INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-28 18:49 UTC

CVE-2026-25874: Unpatched Pickle Deserialization in LeRobot Exposes AI Inference Servers to Unauthenticated RCE

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0087
Type	CVE Vulnerability
CVE ID	CVE-2026-25874
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0006 (19th percentile)
Affected Products	Hugging Face LeRobot <= 0.4.3 (open-source robotics platform; PolicyServer and robot client components)
Published	2026-04-28T07:18:00
Discovery Source	Rss

Executive Summary

A critical unpatched vulnerability in Hugging Face's LeRobot robotics platform allows an unauthenticated attacker to execute arbitrary code on AI inference servers by sending a malicious network message. Organizations running LeRobot's PolicyServer or robot client components in production AI/ML environments are directly exposed until a fix ships in version 0.6.0. Because these components often run with elevated system privileges, a successful attack could result in full server compromise, manipulation of AI-controlled robotic systems, and lateral movement across connected infrastructure.

Technical Analysis

CVE-2026-25874 affects Hugging Face LeRobot versions 0.4.3 and earlier. The vulnerability chain combines three weaknesses: unsafe deserialization of attacker-controlled pickle data (CWE-502) received over gRPC channels that require no authentication (CWE-306) and transmit in cleartext (CWE-319). An unauthenticated remote attacker who can reach an exposed gRPC endpoint can send a crafted pickle payload that triggers arbitrary code execution on the server, no credentials, no prior access required. The PolicyServer and robot client components are the affected surfaces; both frequently run with elevated OS privileges in inference environments, which amplifies post-exploitation impact. CVSS base score is 9.5, meeting the critical threshold. EPSS score is 0.00062 (19th percentile) as of the item capture date; exploitation activity has not yet been widely

observed, but the attack primitive (pickle deserialization for RCE) is well understood and weaponizable. No patch is available. Remediation is planned for version 0.6.0. MITRE ATT&CK techniques relevant to exploitation and post-exploitation: T1190 (Exploit Public-Facing Application), T1059 (Command and Scripting Interpreter), T1210 (Exploitation of Remote Services), T1021 (Remote Services), T1565 (Data Manipulation), T1552 (Unsecured Credentials). NVD entry and secondary source URLs are listed in item metadata but could not be actively verified at response time; analyst should confirm directly at nvd.nist.gov.

