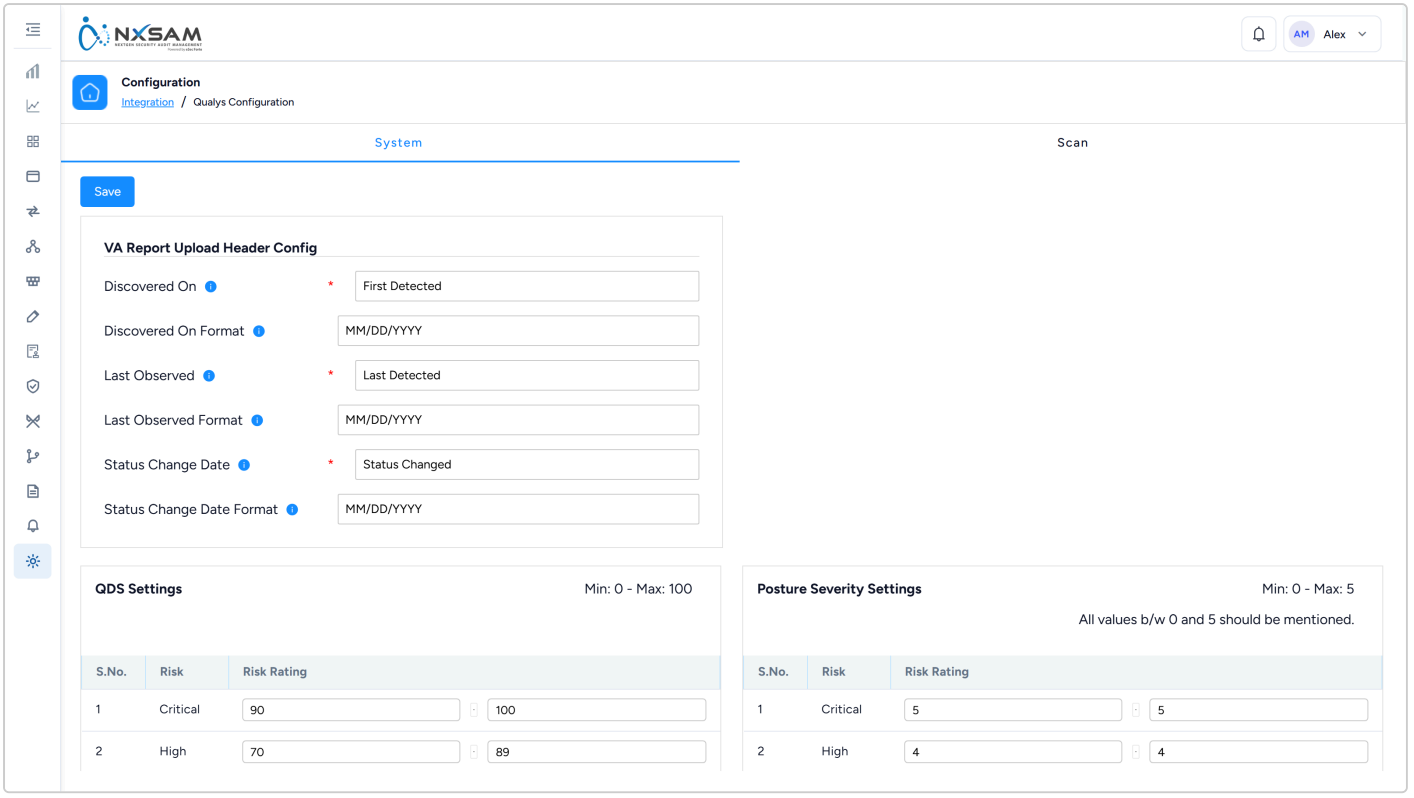
- **Checkmarx**, **Checkmarx SCA** and **Checkmarx Standalone SCA** are three integrations. SCA covers third-party libraries rather than your code.
- **Prisma Cloud** appears twice — once for container security, once for cloud posture — with separate credentials.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Connecting the wrong one produces an integration that works and populates nothing.

2. Enter the connection

Configurations → **Integration**, then the tool.



Configurations → Integration → Qualys Configuration

Supply the URL, port, credentials and the client the imported data belongs to.

3. Map the columns

Report-based and several API integrations map the tool's columns onto NxSAM's fields. Two groups matter.

Dates — each date names both the column and the format it is written in:

- Discovered On, with its format
- Last Observed, with its format

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

- Status Change Date, with its format

> ⚠ **A wrong date format is worse than a missing one**
>
> `03/04/2026` is a valid date in two formats five weeks apart. A missing date is rejected and reported. A misread one is imported and silently wrong, and it shifts every SLA calculation built on it.
>
> Check the format against a current export from the tool, not against its documentation — vendors change export defaults between versions.

Identity — which column decides that an incoming finding is one NxSAM has seen before.

> ⚠ **Changing identity later splits your history**
>
> Change it after data exists and the next import will not match the previous data: existing findings stop being observed while fresh copies arrive with no history, and counts appear to double.
>
> It is not undone by changing the setting back. Decide it before the first sync.

---

4. Test the connection

Use **Test Connection** before saving. Several integrations will not save an enabled connection that has not been tested — deliberately, because a typo in the URL otherwise surfaces days later as an empty sync.

---

5. After the first sync

The first collection takes substantially longer than later ones; it has history to fetch. Then check three things:

1. **Did unexpected assets appear?** Some integrations create assets for scan targets NxSAM did not know about.
2. **Did audits appear that nobody created?** Several integrations create an audit per scan. That is normal — the control is the integration's scope, not deleting audits.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

3. **Do the counts roughly match the tool?** Small differences are deduplication; large ones are scope.

6. Confirm coverage, not just import

A clean import says nothing about what was *missed*. For Tenable, the **IO** tab compares the two asset lists directly:

- **Tenable IO Only** — hosts being scanned that NxSAM has no asset for, so their findings have nowhere to land.
- **NxSAM Only** — assets you believe are being assessed and are not. This is a coverage gap, and it is the number worth acting on.

When something looks wrong

Symptom	Usual cause
Authenticates, imports nothing	API account scope
Fails like a wrong password	Wrong regional endpoint
Dates are weeks out	Date format mapping
Counts doubled after an import	Identity configuration changed
Fewer findings than the tool shows	Import classification rules (Checkmarx-style)
Module still empty after connecting	The other integration of the same vendor

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

How an import actually runs

This is the path a finding takes between a scanner and a screen, and the places it can stop. Most "the import did not work" tickets are the import working correctly at a stage the reporter cannot see.

Nothing happens synchronously

Submitting an import does not process it. It writes a queue entry and returns. A worker picks the entry up afterwards and does the work.

That is why the upload succeeds instantly and the data appears minutes later, and why "the upload worked but nothing changed" is the normal appearance of a queued job, not evidence of a failure.

```
upload → staged → queued → worker consumes → matched → filed → visible
```

The five stages, and what failure looks like at each

Stage	What happens	If it stops here
Upload	The file is accepted and stored	An immediate error on screen — the only stage that reports failure to the person who triggered it
Staging	Rows are parsed into staging	Rows are set aside as failed with a per-row reason; the job still reports as processed
Queue	A message is published to RabbitMQ	Status stays `PENDING` indefinitely — the queue took it, nothing is consuming
Worker	A consumer processes the message	Status goes to `FAILED`, or sticks at `IN PROGRESS` if the worker died mid-message
Merge	Rows are matched and written to live tables	Data is present but attached to the wrong asset, or duplicated

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Queue status is one of `PENDING`, `IN PROGRESS`, `COMPLETED`, `FAILED`. The tool-sync queues additionally use a **partially completed** state, which means exactly what it says: some hosts or assets in the batch succeeded and others did not, and the job will not be retried automatically.

> ⚠ **`COMPLETED` does not mean every row was imported**
>
> A job completes when the worker finishes processing the message. Rows rejected during staging had already been set aside as failed before the worker ran. A completed import with fewer findings than the source file is the expected outcome of a partly invalid file — not a contradiction.
>
> Check the failed-row count before concluding the import lost data.

Where to look, in order

1. **Process Manager** — is the job `PENDING`, `IN PROGRESS`, `FAILED` or `COMPLETED`? This answers "is it stuck or is it working" in one look, and it is the first thing to check on any import ticket.
2. **The import queue log** — per-job history, including how long it took. A job that normally takes two minutes and has been running for forty has not hung by design.
3. **Failed rows** — the per-row rejection reasons. Almost always a column mapping problem or a date format the parser did not accept.
4. **The module screen** — with the audit filter cleared. Findings do arrive but land against a different audit than expected more often than they go missing.

Nothing is moving at all

If every job across every module is `PENDING`, the problem is not the import. Either the message broker is not reachable or no workers are running, and the same symptom will show on scheduled reports and outbound email, because they use the same queues.

Confirm by checking whether a report queued now also stays pending. If it does, this is an infrastructure question, not a data one.

Matching: why a re-import does not double everything

Each finding is matched against what NxSAM already holds before it is written. A finding seen before updates the existing record and keeps its history, its SLA clock and its ownership; only genuinely new findings are created.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Two consequences worth knowing:

- **Re-importing the same file is safe.** It will not duplicate findings. It will, however, update `last seen` dates, which shifts what "recently detected" shows.
- **If matching changes, history splits.** Change the fields that identify a vulnerability and the next import will not recognise existing findings as the same ones. They will be created fresh — the old ones stay open with no new sightings, the new ones arrive with no history. This is the single most disruptive configuration change available in the product, and it is not reversible by changing the setting back.

Timing and the "missing" findings

The asset must exist before the finding can attach to it. An import that references an asset NxSAM does not have will either create it or reject the row, depending on the module. Where assets come from a synchronised source, an import that runs before the asset sync completes can reject rows that would have succeeded twenty minutes later.

If an import failed once and succeeded on retry with no change made in between, this is almost always the reason.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Setting up each integration

What each connection needs, what it contributes, and the thing about it that most often goes wrong. Nineteen integrations share four patterns — read the pattern first, then the tool.

The four patterns

Pattern	Tools	What NxSAM does
API pull	Qualys, Tenable, Nexpose, Prisma, Cortex XDR, CSPM	Connects on a schedule and collects completed scans
API pull with project mapping	Checkmarx, Fortify, AppScan, Snyk, Gitleaks, JFrog, Dependency Track	As above, plus a decision about which project maps to which asset
Report upload	Burp Suite, Netsparker, Acunetix	You export from the tool; NxSAM parses the file
Posture pull	Prisma CSPM, SentinelOne CSPM, Twistlock	Cloud or container posture rather than vulnerabilities

The pattern decides where a failure can occur. An API pull can fail on credentials, authorisation or reachability. A report upload cannot — it fails on parsing instead.

What every API integration needs

1. **An endpoint URL.** Region-specific for several vendors. A wrong region fails authentication in a way that looks exactly like a wrong password.
2. **An account with API access enabled.** Separate from console access in most tools. A security team will often issue a console login and assume it works.
3. **Scope granted to that account.** This is the one that bites: the connection succeeds, the account signs in, and it can see none of the scans. Ask the tool's administrator what the *API account* can see, not what your own account can see.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

4. **Network reachability**, including any IP allow-list on the vendor side.

Column mapping: what it is and why it matters

Report-based and several API integrations map the tool's columns onto NxSAM's fields. Two groups of settings do this.

Date fields. For Qualys and Tenable, the mapping names the column in the uploaded report that holds each date, *and* the format that column is written in:

- **Discovered On**, with its **Discovered On Format**
- **Last Observed**, with its **Last Observed Format**
- **Status Change Date**, with its **Status Change Date Format**

Tenable repeats the same three pairs for Policy Compliance, and adds a **Key Header** naming the column that identifies a vulnerability — the one used for matching when **Enable Key for VA Vulnerability** is on.

> ⚠ **A wrong date format is worse than a missing one**
>
> `03/04/2026` is a valid date in two formats five weeks apart. A missing date is rejected and reported; a misread one is imported and silently wrong, and it shifts every SLA calculation built on it.
>
> Check the format against a current export, not against the documentation — vendors change export defaults between versions.

The vulnerability key. This names the column NxSAM matches on to decide whether a finding is one it has seen before. Changing it after data exists splits history: existing findings are no longer recognised, so they stay open with no new sightings while fresh copies arrive with no history. It is not reversible by changing the setting back.

Checkmarx: the approval rules

Checkmarx integrations carry an extra step no other tool has. Checkmarx records a triage state per result, and the integration translates that into NxSAM's own states through three configurable rules:

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Rule	Decides
Approval state	Which Checkmarx value means the finding is confirmed
Non-approval state	Which value means it is not yet confirmed
False positive state	Which value means it was dismissed

Each rule is a comparison against a named column of the uploaded report.

This is why the same Checkmarx scan can produce very different NxSAM results at two customers: the rules encode how *that* organisation uses Checkmarx triage. If findings arrive already approved when they should be pending, or false positives arrive as open findings, these three rules are the cause — not the import.

