PICUS | ATLASSIAN

Ticketing

SOLUTION BRIEF

PICUS PLATFORM AND JIRA CLOUD/SERVER INTEGRATION

Integration Benefits

✔ Validated Findings, Not Noise

Teams open tickets from the exposures they choose to act on, prioritized by what Picus has validated against your live controls, so backlogs fill with work that measurably reduces risk instead of raw scanner output.

✔ Faster Mean Time to Remediate

Teams open tickets directly from Picus and can attach the Remediation Report in the same step, so IT receives the exposure, its validation status, and the mitigation detail together instead of a spreadsheet export.

✔ Evidence-Backed Prioritization

Every ticket carries the exposure's exploitability score and validation context, giving the Priority field a factual basis rather than a CVSS score alone.

✔ Clear Ownership from Day One

Reusable ticket templates predefine target project, work type, priority, assignee, and reporter, so each validated exposure lands on the right board with a named owner.

✔ Cloud and On-Prem Coverage

Picus integrates with both Jira Cloud and Jira Server, so teams get the same ticketing workflow whether Jira runs in Atlassian Cloud or in their own environment.

Closing the Loop Between Validation and Remediation: The Picus x Jira Integration

Jira sit at the center of how most organizations plan, assign, and track work. Security teams, however, usually live somewhere else, and the gap between the two is where validated findings stall. They get exported as a report, pasted into a chat thread, and forgotten before anyone owns them.

Picus integrates with Jira to move validated exposures directly into the workflow your engineering and IT teams already run. **Picus Exposure Validation** determines which findings are genuinely exploitable against your controls, and the integration turns those findings into structured Jira tickets in a single step from the Picus Platform.

By combining validation evidence with templated ticket creation, the Picus Platform ensures that the work reaching your boards is the work that changes risk. Drawing on **Breach and Attack Simulation**, **Autonomous Pentesting**, and **Exposure Validation**, Picus confirms which exposures are truly exploitable, which controls failed to stop them, and then supplies the mitigation content inside the ticket that carries them to closure.

How Picus x Jira Integration Works

1. Picus aggregates exposures from your existing scanners and security tools, then validates them with real attack techniques to confirm which are exploitable against your current controls.
2. A Picus Admin connects the integration entering the Jira instance URL, integration account, and API token.
3. Ticket destination settings and reusable templates define the target project, work type, short description, description, priority, assignee, and reporter for every issue created from Picus.
4. Teams open Jira tickets from Picus with critical fields populated from the template and mitigation guidance available as an attachment, so remediation happens inside the normal IT workflow.

ABOUT PICUS

At Picus Security, our priority is making it easy for security teams to prove what attackers can exploit and what their defenses actually stop.

The Picus Platform brings Breach and Attack Simulation, Autonomous Pentesting, and Exposure Validation together to deliver that proof, then close real gaps with ready-to-deploy fixes and re-validate to confirm.

We help organizations operationalize Continuous Threat Exposure Management (CTEM) with validated evidence and faster remediation.

Experience Picus *in Action*

GET A DEMO

FALL 2026 Leader

4.8/5.0
#1
Solution Provider

4.8/5.0
Highest-rated Vendor

Gartner Peer Insights Customers' Choice 2026

Gartner, Voice of the Customer for Adversarial Exposure Validation
Peer Contributors, 27 August 2026

Strengthening Remediation Workflows With Picus x Jira Integration

Together, Picus and Jira give security and engineering teams a shared, evidence-based definition of what needs fixing. Validation determines whether an exposure is exploitable, and Jira carries that decision through assignment, tracking, and closure without anyone rekeying findings by hand.

The integration removes the friction that usually surrounds remediation handoffs, including inconsistent ticket quality, missing context, unclear ownership, and findings that arrive without any indication of whether they are reachable in the environment. Templates keep every ticket structured the same way, and validation evidence travels with it, so the engineer who picks up the work can see why it matters before opening a second tool.

By pairing Picus Exposure Validation with Jira's project and workflow model, organizations gain a remediation pipeline where prioritization is defensible, ownership is explicit, and progress is measurable against exposure reduction rather than ticket volume.

Validated Ticket Intake Create
Jira tickets only for exposures proven exploitable against your controls, keeping engineering effort focused on real risk.

Faster Handoff to Owners
Faster Handoff to Owners Send each validated exposure to the correct project, work type, and assignee in one step, with no re-entry of findings by hand.

Consistent Ticket Structure
Reusable templates predefine description, priority, assignee, and reporter, so every ticket arrives complete and comparable.

Actionable Remediation Guidance
Attach the Remediation Report to any ticket, so IT receives the mitigation steps alongside the exposure and can start fixing immediately.

Picus Autonomous Exposure Validation Platform

The **Picus Platform** validates security controls, proves exploitability, and prioritizes the fixes that matter. **Breach and Attack Simulation** proves what your controls block and detect, **Autonomous Pentesting** executes real exploit chains to show what an attacker can reach, and **Exposure Validation** confirms whether an exposure is truly exploitable in your environment, even when no working exploit exists or the asset is too critical to execute one against.

Why is exposure validation important?

- CVSS scores show how bad a flaw could be, not whether it is exploitable in your environment.
- Most vulnerabilities are never exploitable once your controls are in place, so validation surfaces the few that carry real risk.
- Patch backlogs grow faster than teams can remediate, and validation shows what to fix first.
- Decisions to patch, mitigate, or accept a risk need evidence, not assumptions.