Xi-Exec

Controlled remote execution for Linux infrastructure

Xi SOFTWARE

Tuesday 17 March 2026

Contents

1 The Problem

Every organisation running Linux servers faces the same operational gap. Administrative tasks need to happen on remote hosts — restarting services, running backups, checking system health, deploying changes, managing scheduled processes. The standard answer is SSH.

SSH works. It also gives whoever is on the other end a full interactive shell. That shell can reach anything the account is permitted to access: every file, every process, every system reachable from that host. Access controls and audit logging can be layered on top, but the surface itself — the unbounded shell — remains.

When it is one trusted administrator, this is manageable. As organisations grow — delegating to junior staff, bringing in contractors, integrating automated systems, or deploying AI agents — the risks compound. A compromised credential hands an attacker everything the account can reach. A mistake by an authorised operator can have the same effect. And when something goes wrong, the audit trail is incomplete: SSH logs who connected, not what they ran.

2 What Xi-Exec Does

Xi-Exec replaces the SSH shell with a defined, controlled execution interface. A lightweight agent runs on each managed host. The controller dispatches named scripts to those agents over mutual TLS. Each agent runs only the scripts its operator has explicitly approved. Everything else is structurally unreachable — not restricted, not filtered, simply absent from what the system can do.

> The allowlist is not access control on top of a shell. It is the complete definition of what is possible.

The result is an operational boundary that is explicit, auditable, and proportionate to the risk. Operators get the capability they need. Organisations get the control they require.

3 How It Works

Xi-Exec consists of two components:

Controller (`xi-exec`) Runs on the management host. Manages the certificate authority, maintains the agent registry, and dispatches commands. Available as a CLI and a REST API. Stateless — each operation opens connections, collects results, and exits.

Agent (`xi-exec-agent`) A lightweight daemon running on each managed host. Listens on a private mTLS port. Enforces the local script allowlist. Executes scripts on request, captures output, and returns structured results. Holds no state that cannot be reconstructed from its configuration files.

A request follows a consistent path: the controller authenticates to the agent over mutual TLS; the agent validates the script name against its allowlist; the auth hook evaluates the caller's credentials and any additional policy conditions; the script executes and the result — stdout, stderr, exit code — is returned to the controller and logged on both sides with a correlated request ID.

4 Core Capabilities

Allowlist enforcement Each agent maintains a list of the scripts it is permitted to run, mapped by name to absolute paths. A script not on the allowlist cannot be invoked regardless of credentials. The scope of what is possible is explicit and operator-defined.

Mutual TLS authentication Every connection is authenticated cryptographically on both sides. No shared passwords. No SSH key distribution. Certificates are signed by a private CA created on the controller. Agents store the controller's certificate serial at pairing and verify it on every subsequent connection — a replacement certificate is rejected until all agents are updated.

Centralised access control via auth hook An auth hook is called before every execution. It receives the full request context — script name, arguments, caller identity, source IP, timestamp — and can query any external system to make its decision. LDAP directory membership, OIDC token validation, time-of-day or change window restrictions, rate limiting, and argument-level policy are all implementable as hooks. The hook is evaluated outside the agent's control and cannot be bypassed by callers.

Argument-level policy Arguments are passed to the auth hook as a structured JSON array. The hook can inspect individual argument values and apply policy at that level — permitting a script with read-only flags while blocking the same script with write flags, or restricting which database names a given token may specify.

Structured audit log Every operation produces a structured log entry on both controller and agent, with a correlated request ID, script name, arguments, caller identity, source IP, and exit code. The log is on both hosts and cannot be cleared by the operator who ran the script. Feed to a SIEM for tamper-evident compliance evidence.

Parallel fleet execution A single command dispatches to any number of agents in parallel and collects structured results. Suitable for routine fleet operations, incident response, and automated pipelines.

Automatic certificate renewal Agent certificates are renewed automatically when remaining validity falls below threshold. No manual intervention required during normal operation.

REST API with live OpenAPI spec A full HTTP API exposes all operations for integration with external systems. The live OpenAPI spec is generated dynamically, reflecting the actual scripts available on each connected agent — always accurate to the current fleet state.

Fleet discovery The discovery endpoint returns all registered agents, their available scripts, and their tags. External systems — dashboards, monitoring tools, NMS platforms — can query current fleet state without maintaining their own inventory.

Agent tags Arbitrary key-value metadata on each agent (`env=production`, `role=database`, `site=london`). Returned in discovery responses. Available for use by external tooling to target logical groups.

Wide platform support Runs on Debian, Ubuntu, Alpine Linux, and OpenWrt. Runtime dependencies are system Perl and OpenSSL — present on any Linux system worth managing. No external package registries are accessed at install or runtime.

5 Integration with Xi-Batch and Xi-Text

Xi-Exec is the remote execution transport for the Xi Software product family, providing the controlled infrastructure layer that Xi-Batch and Xi-Text operate across.

Xi-Batch Xi-Exec dispatches batch jobs to remote hosts and returns structured results. Existing network management systems can query, create, and manage batch processes through Xi-Exec's REST API without requiring direct host access. Custom APIs can be built on top of the same interface. The separation means operations teams control the execution boundary independently of the scheduling and workflow logic.

Xi-Text Xi-Exec manages Xi-Text instances remotely — triggering document generation, managing output queues, and controlling print distribution across a distributed estate. Xi-Exec can act as a secure printer transport, instructing a remote Xi-Text instance to render and deliver output without direct network printer access from the originating system.