Per-tool notes

Network vulnerability assessment

- **Qualys** — the widest column mapping. Also supports QDS-to-severity mapping, so severity can follow the Qualys Detection Score rather than CVSS.
- **Tenable** — one integration covers Tenable.io, Tenable.sc and Nessus, on three tabs: **System**, **SC** and **IO**. It carries two header sets, one for VA and one for Policy Compliance.
- **Nexpose** — API pull; findings land in Network VA alongside the others.
- **Cortex XDR** — a newer source, using endpoint host insight data as VA findings. Ships unconfigured; the screen shows a prompt until credentials are entered.

Application security

- **Checkmarx / Checkmarx SCA / Checkmarx Standalone SCA** — three separate integrations. SCA covers third-party libraries, not your code, and standalone SCA runs independently of the main platform. Connecting the wrong one is a common first-time error.
- **Fortify, AppScan, Snyk Code** — static analysis. Project-to-application mapping is the setup step that matters.
- **Burp Suite, Netsparker, Acunetix** — dynamic. Report upload rather than API pull, so the failure mode is parsing rather than authentication.
- **Gitleaks** — secrets in source control. Findings are credentials, so treat the finding detail itself as sensitive.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

- **JFrog Xray, Dependency Track** — composition analysis. Dependency Track works from an SBOM rather than from a scan, so it reports on what you declare rather than what it finds.

Container and cloud

- **Prisma Cloud** — appears twice, as container security and as CSPM. They are separate integrations with separate credentials despite the shared vendor.
- **Twistlock** — the earlier name for the Prisma container product; present for deployments still running it.
- **Cortex CSPM** — Palo Alto's posture platform, feeding the **CSPM** screens (Assets, Issues, Cases, Rules, Policies) rather than the Prisma ones. The connection takes the tenant's **FQDN**, an **API Key ID** and the **API Key**, with an optional proxy; **Test Connection** checks them before saving. It then has two sync actions, and the order matters: **Master Sync** first, which brings the rules and policies, then **Sync**, which brings assets, issues and cases against them. A sync run before the master sync has nothing to attach issues to.
- **SentinelOne CSPM** — an alternative posture source, also feeding the CSPM screens. One **Data Sync** action brings everything.

> **Prisma CSPM and Cortex/SentinelOne CSPM are different screens**
>
> Prisma feeds the four *Prisma CSPM* screens; Cortex and SentinelOne feed the five *CSPM* screens. Connecting one does not populate the other, and an empty CSPM module after a successful Prisma sync is expected rather than a fault.

> **Three integration screens log a template error on load**
>
> Checkmarx, Checkmarx SCA and Fortify each report an unknown-property binding in the browser console. The tabs work and the integration functions. It is a known product defect, not a sign of a broken connection.

After the first sync

Expect the first collection to take substantially longer than later ones — it has history to fetch. Check three things once it completes:

1. **Did assets get created that you did not expect?** Some integrations create assets for scan targets NxSAM does not know about.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

2. **Did audits appear that nobody created?** Several integrations create an audit per scan. That is normal; scoping what the integration collects is the control.
3. **Do the counts roughly match the tool?** Small differences are deduplication. Large ones are scope — the API account seeing more or less than you assumed.

Reconciling the asset list with Tenable

The Tenable **IO** tab compares its asset list against NxSAM's and reports the overlap, which answers question 3 directly rather than by inference. **Generate** produces a summary; **Refresh** re-reads the latest one.

Column	What it means
Common IPs (Both)	Hosts both systems know about — the healthy case
Tenable IO Only	Hosts Tenable scans that NxSAM has no asset for
Scanned by Tenable	Of the common hosts, how many Tenable has actually scanned
NxSAM Only	Assets in NxSAM that Tenable is not scanning
Status, **Initiated at**	The comparison job itself

The two "only" columns are the ones to act on, and they mean opposite things:

- **Tenable IO Only** is data you collect but do not manage — findings on those hosts have nowhere to land.
- **NxSAM Only** is a **coverage gap**: assets you believe are being assessed and are not. A clean NxSAM with a large NxSAM-Only count is not a healthy estate, it is an unscanned one.

Run this before trusting a first sync

A sync that imports cleanly still tells you nothing about what it *missed*. This comparison is the only place in the product that shows the difference between the two inventories, and it is quicker than reconciling two exported spreadsheets.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Integrations: every screen, column and action

Complements Setting up each integration. This page covers the screens themselves — the mapping tables, the sync queues and the vulnerability catalogues behind them.

Project mapping — the screen most tickets are about

Every code-scanning integration shows the same table: the tool's projects on the left, your NxSAM assets on the right, and whether they are joined.

Column	What it holds
Project Details / **Original Project Details** / **ProjectVersion Details**	The project as the tool names it
Asset Details / **Mapped Asset Details** / **SSC Project**	The NxSAM asset it feeds
Account	Which tool account it came from, where the integration serves several
Mapping Status	Whether a mapping exists
Sync Status	Whether the last sync succeeded
Last Sync Time	When it last ran

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

> ⚠ **Unmapped means collected and discarded**
>
> A project with no mapping is visible to the integration and has nowhere to file its findings. The sync reports success, and nothing appears in the module.
>
> "The scan ran but there are no findings in NxSAM" is this, far more often than it is a credential or a parsing problem. Check Mapping Status before anything else.

Mapping Status and Sync Status are independent. A mapped project can fail to sync, and an unmapped project can sync successfully and file nothing. Reading only one of the two gives the wrong diagnosis.

Checkmarx additionally distinguishes **Original Project Details** from **Project Details** — the project as first discovered against its current name — so a renamed project stays recognisable rather than appearing as a new one.

Container and cloud sync queues

Prisma reports `Sync Type`, `Image Type`, `Status`, `Initiated at`, `Completed at`.

Image Type separates deployed images from those merely present in the registry. Syncing everything in a registry produces findings on images nobody runs, which is the usual cause of a container backlog that nobody can prioritise.

Cortex XDR endpoints

Column	What it holds
Endpoint Name	The host as the agent reports it
IP Address(es)	Every address it has — plural because endpoints move
Status	Whether the agent is connected
Platform	Operating system
Group	The agent's grouping

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Column	What it holds
Alias	An alternative name
First Seen / Last Seen	The window over which the agent has reported

IP Address(es) and Alias together are why endpoint matching is harder than host matching. A laptop has several addresses over time and may be known by a different name in your CMDB. An endpoint that will not match an existing asset usually needs the alias, not the address.

Connection tables

Each tool integration lists its configured connections:

Column	What it holds
ID	The connection
Clients / **Organization**	Which of your organisations it serves
URL / Base URL and **Port**	Where the tool is
Last Sync / **Latest Config Sync**	When data and configuration were last pulled

Last Sync and Latest Config Sync are different clocks. Configuration sync pulls the tool's projects, policies and scanner definitions; data sync pulls findings. A connection with recent findings and stale configuration will not know about projects created since, which is a common cause of a new project never appearing for mapping.

Clients matters because one scanner installation often serves several organisations. A connection scoped to the wrong client files findings against the wrong tenant.

Severity mapping

Qualys and Tenable both expose a `Risk` to `Risk Rating` table: the tool's own risk value on the left, the NxSAM severity it becomes on the right. Qualys additionally maps by `Scanner`, `Org` and `BU`, with `Configured` showing whether a mapping has been set.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Changing this re-classifies existing findings, because severity is derived rather than stored — the same trap as the CVSS configuration.

Qualys also carries `Asset Property Name` / `Asset Property Value`, which copies a scanner attribute onto the NxSAM asset during sync, so scanner-side grouping survives the import.

Policy mapping

`Policy ID`, `Policies`, `Created On`, `Modified On`, `Audit Files`, `Configuration` — maps a Policy Compliance benchmark to the tool policy that implements it. **Audit Files** are the source reports; **Modified On** is what tells you a mapping changed between two runs that produced different numbers.

Audit Files opens a dialog listing each file behind the mapping: `Title`, `Type`, `Template`, `File Name`, `Created On`, `Modified On`. It exists so that "which benchmark file did this policy actually run" can be answered from the mapping rather than from the scanner — two mappings with the same policy name and different audit files are different benchmarks, and produce different numbers legitimately.

On the Tenable configuration, postures that came from a Tenable scan rather than from an uploaded report are marked **Scanned by Tenable**, which is how to tell the two sources apart when both feed the same audit.

Nexpose configuration reference

`Character`, `Description`, `Example` — the escaping and wildcard syntax accepted in Nexpose configuration fields, with a worked example for each. It is reference material rather than a working screen, and it is there because the syntax is not guessable.

Scan profiles

Where an integration launches scans rather than only collecting them, **Sync Profile** pulls the profile definitions from the tool and **Profile Details** shows what a profile contains — which checks it runs, and against what.

Reading the profile matters when results look thin: a scan that completed against a profile covering forty checks is not evidence of a clean host, and the profile is the only place that distinguishes "nothing found" from "little looked for".

Vulnerability master catalogues

One per tool. The richest are the network scanners:

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

Qualys — `QID`, `Title`, `Vuln type`, `Category`, `Group`, `Sub Group`, `Severity`, `Exploitable`, `Patchable`, `False Positive`.

Nexpose — `VID`, `Nexpose ID`, `Title`, `Category`, `Group`, `Sub Group`, `Severity`, `Exploitable`, `False Positive`.

Nessus — `ID`, `Name`, `Group`, `Sub Group`, `Version`, `Exploitable`, `Risk`, `False Positive`, plus `Family ID`, `Family Name` and `Family Type`. Nessus organises plugins into families, and the family is often the useful unit: excluding a whole family is a single decision where excluding its plugins individually is dozens.

Snyk Code — `Rule ID`, `NAME`. **Acunetix** — `VID`, `VNAME`. **Dependency Track** — `UUID`, `Name`. **Network** and **Red Teaming** — `ID`, `Title`.

AppScan and **Netsparker** carry `Audit Exception` and `Client Exception` — whether an issue is excluded for one audit or for the whole client. The two are separate because a finding irrelevant to one assessment is not necessarily irrelevant everywhere, and recording it at the wrong level either suppresses too much or too little. Netsparker also carries `Issue Type`.

> **Exploitable, Patchable and False Positive are catalogue-level flags**
>
> They apply to the vulnerability definition, not to one occurrence. Marking a QID as a false positive in the catalogue affects **every** finding of that QID, across every host and every audit, including ones imported in future.
>
> That is the right tool for a check your environment genuinely never fails, and the wrong one for a single host where the finding does not apply — use the finding's own status for that.

Patchable is Qualys-specific and worth using: it separates findings a patch resolves from those needing a configuration change, which are different pieces of work assigned to different people.

Qualys detail

`ID`, `URL`, `Threat Intel`, `Remote`, `Authentication Type`, `Additional Info`, `Type`, `Section`, `Description`, `SRC NAME`, `REF`, `Link`, `MW TYPE`, `MW PLATFORM`, `MW ALIAS`, `MW RATING`, `Comments`.

The `MW` fields are malware association — type, platform, alias and rating. A vulnerability with a malware rating is one being actively used, which outranks a higher CVSS score with no known exploitation.

Remote and **Authentication Type** say whether the check ran across the network or needed credentials. An unauthenticated scan that found nothing on a host is much weaker evidence than an authenticated one, and this is where you confirm which it was.

Nexpose detail

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

`Solution Type`, `Summary`, `Title`, `Description` — **Solution Type** distinguishes a patch from a workaround from a configuration change.

Vulnerability mapping tables

One per tool, all the same shape: `ID.`, `<Tool>` *Issue*, `Issue Master`.

This is what makes the same weakness from two scanners one finding rather than two. The tool's own issue identifier on the left, the NxSAM master entry on the right. Acunetix, Burp Suite, Checkmarx, Checkmarx SCA, Dependency Tracker, Fortify, Gitleaks, JFrog X-Ray and Netsparker each have one — the *JFrog X-Ray Issue* column, for example, holds the identifier as Xray reports it.

