SEVEN SWELL AI LABS



RESEARCH WHITE PAPER

Port Guardian

Network & Process Lease MCP

PID-Bound TCP Port Leases & Packet Aggregation.

Preventing port collisions and orphaned listener processes across concurrent local agent runs.

Field	Value
Document	Architecture Note 02-B
Author	Thaddeus Sylvester Lee
Date	November 2025 (Updated 2026)
Classification	Production Lab Tooling

Seven Swell AI Labs
https://sevenswell.ai/#port-guardian

Document Profile

Field	Value
Document	Architecture Note 02-B
Headline	PID-Bound TCP Port Leases & Packet Aggregation.
Lead	Preventing port collisions and orphaned listener processes across concurrent local agent runs.
Author	Thaddeus Sylvester Lee
Company	Seven Swell AI Labs
Date	November 2025 (Updated 2026)
Classification	Production Lab Tooling

Abstract

Concurrent local agent workflows frequently collide on localhost ports and leave orphaned listener processes behind after crashes. Port Guardian is an MCP server providing PID-bound TCP port leases, sliding-window TCP/UDP packet aggregation, and automated orphan cleanup, paired with a Cloudflare Zero Trust MCP tool for automated tunnel provisioning.

01 / The Problem

Reference Signals

Signal	Value
Unbound Local Ports	Collisions
Lease Registry	PID-Bound

Port Collisions & Orphan Listeners

When multiple agents start dev servers and test APIs simultaneously, hard-coded ports collide and crashed sessions leave stale processes listening on sockets without attribution.

02 / Core Capabilities

PID-Bound Lease Controller

Allocates TCP ports with explicit metadata binding project name, session ID, sandbox path, and process ID to every active lease.

Sliding-Window Packet Sniffer

Aggregates live TCP and UDP traffic on leased ports into sliding time windows so agents can diagnose connection errors without parsing raw packet floods.

Orphan Process Cleanup

Background sweeps detect processes listening on unregistered ports or linked to dead sessions and terminate them to reclaim localhost capacity.

Cloudflare Zero Trust Integration

Works alongside a Cloudflare Zero Trust MCP server that provisions tunnels and maps local leased ports to public hostnames via API without manual setup.

03 / Operational Workflows

Port Lease Control

Field	Value
Subtitle	Lifecycle Binding
Overview	Preventing port collisions across local services.
Objective	Bind local TCP ports to explicit process IDs and session metadata during multi-agent runs.

Phase 01 / Allocation

1. Lease Request

Reserve an available port range for a named agent task and project.

Tools: Port Lease Registry

2. PID Binding

Record the owning process ID and workspace path against the active lease.

Tools: Process Tracker

Phase 02 / Teardown

1. Explicit Release

Terminate leased child processes and free registered ports on session exit.

Tools: Lifecycle Reaper

Traffic Aggregation

Field	Value
Subtitle	Packet Telemetry
Overview	Diagnosing local TCP and UDP failures in real time.
Objective	Aggregate live local traffic into time-windowed summaries for fast agent diagnostics.

Phase 01 / Capture

1. Port Filtering

Filter live TCP and UDP packets to the ports registered for the active session.

Tools: Packet Sniffer

2. Window Rollup

Summarize request rates, status codes, and connection drops into sliding intervals.

Tools: Telemetry Store

Phase 02 / Diagnosis

1. Event Correlation

Match socket failures and timeouts against recent service restarts.

Tools: Telemetry Store

Orphan Process Cleanup

Field	Value
Subtitle	Listener Reclamation
Overview	Detecting and clearing stale background listeners.
Objective	Reclaim local ports held by orphaned processes whose parent agent sessions have exited.

Phase 01 / Detection

1. Registry Sweep

Compare active OS listening sockets against valid entries in the lease table.

Tools: Port Lease Registry

2. Heartbeat Check

Verify whether the owning session stream is still active.

Tools: Lifecycle Reaper

Phase 02 / Reclamation

1. Process Termination

Stop unleased orphan listeners and release their ports back to the pool.

Tools: Lifecycle Reaper

Tunnel Provisioning

Field	Value
Subtitle	Zero Trust API
Overview	Automated Cloudflare Zero Trust tunnel setup.
Objective	Expose local services securely through Cloudflare Zero Trust tunnels without manual dashboard steps.

Phase 01 / Configuration

1. Route Registration

Map a leased local port to a configured Cloudflare tunnel hostname via API.

Tools: Cloudflare Zero Trust MCP

2. Connectivity Check

Verify ingress routing and TLS termination before handing the URL back.

Tools: Health Check

Summary

Deterministic local networking.

Port Guardian gives multi-agent testbeds strict port ownership, live socket visibility, and clean teardown.

Source

URL: https://sevenswell.ai/#port-guardian