Customer Kubernetes VDR Testing Summary

A practical pilot guide for risk-based vulnerability data & scanner evaluation

> **Executive Overview:** This pilot evaluates whether Trivy VDR produces explainable, operationally actionable vulnerability data for a customer's Kubernetes environment. The open-source Trivy VDR plugin serves as the local testing engine so teams have full transparency and confidence into how vulnerability determinations are made. In live operations, exploitability, CISA KEV matching, and PAIN scoring are managed in-boundary by **ThreatAlert Security Workbench (TSW)**. The customer retains control of the cluster, classification decisions, and policy application.

Prerequisite: Understand Asset Security Requirements & PAIN

The FedRAMP VDR/VER method is fundamentally tied to CVSS 3.1 Environmental Metrics. A vulnerability's potential operational impact is evaluated in the context of the affected asset's security requirements for confidentiality, integrity, and availability (**CR / IR / AR**). Each workload needs an independently dimensional security-impact profile for PAIN (Potential Agency Impact) to accurately reflect the consequences of compromise.

Customers should review the comprehensive VDR/VER whitepapers and method material, which detail our entire process—our implementation is 100% aligned with them.

> 💡 **Highly Recommended: Interactive PAIN Playground**
> Experimenting with the **PAIN Playground** (`https://stackarmor.github.io/rfc-fedramp-vdr/pain-playground.html`) is **very helpful** for building intuition on how asset security requirements (CR/IR/AR), internet reachability, and CVSS impact severity dynamically change PAIN levels and remediation deadlines.

Labeling every Kubernetes workload manually can be time-consuming and difficult to maintain. The automated ConfigMap skill in `vdr-agent-skills` addresses this by inspecting the Kubernetes environment read-only and making a best-effort assignment for each workload using available evidence (workload names, environment variables, image names, and the established security-impact taxonomy).

Architecture Context: Local Scanner vs. Live TSW

- **Open-Source Transparency:** The Trivy VDR plugin is open source so customers can inspect exactly how the scanner evaluates metadata and reaches determinations.
- **In-Boundary TSW Engine:** When going live, TSW serves as the authoritative engine for orchestrating scans, exploitability tracking, KEV matching, and PAIN scoring. stackArmor continuously ensures the open-source scanner and TSW logic remain closely aligned.
- **Ephemeral Workloads Logic (NIRV in TSW):** In TSW, certain ephemeral workloads (such as batch Jobs, CronJobs, and initContainers where `restartPolicy` is not `Always`) are automatically marked **NIRV (Not Internet Reachable)**. While this logic may not be fully reflected in the local CLI scanner, TSW holds the authoritative scoring rules, which will be published in official product documentation.
- **Internet Reachability & CVE Enrichment:** In the scanner, reachability is determined when a workload is internet-accessible and the CVE Attack Vector is Network (`AV:N`). This represents current scanner capability because deeper metadata (affected protocol, attack path exposure, etc.) is not machine-readable in public CVE feeds today—a known gap that **NIST and NVD are actively seeking to rectify**. stackArmor is leading the way on addressing this and will be submitting our research (`radar-trace-public`) to NIST/NVD for this RFI, with plans to integrate these capabilities directly into TSW in the near future (targeting December 7).

Required Customer Testing Steps

1. Review the Method & PAIN Playground (Optional but Recommended)

Before testing, customers may review the VDR/VER whitepapers and experiment with the **PAIN Playground** to understand the differences between: scanner severity, PAIN (Potential Agency Impact), exploitability / known exploitation (CISA KEV), internet reachability, and FedRAMP remediation deadlines.

2. Installation Prerequisite & Cluster Permissions

Trivy **0.72.0** is the minimum version for the pilot. Newer releases are fully supported for local testing. Install Trivy and the VDR plugin:

```
# On macOS or Linux with Homebrew:
brew install trivy    # or: brew upgrade trivy

# On Linux via installer script:
curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh \
  | sudo sh -s -- -b /usr/local/bin

# Install and verify the VDR plugin:
trivy plugin install github.com/stackArmor/trivy-plugin-vdr
trivy plugin list
trivy vdr --help
```

> 🔑 **Cluster Permissions & IAM Access**
>
> **Local Pilot Testing:** Ensure your active `kubectl` context has sufficient permissions across cluster namespaces as defined in the plugin documentation. A `cluster-admin` role (or equivalent broad read access across workloads, services, ingresses, and configmaps) is sufficient.
>
> **Live Production Rollout:** When preparing to go live, stackArmor will reach out to coordinate and ensure our dedicated service account / IAM role is provisioned with the necessary read-only access on your cluster.

3. Generate the Candidate ConfigMap (`vdr-agent-skills`)

Run the ConfigMap-generation skill in the `vdr-agent-skills` repository (formerly *trivy-plugin-vdr-skills*) against your Kubernetes context. Generation is read-only: it inventories workloads using `kubectl get`, never retrieves Secret values, and does not alter the cluster.