An unmapped tool issue produces findings that do not deduplicate against anything, so the same vulnerability found by Checkmarx and Burp appears twice with separate SLAs. When a customer reports duplicate findings across tools, this table is the answer.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

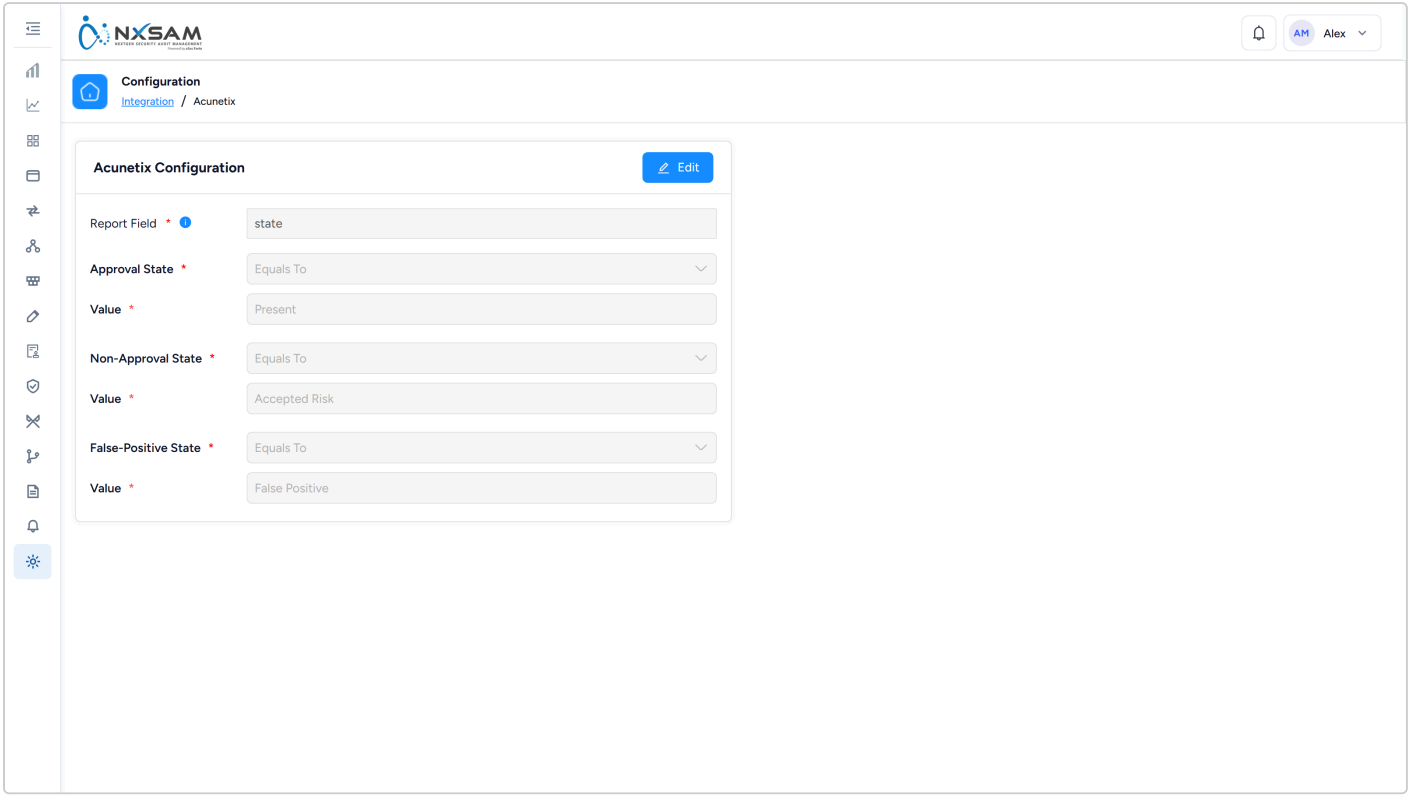
Integrations: find the right screen

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.

Acunetix

Dynamic application scanning.

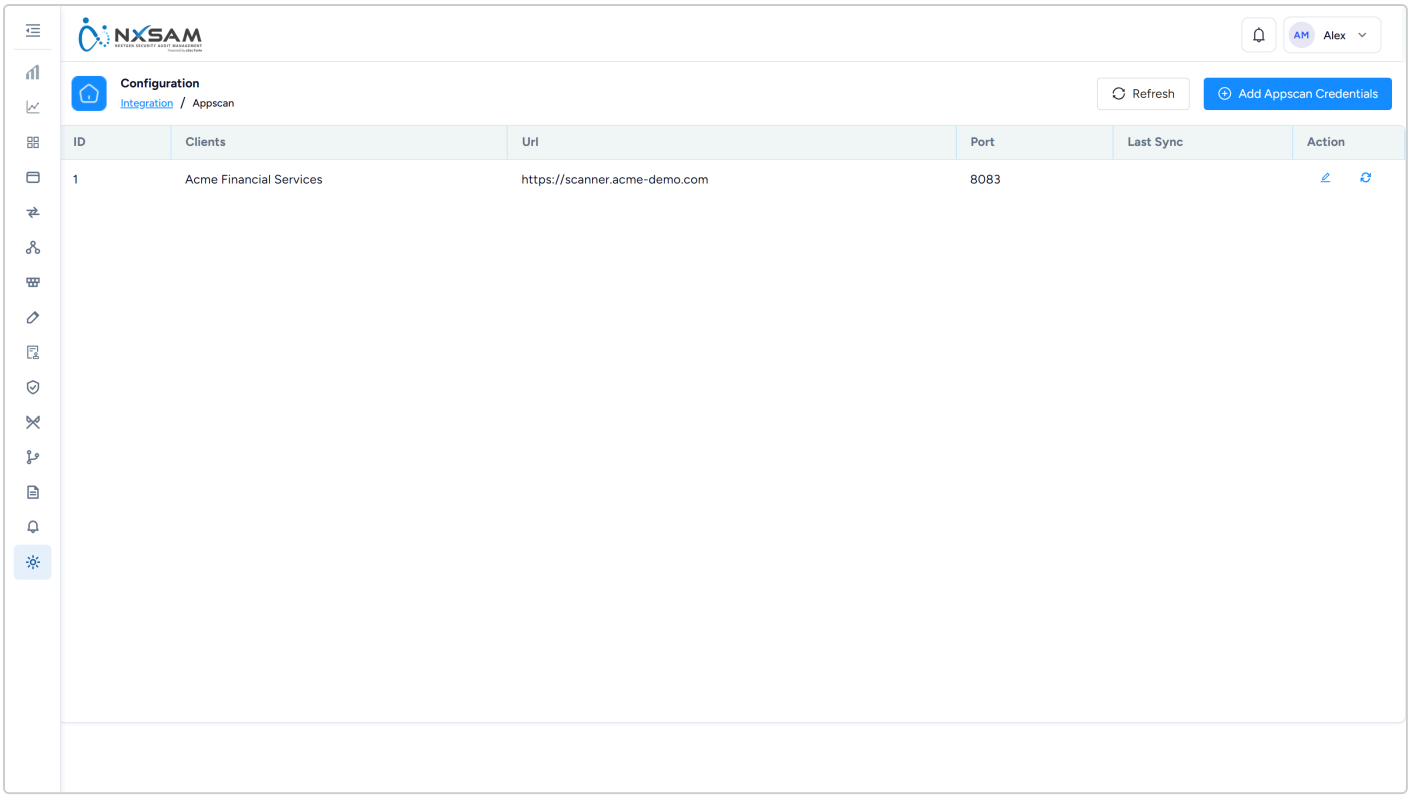


Setting → Integration → Acunetix

AppScan

Application security testing, static and dynamic.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

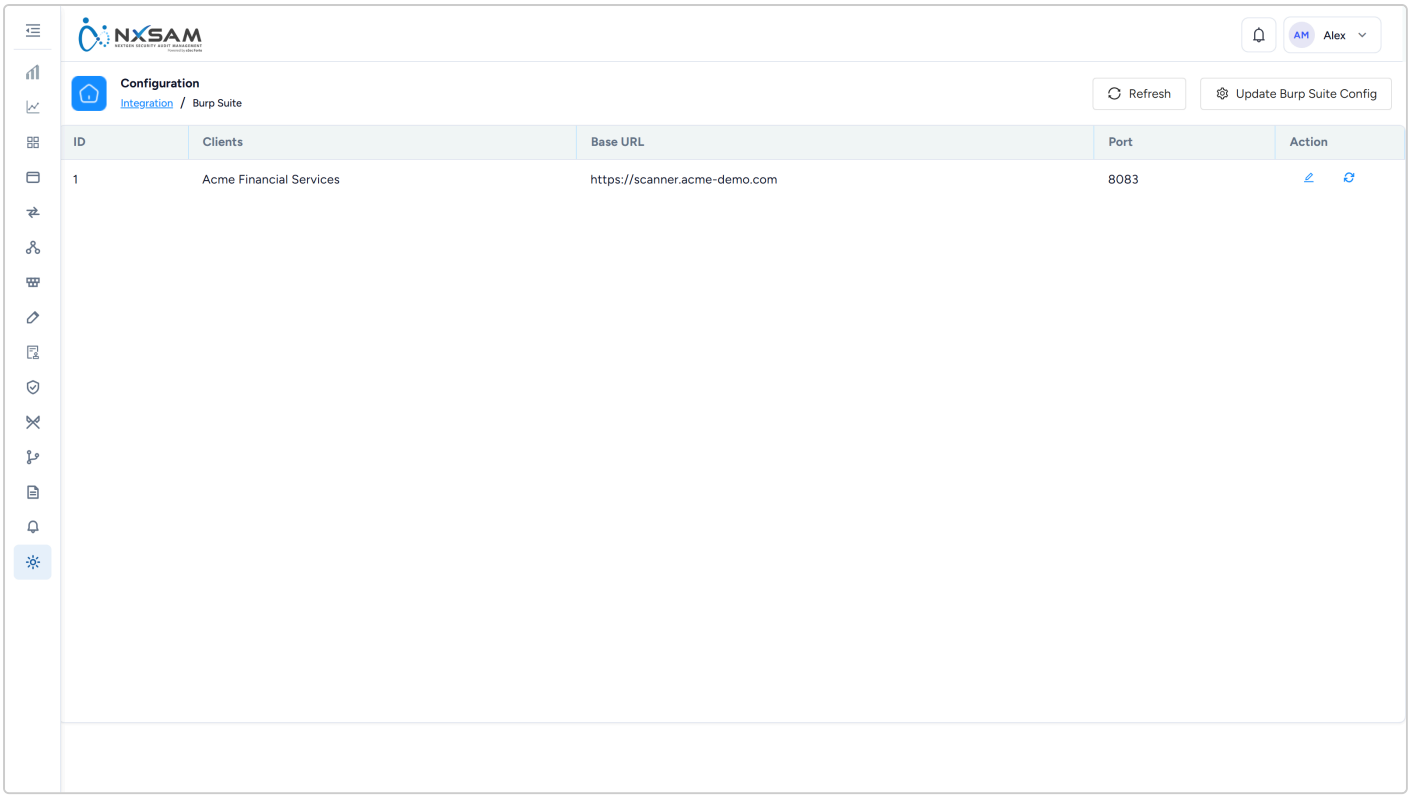


Setting → Integration → AppScan

Burp Suite

Dynamic application testing. Findings arrive against the URL that was tested.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

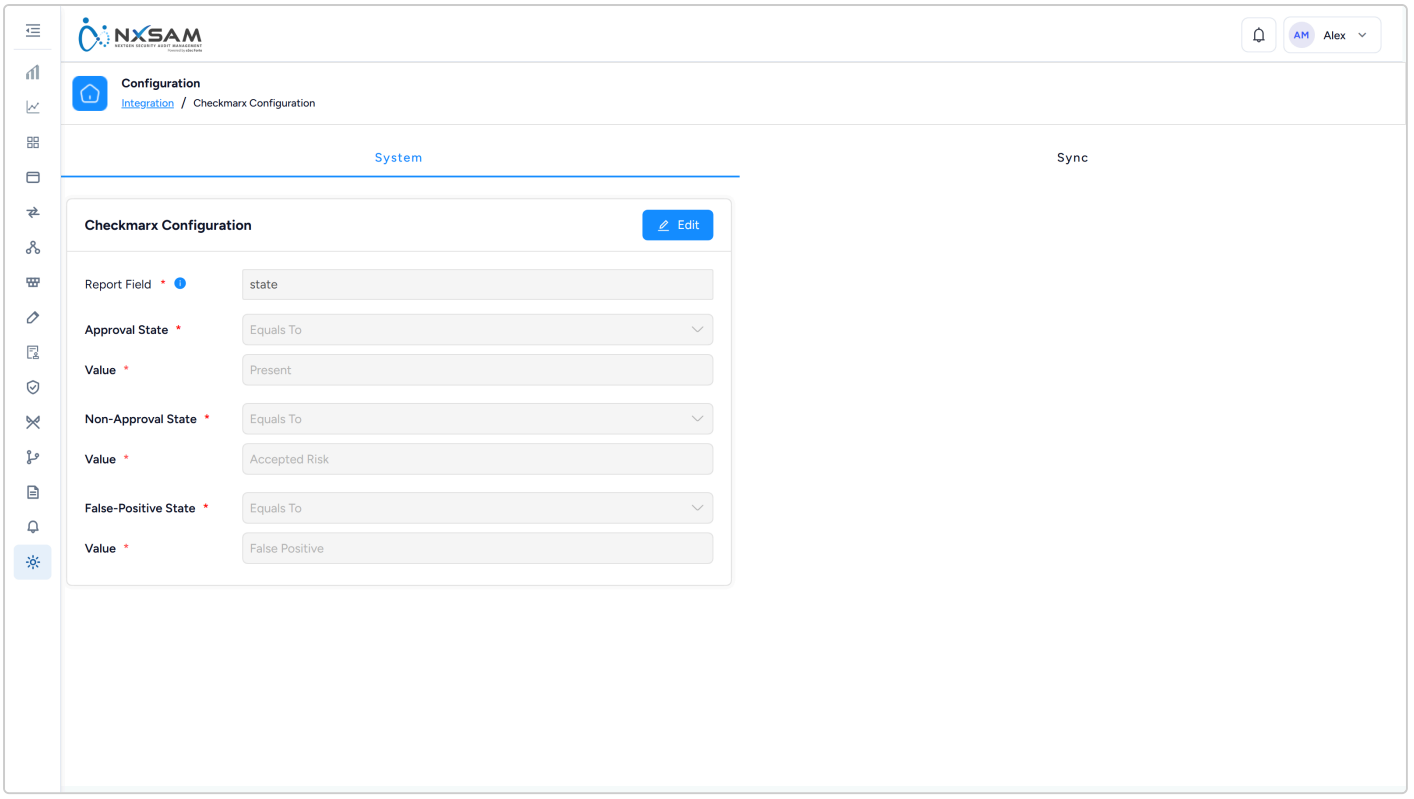


Setting → Integration → Burp Suite

Checkmarx

Static application security testing. Projects map to NxSAM applications.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