The architecture creates a clean separation of concerns: Xi-Exec owns the secure execution channel and the infrastructure boundary; Xi-Batch and Xi-Text own the workflow and content logic. Changes to either layer do not require changes to the other.

6 Use Cases

6.1 Infrastructure Operations

Privileged access management without enterprise tooling The allowlist is the hard gate. Operators run named, pre-approved scripts. No shell. No lateral movement. The auth hook maps identities to permitted operations. Cost is zero beyond infrastructure already running — no additional platform, no dedicated PAM team.

Compliance evidence for audit frameworks PCI-DSS, ISO 27001, SOC 2, Cyber Essentials, and similar frameworks require attributable, complete records of privileged operations. Xi-Exec provides structured, tamper-evident logs on both the controller and agent that satisfy audit requirements. The auth hook can add further context. SSH logs who connected; Xi-Exec logs exactly what was done.

Third-party and contractor access A contractor receives a token scoped to specific scripts on specific hosts. When the engagement ends, the token is invalidated — no key rotation, no account deletion across multiple systems. The audit log shows exactly what was run and when. The contractor never has shell access and cannot enumerate what else is on the host.

Incident response A single command runs a remediation script across all affected hosts in parallel. Results — exit codes, output — are collected together. The operator sees immediately which hosts succeeded and which failed, without managing multiple terminal sessions. The request ID ties the incident response actions together across all hosts for the post-incident review.

Runbook automation Runbook steps become named scripts on each agent. The Xi-Exec CLI or API executes them in sequence or in parallel, with structured results. No orchestration platform required. Integrates with existing monitoring, ticketing, and pipeline tooling via the REST API.

Replacing cron-driven SSH scripts The Xi-Exec API is a direct replacement for fragile cron-to-SSH patterns. Structured output, reliable exit codes, and automatic certificate management replace the ad-hoc shell scripts that accumulate unmanaged in most infrastructure estates.

6.2 Deployed Asset Management

Organisations managing deployed Linux-based assets — EV chargers, ATMs, digital signage, industrial monitoring, vending equipment — face a specific operational challenge. The assets run a defined set of operations. They are deployed in third-party locations. Physical access is expensive or slow. The people responsible for keeping them running are not sysadmins.

Xi-Exec is well suited to this profile: a small, explicit set of named operations, executable remotely, with structured results and a full audit trail, across a fleet of devices that would otherwise require individual SSH sessions or a proprietary vendor management platform.

> A network operator with 300 deployed units receives an alert that 18 devices at four sites are showing a fault state. Without Xi-Exec: field service scheduling, site visits, manual resolution — elapsed time measured in days. With Xi-Exec: run the remediation script across the 18 units in parallel, confirm structured success responses, close the incident — elapsed time measured in minutes.

The business case is direct and quantifiable. A field service visit typically costs £150–£400. For a fleet where Xi-Exec can resolve a proportion of incidents remotely, the saving is measurable in the first year of deployment.

6.3 Agentic AI Operations

AI agent platforms operating against live infrastructure — OpenClaw, NanoClaw, and similar — give agents broad access to the systems they manage. This creates a well-documented risk: a manipulated agent, or one acting beyond its intended scope, has the same access as a legitimate operator.

Xi-Exec is the execution boundary for agentic deployments. The agent calls Xi-Exec rather than SSH or bare APIs. Xi-Exec enforces the allowlist: the agent can only invoke what has been explicitly permitted. The auth hook can enforce time-of-day restrictions, rate limits, and change window controls — conditions the agent cannot override regardless of what instructions it has received. Every action is logged.

This is the architecture that separates intelligent orchestration from controlled execution. The agent decides what to do; Xi-Exec controls whether and how it is done.

7 Security Model Summary

The security model rests on three independent layers, each of which provides meaningful protection without depending on the others:

Transport identity mTLS with a private CA. Both sides authenticate by certificate. No connection is possible without a certificate signed by the fleet CA.

Execution boundary The allowlist on each agent. The set of possible operations is defined by the agent's operator, not the caller. A compromised credential cannot invoke anything outside the allowlist.

Policy enforcement The auth hook. Every invocation is evaluated against current policy before execution. Revocation, time-window restrictions, and argument-level controls are enforced in real time at the hook, without touching the remote host.

Compromise of any single layer — a stolen token, a misconfigured allowlist entry, a bypassed hook — is contained by the others. Compromise of a credential does not yield shell access. Compromise of shell access does not compromise the audit trail. The audit trail is on two hosts and cannot be altered by the party whose actions it records.

8 Deployment

Xi-Exec installs from a single installer script on Debian, Ubuntu, and Alpine Linux, and via opkg on OpenWrt. The installer checks dependencies, creates system users and directories, and sets up systemd or procd service units. A standard installation takes under ten minutes.

Agents are introduced to the controller through a pairing ceremony: the agent generates its own key pair and submits a certificate signing request; both sides display a verification code; the operator confirms the codes match and approves. No keys are copied or distributed. No files are emailed.

Docker deployment is supported with a reference Compose configuration. All state is on named volumes; containers are stateless and can be rebuilt without losing configuration or pairing.

9 Support and Licensing

Xi-Exec is available from Xi Software under a commercial support contract, with defined response times and SLA terms. Licensed distributions are available for organisations with vendor management or compliance requirements.

For licensing enquiries, support contracts, and pricing, contact Xi Software:

xisl.com/contactus · xisl.com/distributors · info@xisl.com · 033 0088 1380