The workflow produces three artifacts:

- `workload-inventory.json` — all inventoried workloads;
- `vdr-fedramp.yaml` — the candidate classification ConfigMap; and
- `assignment-coverage.json` — the coverage ledger with confidence scores and rationale.

> ⚠️ **Importance of Applying the ConfigMap**
>
> The ConfigMap should be generated and applied to the cluster **before** running the scan. If no ConfigMap is present, the scanner defaults to the **highest security requirements (`H/H/H` / unclassified)** for every unlabeled asset, resulting in significantly noisier scan results.

4. Review and Apply the ConfigMap

Before applying the ConfigMap, review the generated classifications:

> 🎯 **Pilot Labeling Guidance: Pragmatism Over Perfection**
>
> While labeling precision is important when going live on **December 7, do not spend excessive time perfecting every label right now** during the pilot.
>
> - Give generated assignments a cursory review with a quick look at `archetype-guide.md` for CR/IR/AR requirements.
> - If a label does not fit, simply select an existing archetype that has the corresponding security requirements (CR/IR/AR).

During review, confirm that: every workload has an intentional profile; `multiAgency` matches your tenancy model; reachability accurately reflects ingress classes; and the optional `securityRequirementsCeiling` can be safely ignored for this pilot.

```
kubectl apply -f vdr-fedramp.yaml
```

5. Run Trivy VDR Against the Cluster

After the ConfigMap is applied, execute the plugin against the Kubernetes cluster:

```
trivy vdr k8s --all-namespaces --view resources \
  --output vdr-k8s.json --html-output vdr-k8s.html
```

Preserve `vdr-k8s.json` as the source artifact. Open `vdr-k8s.html` locally to interactively inspect findings, PAIN tiers, reachability filters, and remediation deadlines.

Out-of-Scope Workflows & Advanced TSW Capabilities

The following finding-level lifecycle actions are **not in scope** for this local standalone scanner:

- Marking findings as **Mitigated**, **False Positive**, or **Vendor Dependency**; and
- Granular per-finding overrides for **internet reachability** or **exploitation likelihood**.

These capabilities and deviation workflows are being built directly into **TSW**. Our goal is to provide an in-boundary testing phase prior to go-live in December so customers can explore and evaluate these workflows hands-on, as development milestones allow.

> 🔍 **Demo Option: Preview Your Cluster Data in TSW**
> If your team would like an early preview of how your environment looks inside the new interface, you can share your scan JSON (`vdr-k8s.json`) with stackArmor. We will import your data into our **FedRAMP High environment** and host a walkthrough demo to show how your workloads, findings, and workflows render in TSW (note: this walkthrough demo is non-hands-on).

Target Roadmap & Next Steps

stackArmor is actively engaging all customers now to target a complete operational rollover by **December 7, 2026**. While certain systems may fall into a short grace period, we are doing everything in our power to complete all validation ahead of time so systems do not need to rely on grace periods.

Phase	Milestone	Description & Action Items
Phase 1: Local Pilot	Immediate	Use open-source Trivy VDR plugin & `vdr-agent-skills` for local snapshot & baseline exploration.
Phase 2: Preview Update	Prior to Dec 7	Update preview version running in your environment for hands-on validation & feedback.
Phase 3: In-Boundary Go-Live	**December 7, 2026**	Full in-boundary deployment of TSW within your system; provision dedicated IAM access.

Pilot Deliverables & Feedback

Share the reviewed `vdr-fedramp.yaml`, `vdr-k8s.json`, and qualitative feedback via an approved secure transfer channel. Please report which data, classifications, and workflow actions are most useful or require refinement.

Relevant Links & Resources

- **FedRAMP VDR/VER Whitepapers & Specifications:** https://stackarmor.github.io/rfc-fedramp-vdr/
- **PAIN & Remediation Interactive Playground:** https://stackarmor.github.io/rfc-fedramp-vdr/pain-playground.html
- **Payload Path Reachability Explorer:** https://stackarmor.github.io/rfc-fedramp-vdr/reachability-playground.html
- **Trivy VDR Plugin Repository:** https://github.com/stackArmor/trivy-plugin-vdr
- **VDR Agent Skills Plugin Repository:** https://github.com/stackArmor/vdr-agent-skills
- **RADAR Trace Public Repository:** https://github.com/stackArmor/radar-trace-public
- **RFC FedRAMP VDR GitHub Discussions & Feedback:** https://github.com/stackArmor/rfc-fedramp-vdr/discussions
- **NIST SP 800-60 Rev. 2 Draft & Information Types RFI:** https://csrc.nist.gov/publications/detail/sp/800-60/rev-2/draft