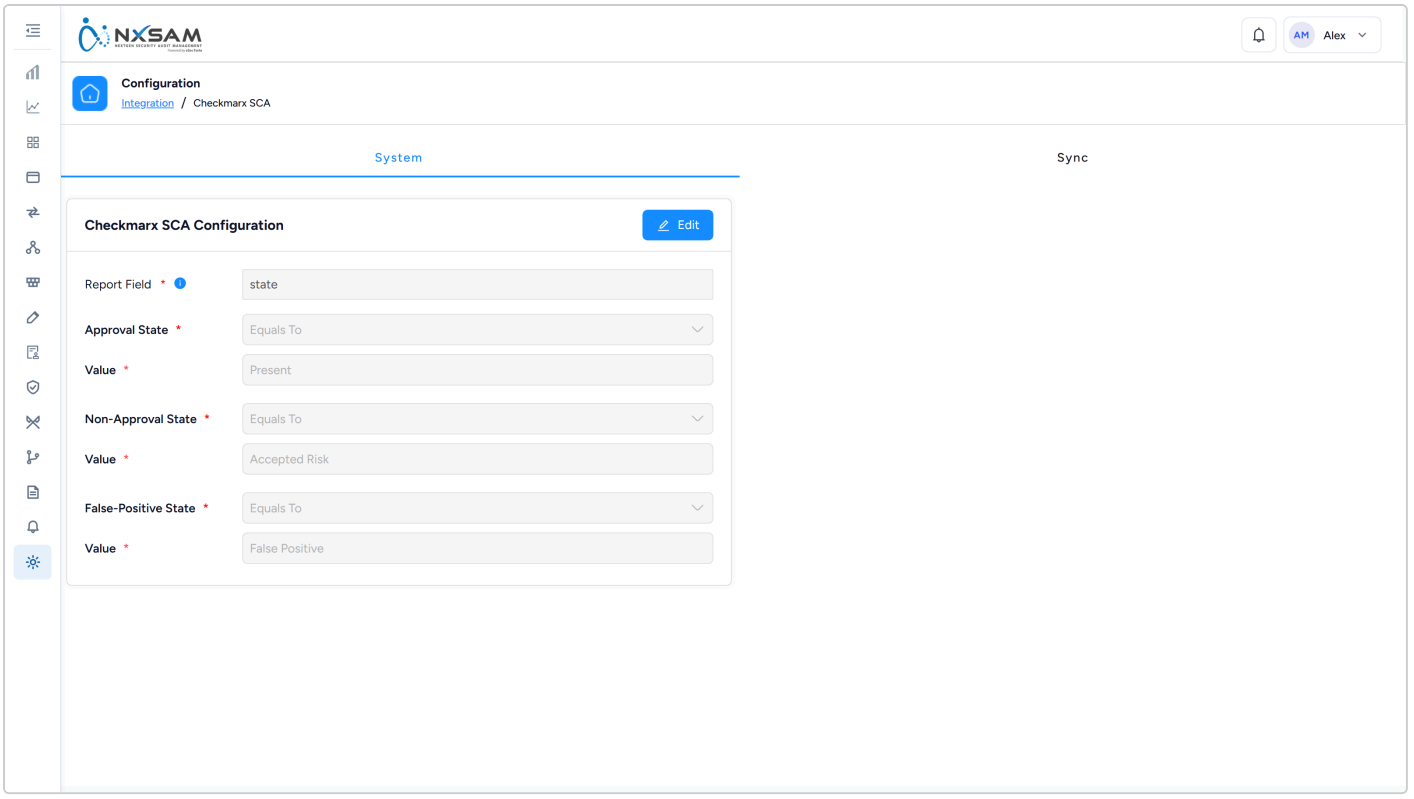


Setting → Integration → Checkmarx

Checkmarx SCA

Software composition analysis — third-party libraries and their known vulnerabilities.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

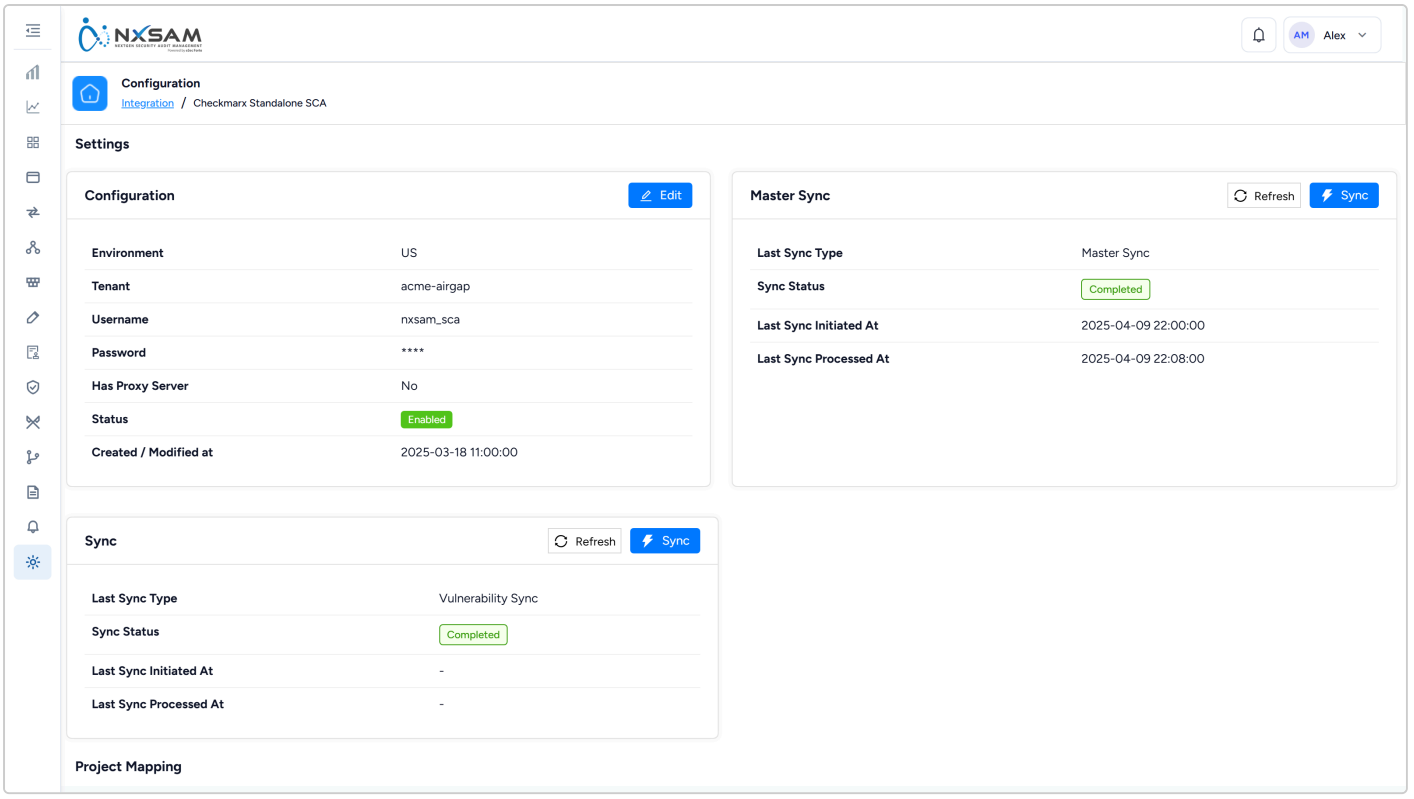


Setting → Integration → Checkmarx SCA

Checkmarx Standalone SCA

SCA run independently of the main Checkmarx platform.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

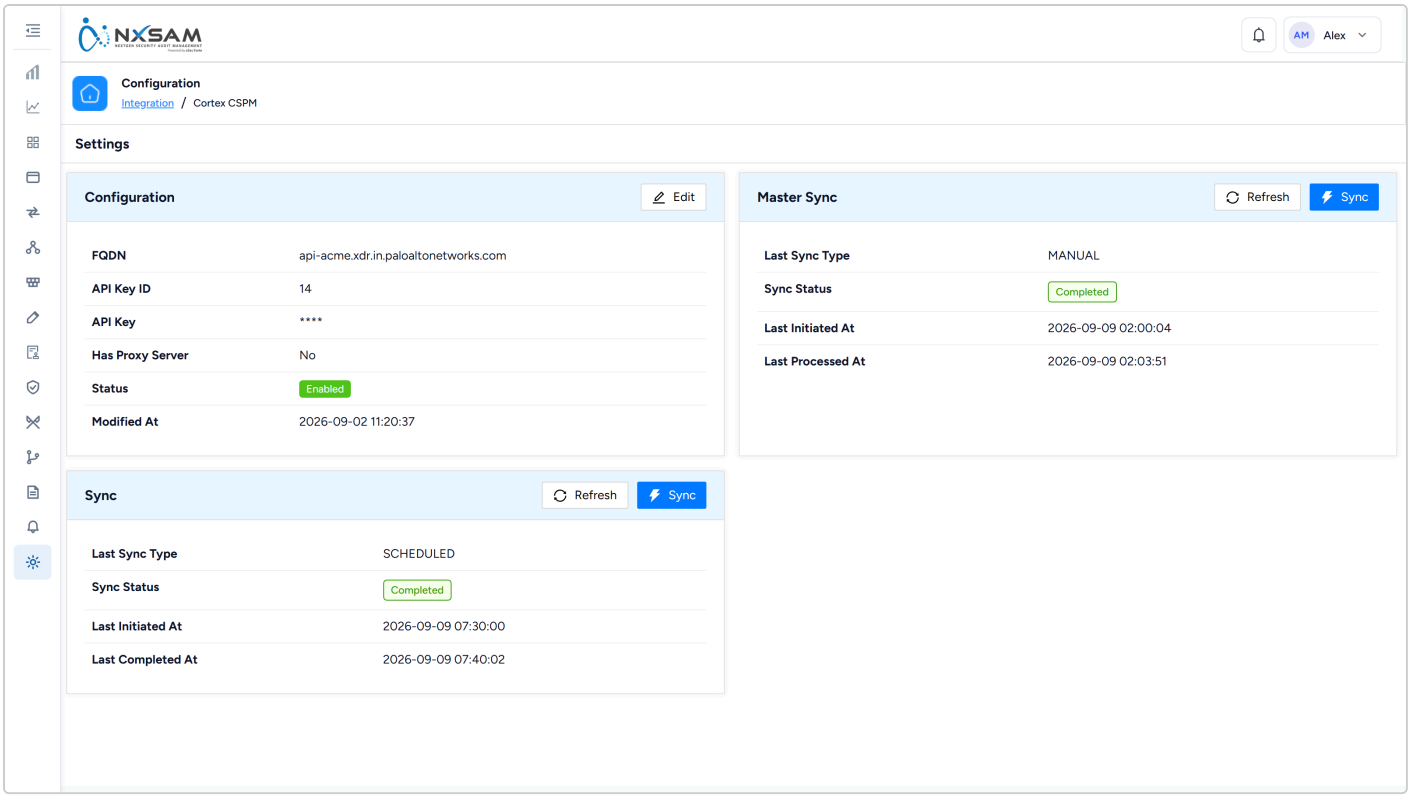


Setting → Integration → Checkmarx Standalone SCA

Cortex CSPM

Palo Alto's posture platform, feeding the CSPM screens rather than the Prisma ones. Master Sync brings rules and policies; Sync brings assets, issues and cases against them — in that order.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

