RoonCyber.AI

RUNTIME AI WORKLOAD SECURITY

Block AI threats where they actually execute.

AI agents call tools, spawn processes, use credentials and move data. RoonCyber.AI sees all of it at the kernel level, and can stop it.

THE GAP

Prompt-layer tools see the request. Runtime sees what happens next.

The most dangerous AI attacks, like instructions hidden in data an agent retrieves, look normal at the prompt. The threat shows up in what the workload does after.

GATEWAYS & PROMPT GUARDRAILS

- See prompts and responses
- Block the request
- Estimate what could be reached

ROONCYBER RUNTIME

- **See every tool call, process, file and connection**
- **Block the process, agent, credential or connection**
- **Show what was actually touched**

THE FOUNDATION

eBPF sensor
Kernel-level view of every process and call.

Your context
EDR, cloud, identity, gateways and MCP.

One execution path
The evidence to block with confidence.

RESPONSE ACTIONS

What RoonCyber can stop at runtime

Request-level tools drop a request. RoonCyber acts on the running workload.

Kill process or agent

Stop a hijacked process or rogue agent.

Block egress

Cut calls to unknown hosts before data leaves.

Block inbound IPs

Shut off the sources driving an attack.

Revoke credentials

Stop keys being used beyond their scope.

Quarantine or isolate

Freeze the pod or isolate the host.

Hold for approval

Pause risky actions until a person signs off.

YOUR COMFORT LEVEL

Recommended, approved or automated. You decide.

More human control — More automation

Recommended

We tell you what to do. Your team makes the change.

BEST FOR

Irreversible or high-risk changes

Approved

The action is staged and runs on one click.

BEST FOR

Isolating hosts, killing processes

Automated

Runs the moment a detection fires.

BEST FOR

Snapshots, alerts, blocking egress

Every detection maps to a playbook, and every action in it can be set to any of the three.

Not ready to automate? Start with recommendations.

The automation is there when you are.

BLOCKING IN ACTION

An instruction hidden in an invoice

A finance agent processes invoices from a RAG pipeline. One carries a hidden instruction.

1	**Invoice retrieved** The pipeline hands invoices to the approval agent.	Looks normal
2	**Agent plans the task** The model returns a standard plan and tool call.	Looks normal
3	**Tool response turns hostile** It tells the agent to copy customer data to an outside host.	Held for review, evidence captured
4	**Agent spawns a script** Python opens a shell and reads the customer database.	Process killed, pod frozen
5	**Data heads out the door** The script sends a large payload to a never-seen host.	Egress blocked automatically

Steps 1 and 2 looked clean. Without a runtime sensor, this attack is invisible. With it, every action is backed by an execution record that holds up to audit.