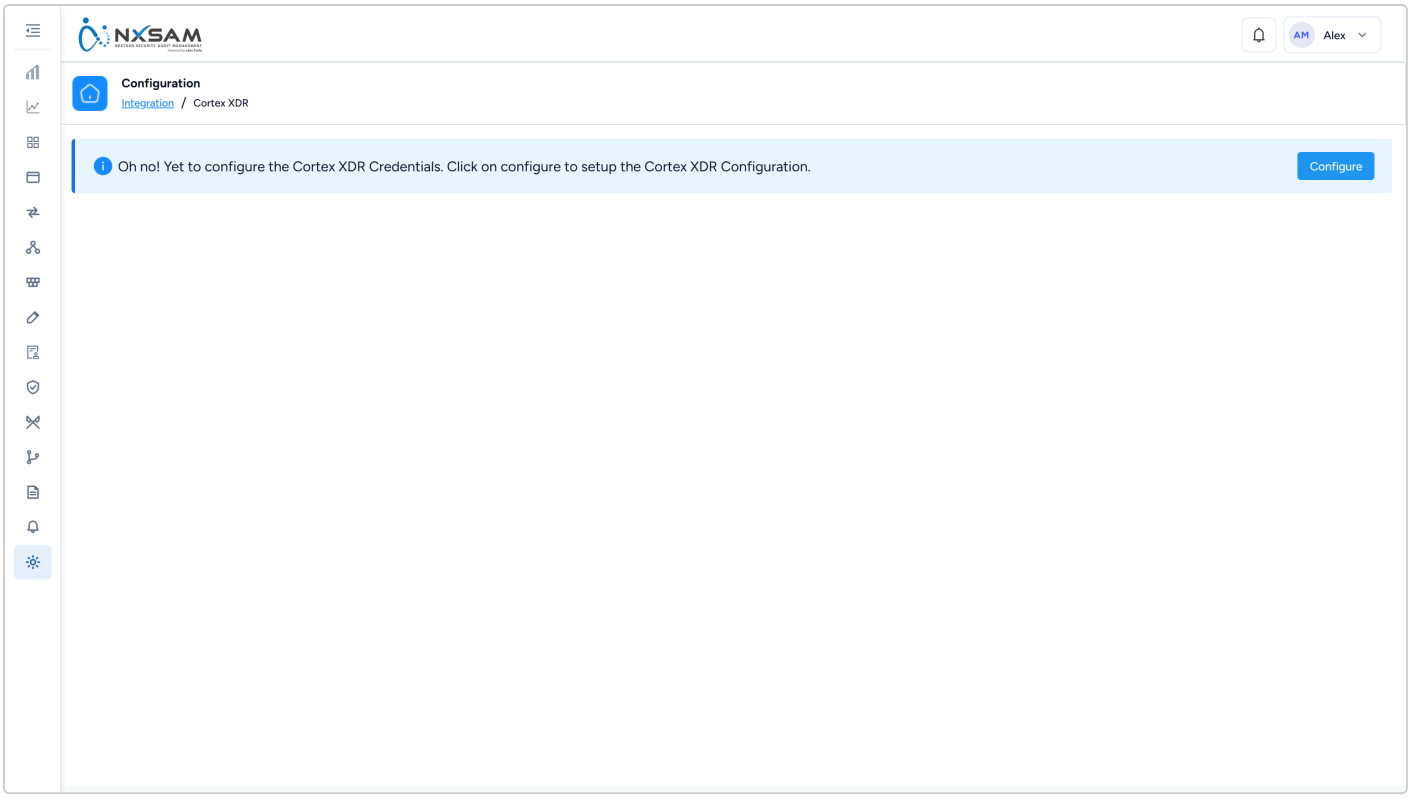


Setting → Integration → Cortex CSPM

Cortex XDR

Host insight findings as a Network VA source. Shown here in its unconfigured state.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

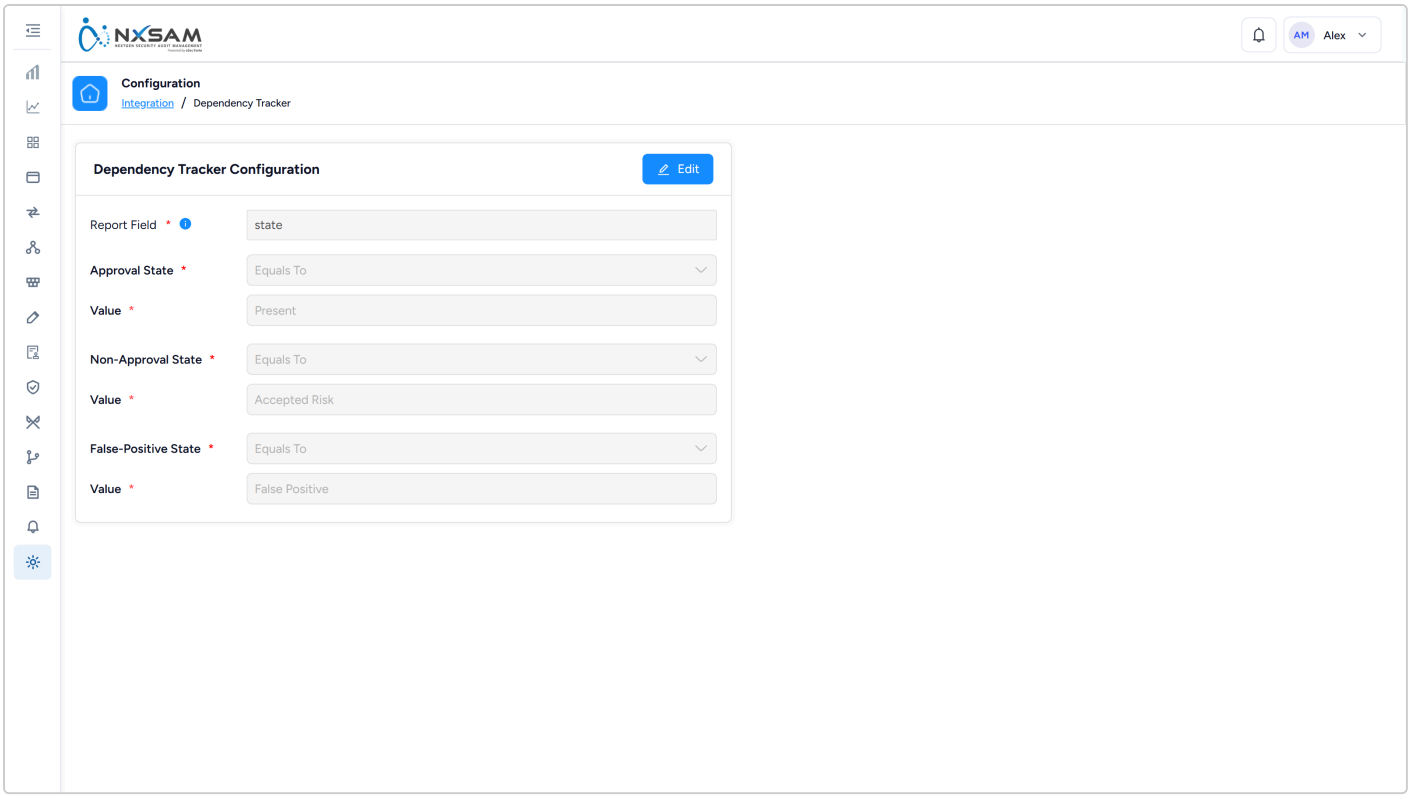


Setting → Integration → Cortex XDR

Dependency Track

Continuous composition analysis from an SBOM rather than from a scan.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

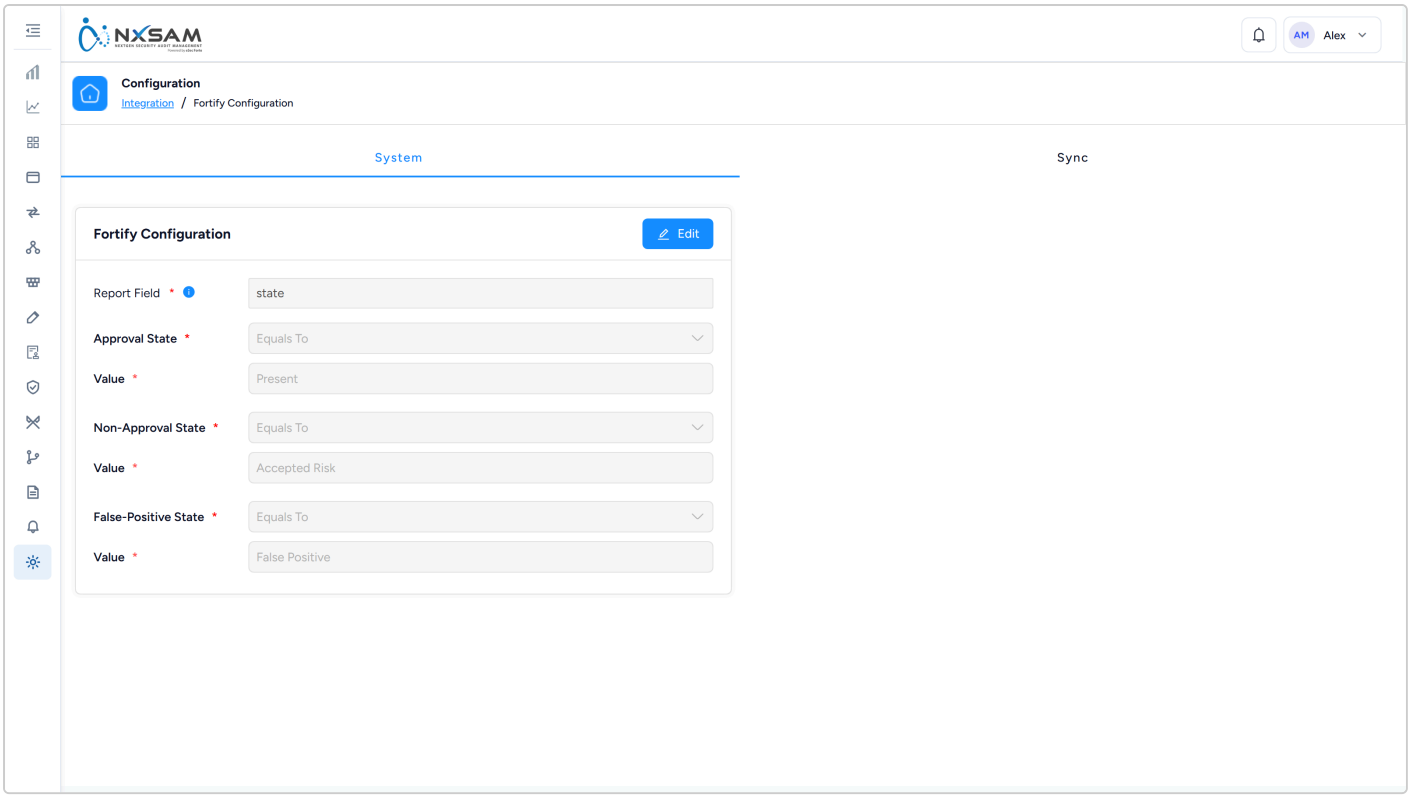


Setting → Integration → Dependency Track

Fortify

Static application security testing.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

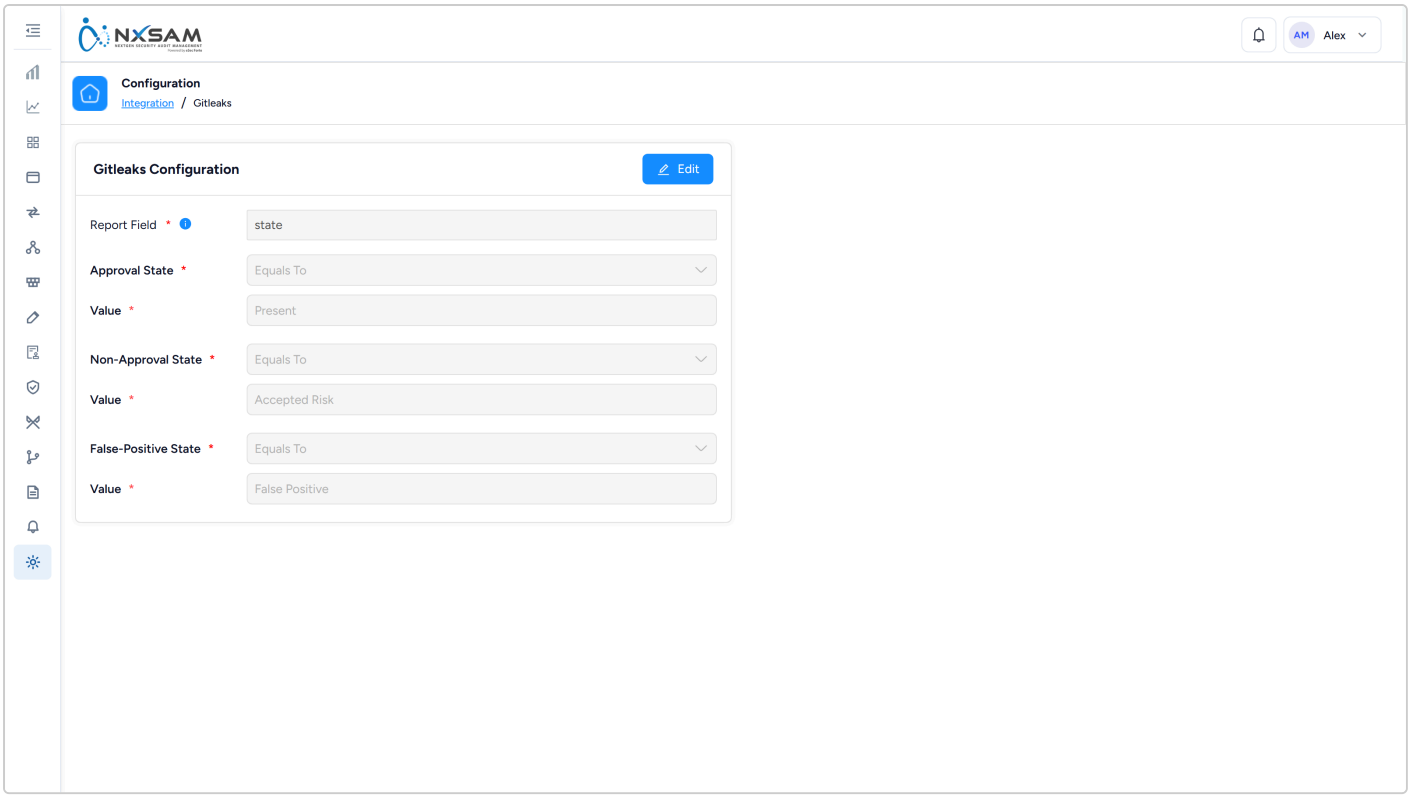


Setting → Integration → Fortify

Gitleaks

Secrets detection — credentials committed into source control.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

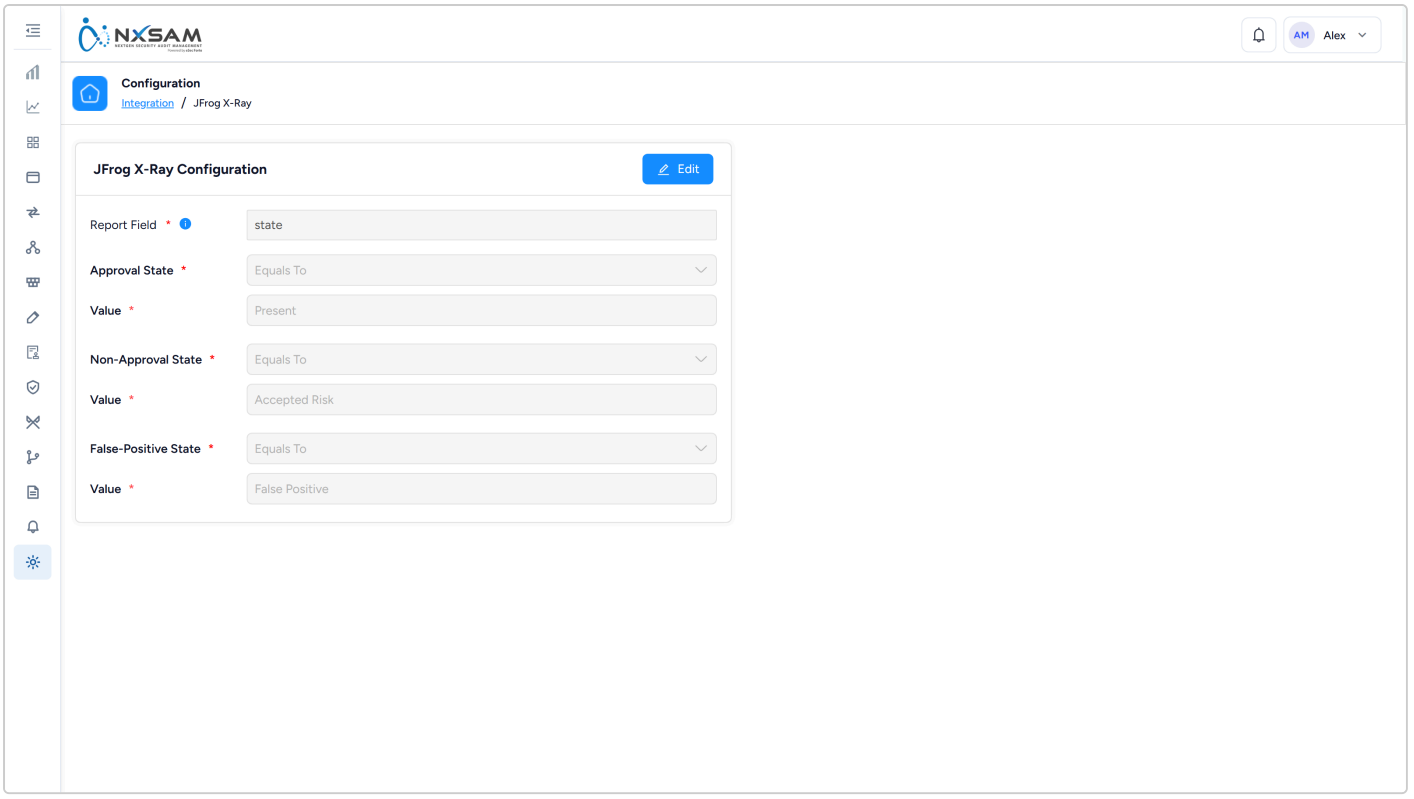


Setting → Integration → Gitleaks

JFrog Xray

Composition analysis over artefacts held in Artifactory.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

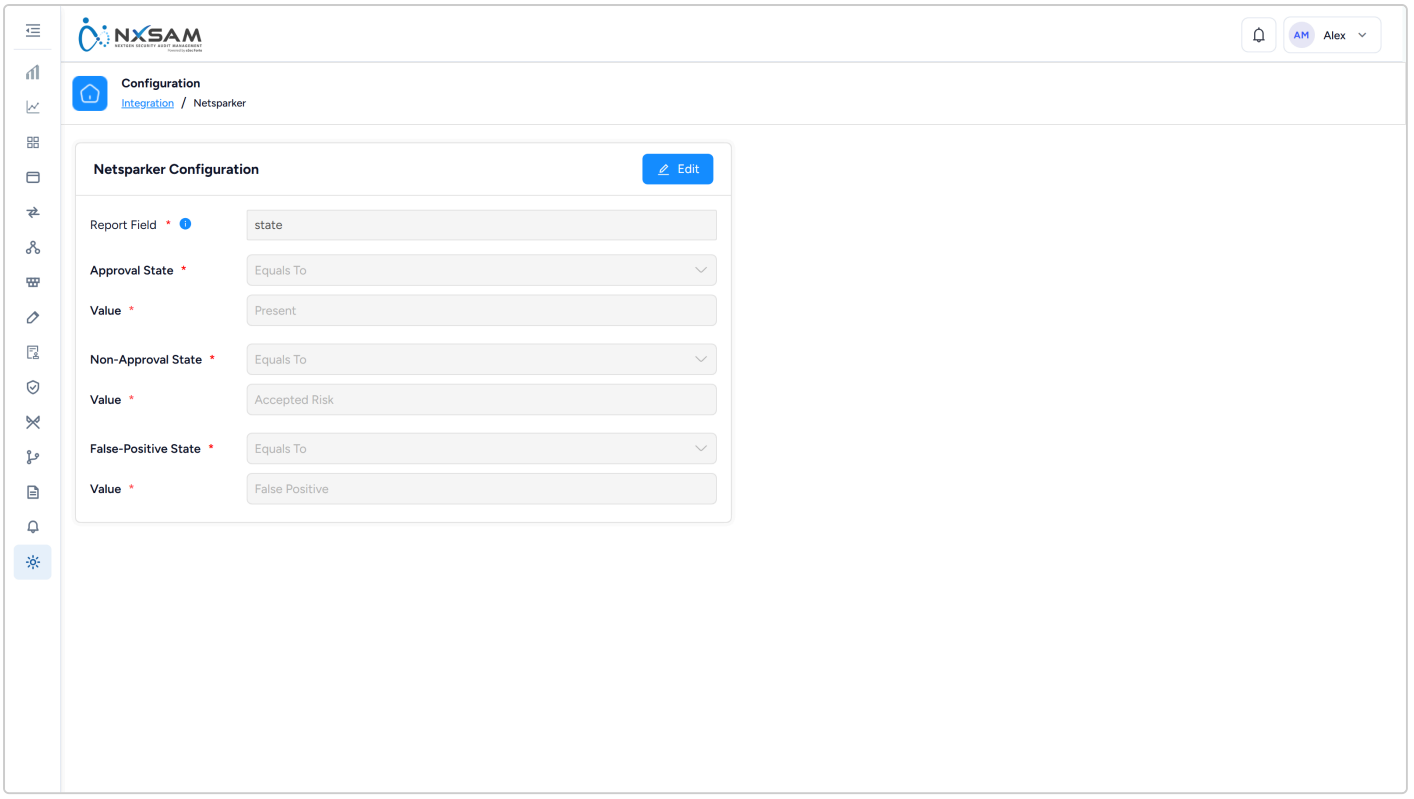


Setting → Integration → JFrog Xray

Netsparker

Dynamic application scanning.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

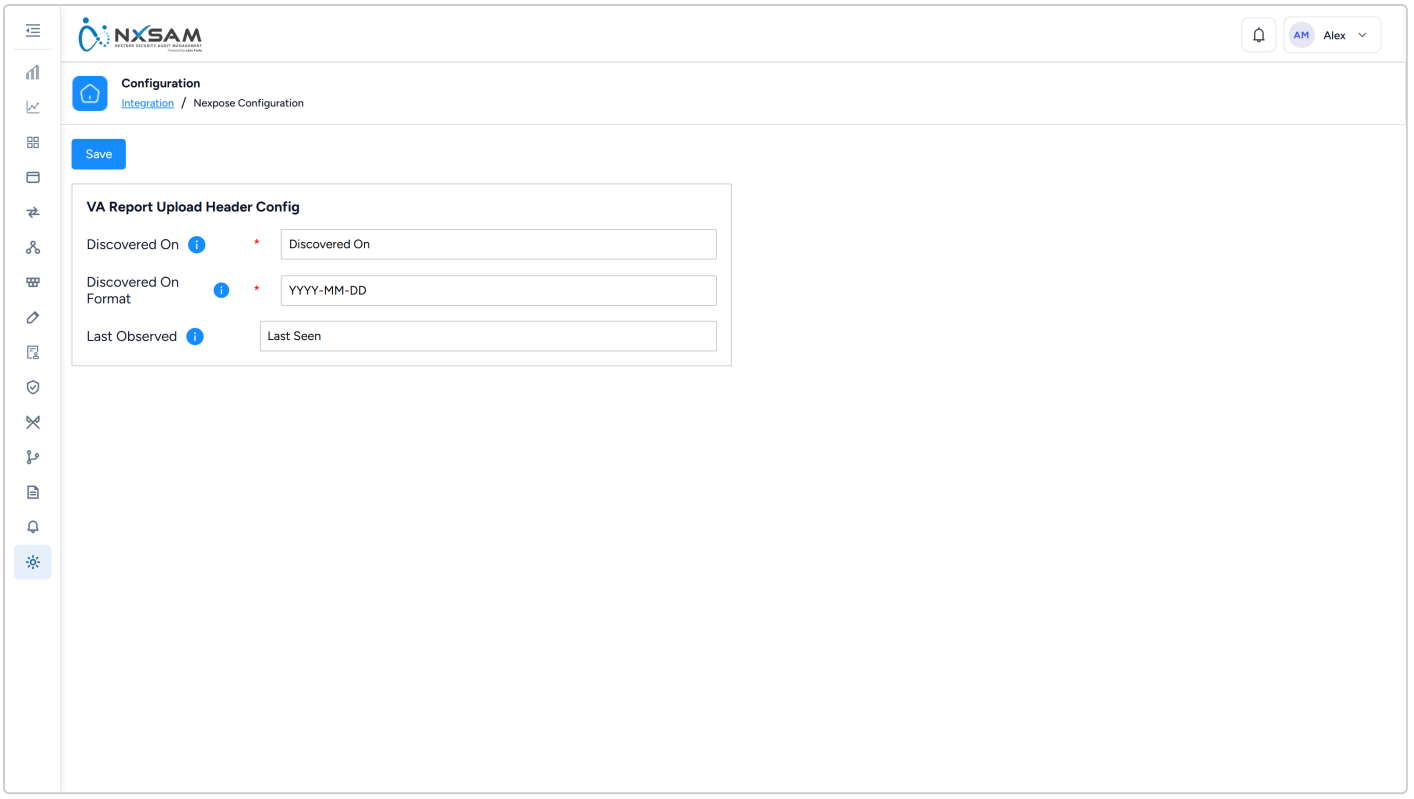


Setting → Integration → Netsparker

Nexpose Upload Mapping

Which column of a Nexpose export carries each date NxSAM needs, and whether it is mandatory.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

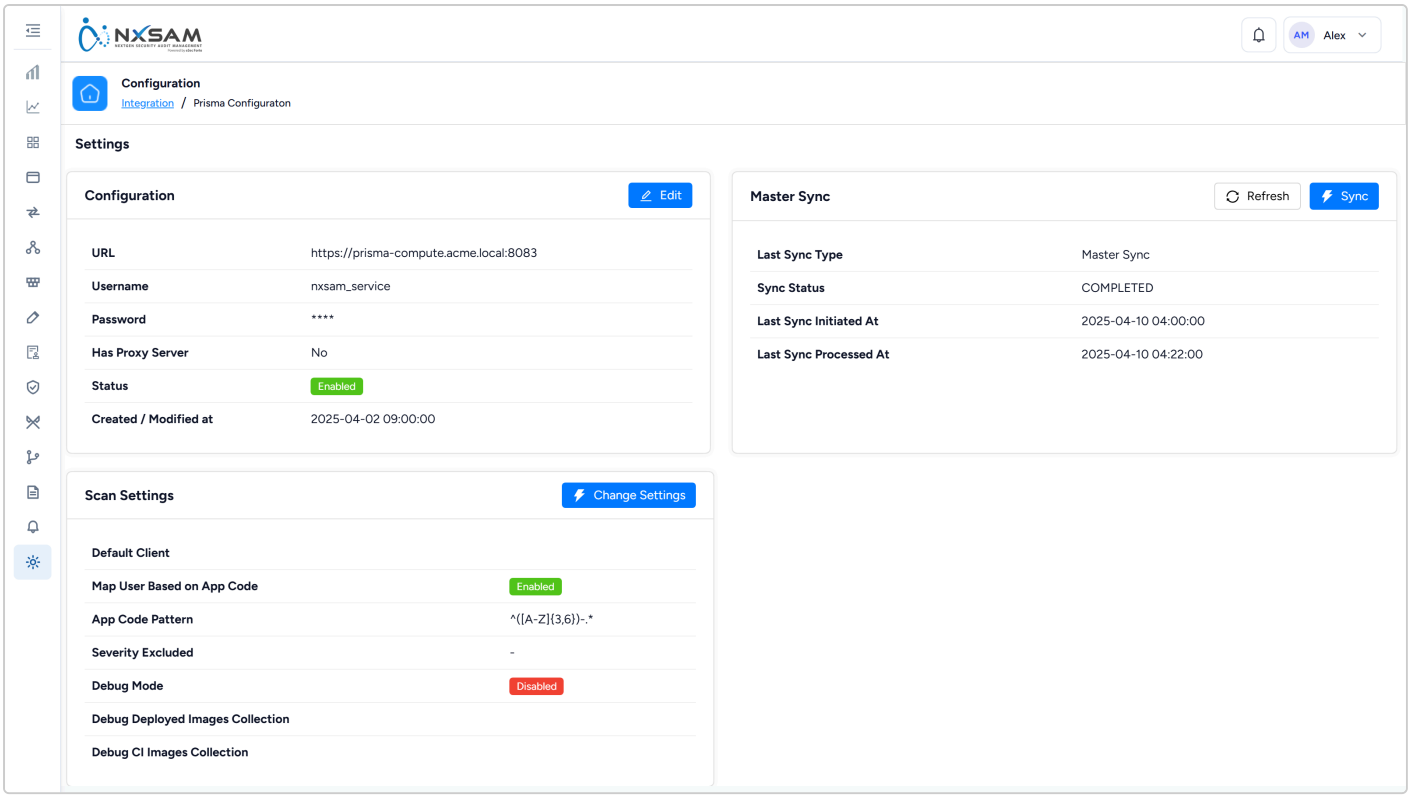


Setting → Integration → Nexpose Upload Mapping

Prisma Cloud

Container security. Connects to the registry scan results and the collections they belong to.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

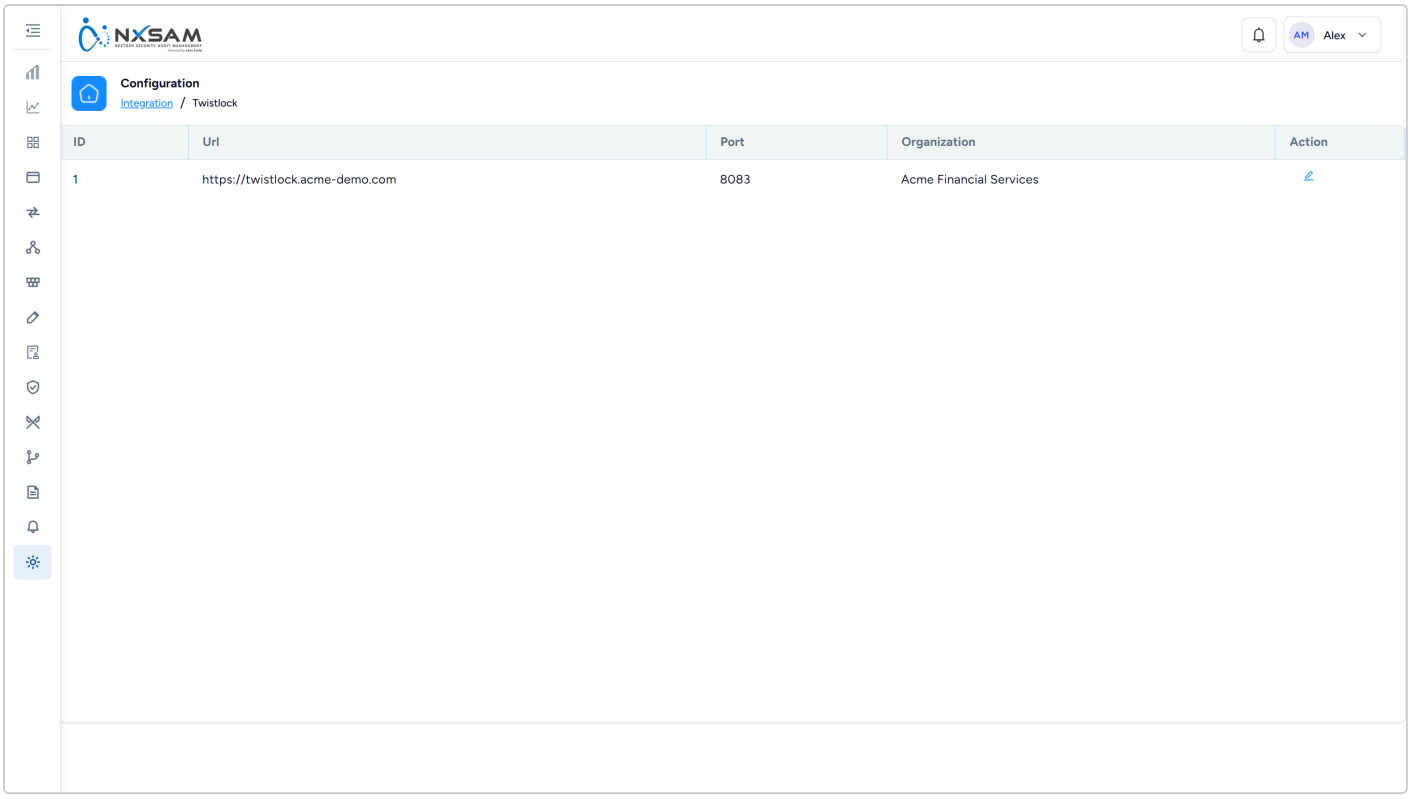


Setting → Integration → Prisma Cloud

Prisma Cloud (Twistlock) Connection

The connection to the container scanner: endpoint, port and credentials.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

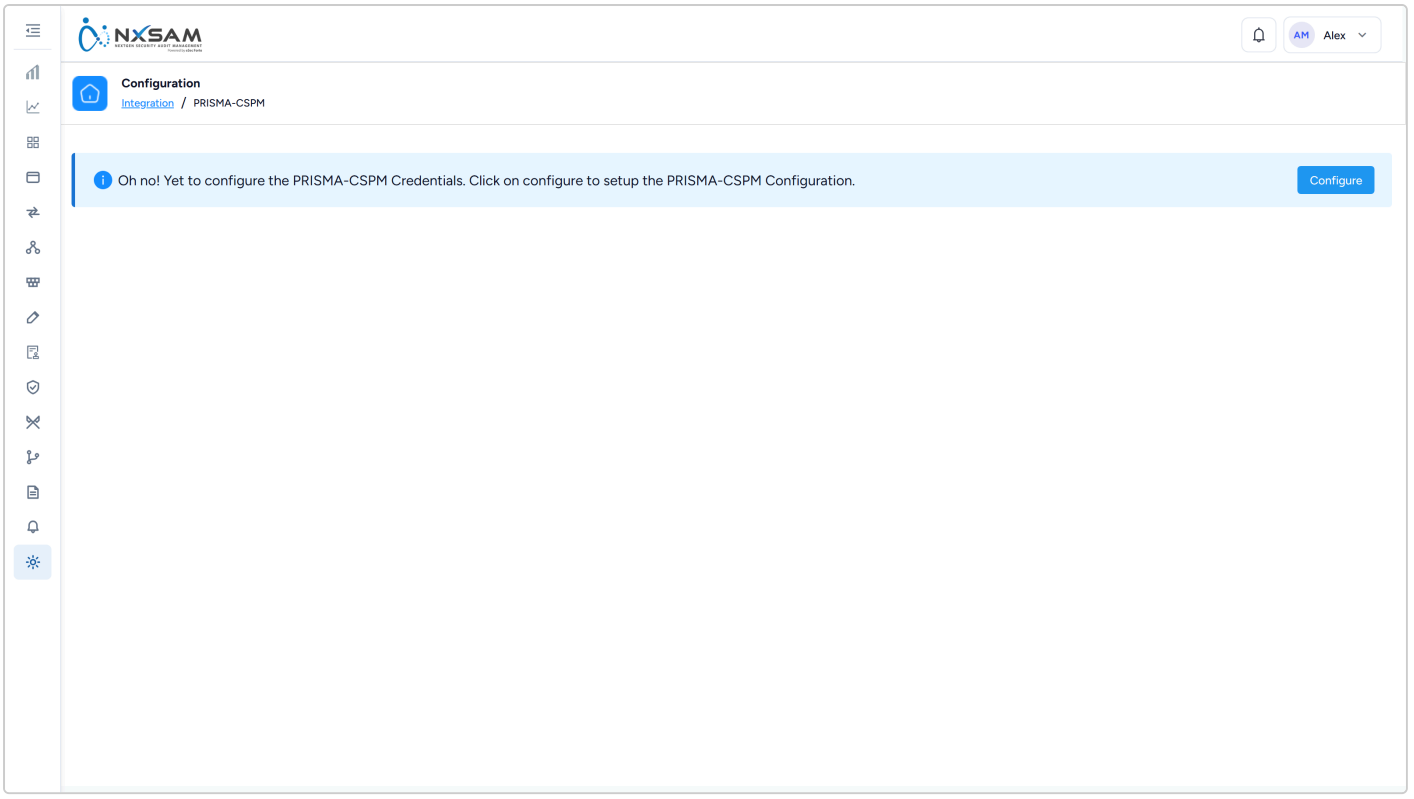


Setting → Integration → Prisma Cloud (Twistlock) Connection

Prisma Cloud CSPM

Cloud posture. Separate from the container integration even though the vendor is the same.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

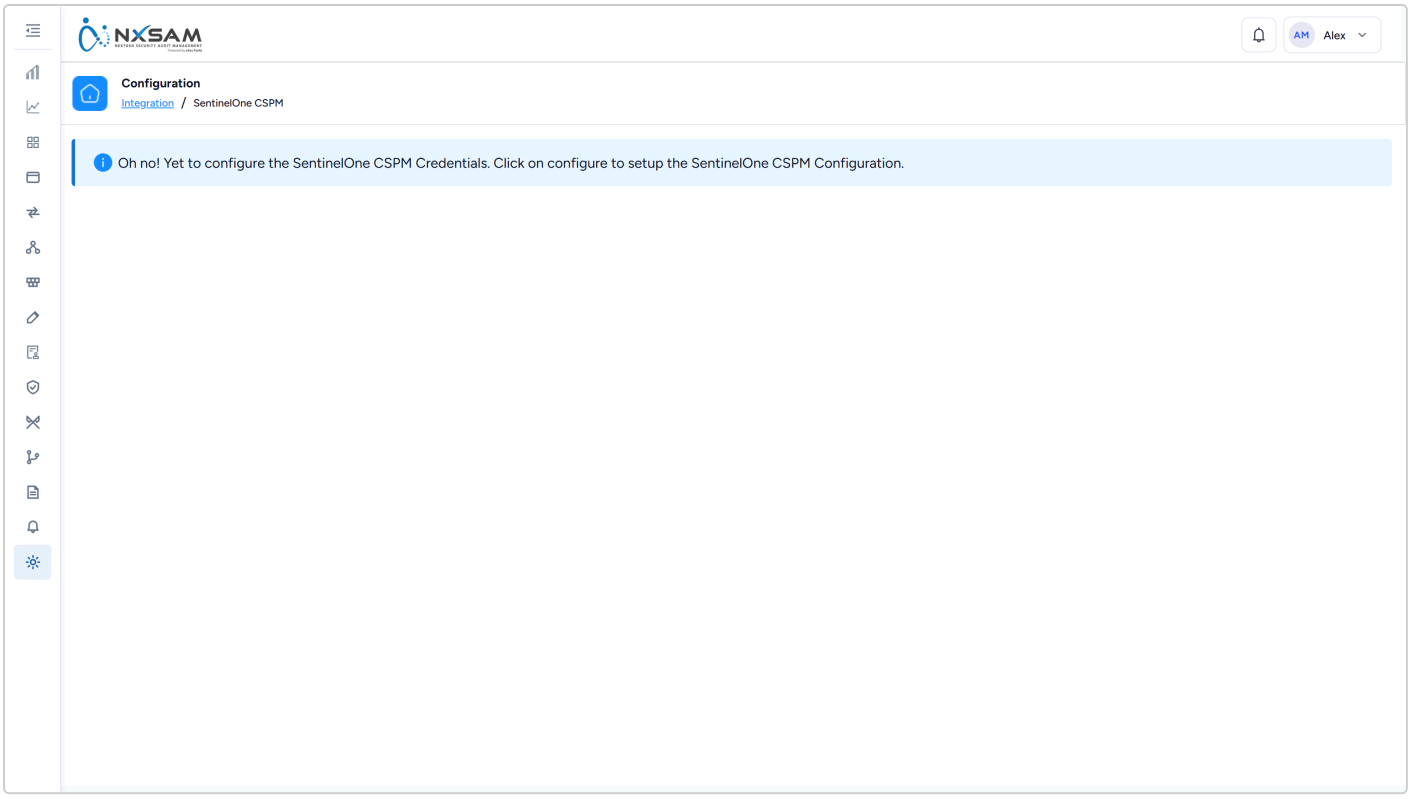


Setting → Integration → Prisma Cloud CSPM

SentinelOne CSPM

An alternative cloud posture source, feeding the CSPM screens. One Data Sync brings everything.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

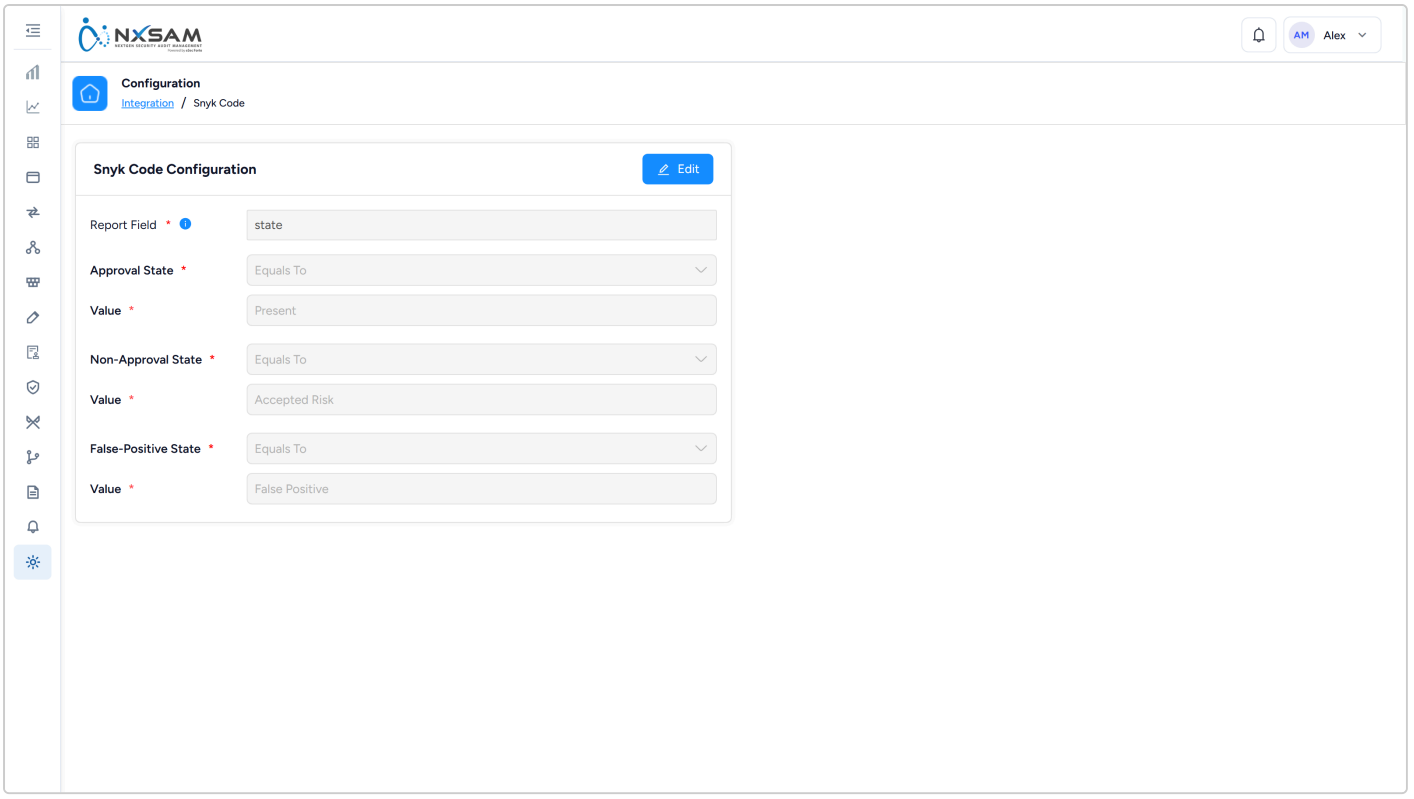


Setting → Integration → SentinelOne CSPM

Snyk Code

Static analysis of source code.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

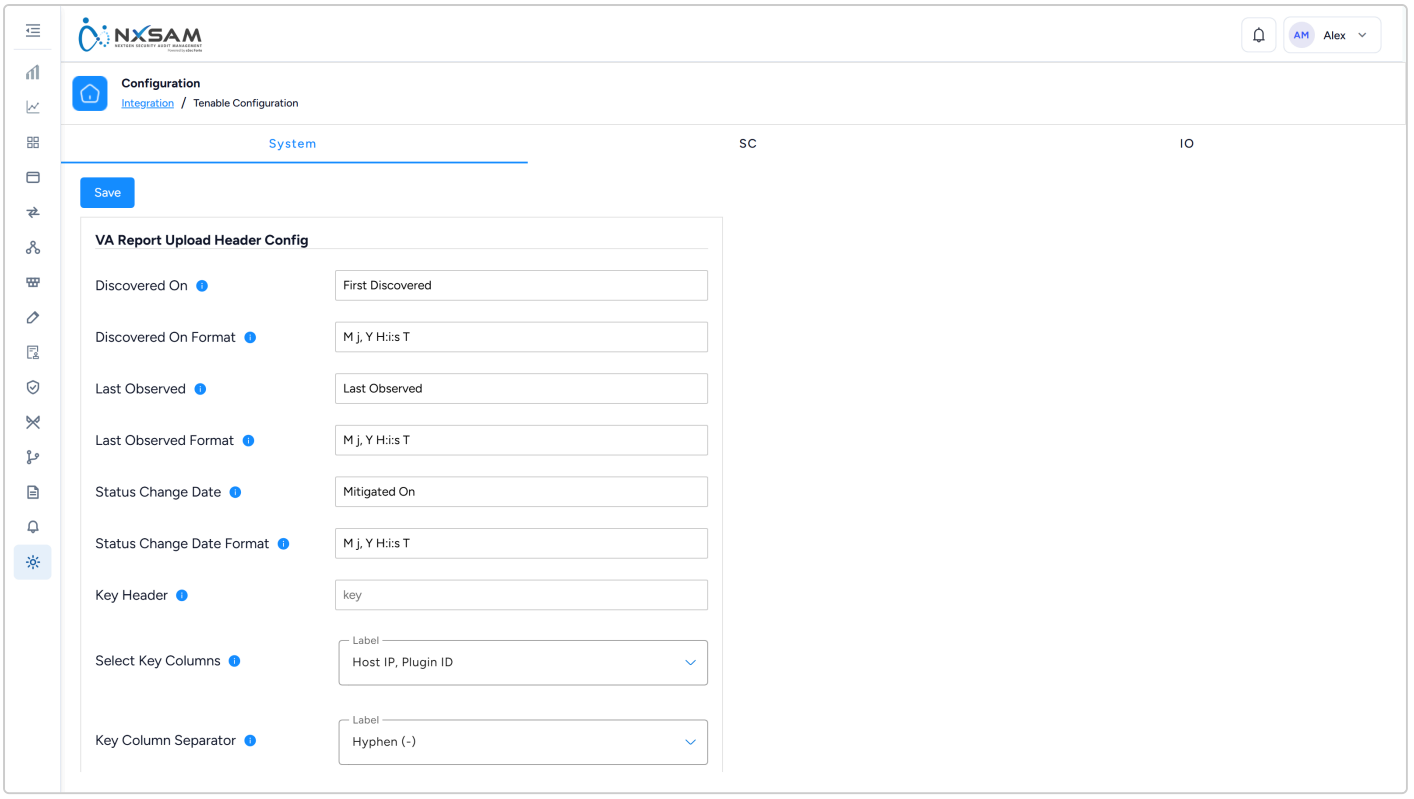


Setting → Integration → Snyk Code

Tenable

Network VA. Credentials, plus the column mapping that decides how the report's dates and keys are read.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.

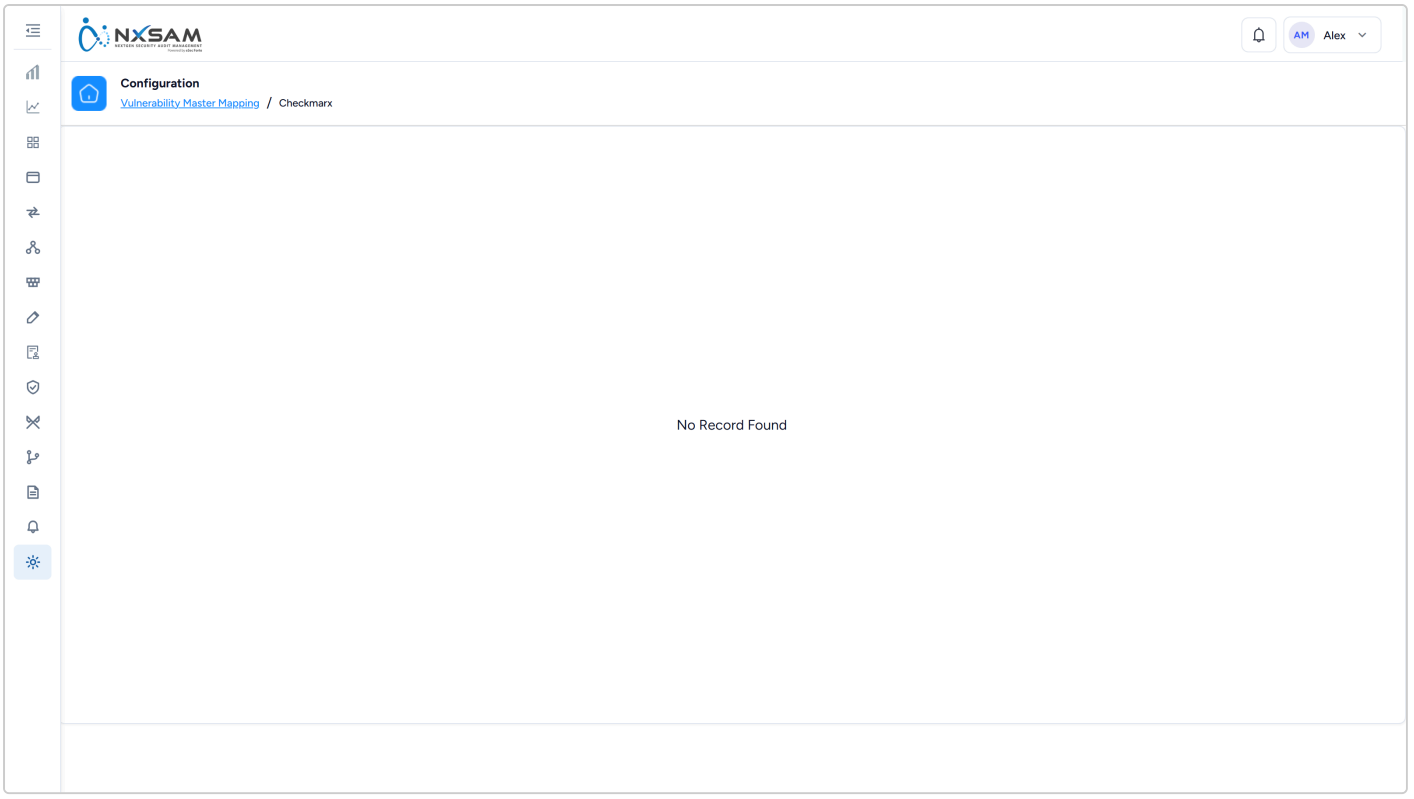


Setting → Integration → Tenable

Vulnerability Mapping — Checkmarx

How this tool's identifiers line up with the catalogue, so the same weakness from two tools is one finding rather than two.

Copyright 2010-2026 eSec Forte® Technologies Private Ltd.



Setting → Vulnerability Master Mapping → Vulnerability Mapping — Checkmarx

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.