Action Checklist

1. Step 1: Containment. Immediately identify all hosts running LeRobot 0.4.3 or earlier with PolicyServer or robot client components active. Block external and untrusted internal network access to the gRPC port used by PolicyServer (default configuration should be audited per your deployment). If internet-facing, take the service offline or place it behind a network control that blocks unauthenticated gRPC traffic until the patch is available.
2. Step 2: Detection. Query asset inventory and container registries for LeRobot package versions 0.4.3 and below. Review network flow logs for unexpected inbound connections to gRPC ports associated with LeRobot deployments. Look for anomalous process spawning from PolicyServer or robot client parent processes (e.g., unexpected shell invocations, outbound connections to external IPs initiated by the inference server process). Check SIEM for T1190 and T1059 indicators on hosts running these components.
3. Step 3: Eradication. No vendor patch is currently available. Apply compensating controls: enforce strict network-layer access control lists (ACLs) so only authorized robot clients and orchestration systems can reach the gRPC endpoint. Require mutual TLS (mTLS) on gRPC channels if your deployment supports configuration of transport security. Do not deserialize pickle data from untrusted sources; if operationally feasible, evaluate switching serialization format pending the v0.6.0 release.
4. Step 4: Recovery. After network controls are in place, validate that gRPC endpoints are no longer reachable from untrusted network segments using an internal port scan or service probe. Review process execution logs on affected hosts for signs of prior exploitation (unexpected child processes, new user accounts, modified files). Monitor inference server behavior baselines for anomalies post-containment. When LeRobot 0.6.0 is released, apply the update and verify the gRPC channel now enforces authentication before restoring full network access.
5. Step 5: Post-Incident. Conduct an inventory audit of all AI/ML inference components in your environment that use pickle serialization or unauthenticated inter-service communication. This vulnerability exposes a control gap in secure-by-default configuration for AI inference infrastructure. Add LeRobot and similar open-source robotics/AI serving frameworks to your continuous vulnerability monitoring scope. Evaluate whether your software supply chain review process for AI/ML libraries includes assessment of serialization mechanisms and authentication requirements on network-exposed endpoints.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate immediately to senior IR leadership and legal/compliance if network flow logs or process execution records show any inbound gRPC connection to PolicyServer from an unauthorized IP prior to containment, any unexpected child process spawned by the PolicyServer process, or any new accounts, cron jobs, or modified files on inference hosts — these indicators suggest confirmed exploitation of CVE-2026-25874, which on a system running with elevated privileges constitutes a full server compromise requiring breach notification assessment, particularly if the inference server had access to sensitive training data, PII, or operational technology (OT) robotics control channels.
Recovery Notes	After applying network ACLs and confirming gRPC isolation via external port scan, maintain enhanced auditd or Sysmon monitoring on all PolicyServer hosts for a minimum of 30 days, specifically watching for execve events under the LeRobot service account, new outbound connections, and filesystem changes in Python site-packages directories — pickle-based implants may have been staged pre-containment and could activate on a delay. Upon release of LeRobot v0.6.0, validate the patch by reviewing the specific commit diff for authenticated gRPC channel enforcement before upgrading production systems, and re-run nmap and process baseline comparisons immediately post-upgrade to confirm no regression. Do not restore full network access to PolicyServer until v0.6.0 is confirmed deployed, gRPC authentication is verified functional, and a clean-bill-of-health check (no anomalous processes, no unexpected files, no unauthorized accounts) has been completed on each host.
Forensic Artifacts	Python process execution records (auditd EXECVE events or Sysmon EventID 1) filtered on the PolicyServer process PID as parent — a successful pickle deserialization RCE against CVE-2026-25874 will manifest as an unexpected child process (bash, sh, curl, nc, or a Python one-liner) spawned directly by the LeRobot PolicyServer Python interpreter process | Network flow records (NetFlow, VPC Flow Logs, pcap) for TCP connections to the gRPC port (default 50051) in the 30 days preceding detection — large inbound payload sizes from unauthorized source IPs are consistent with pickle payload delivery; multiple short-duration connections from a single external IP suggest reconnaissance probing of the unauthenticated endpoint | Filesystem timeline artifacts from /tmp, /var/tmp, the LeRobot installation directory, and Python site-packages — a post-exploitation pickle payload would typically drop a persistence script, reverse shell, or SSH authorized_keys modification; use 'find / -newer -type f' or AIDE database diff to surface attacker-created files | Python interpreter memory snapshot (gcore output or LiME memory dump) from the PolicyServer process if exploitation is suspected prior to containment — malicious pickle payloads executing arbitrary code within the Python interpreter may leave injected code objects or modified built-in function references detectable in heap analysis using Volatility with a Python-aware profile | /etc/passwd, /etc/shadow, crontab files (crontab -l for all users), and systemd unit files in /etc/systemd/system/ with modification timestamps — CVE-2026-25874 grants unauthenticated RCE on a likely elevated-privilege process, making privilege persistence via new accounts, scheduled tasks, or service unit implants the highest-probability post-exploitation action an attacker would take

Per-Action IR Details

Step 1: Containment — Immediately identify all hosts running LeRobot 0.4.3 or earlier with PolicyServer or robot client components active. Block external and untrusted internal network access to the gRPC port used by PolicyServer (default configuration should be audited per your deployment). If internet-facing, take the service offline or place it behind a network control that blocks unauthenticated gRPC traffic until the patch is available.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), NIST CM-7 (Least Functionality), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 12.2 (Establish and Maintain a Secure Network Architecture) — isolate PolicyServer segment from untrusted networks

Compensating: Run 'ss -tlnp | grep python' or 'netstat -tlnp | grep python' on each suspected host to confirm gRPC port binding (commonly 50051 TCP for LeRobot PolicyServer). Apply immediate iptables DROP rule: 'iptables -I INPUT -p tcp --dport 50051 -j DROP' (adjust port per your deployment). For container environments: 'docker ps --format "{{.Names}} {{.Ports}}" | grep 50051' to locate exposed containers, then remove or modify port mappings. Use nmap from a jump host: 'nmap -p 50051 --open ' to confirm exposure scope across the environment without requiring a SIEM.

Evidence: Before isolating hosts, capture full network socket state: 'ss -tnp state established' to document any active gRPC connections to PolicyServer at time of containment. Record source IPs of any established connections to the gRPC port — these are candidate attacker IPs if exploitation preceded discovery. Preserve /proc//net/tcp and /proc//cmdline for the PolicyServer process PID before any service restart. If the host is a container, capture 'docker inspect ' output and network namespace state before isolation to preserve pre-containment configuration.

Step 2: Detection — Query asset inventory and container registries for LeRobot package versions 0.4.3 and below. Review network flow logs for unexpected inbound connections to gRPC ports associated with LeRobot deployments. Look for anomalous process spawning from PolicyServer or robot client parent processes (e.g., unexpected shell invocations, outbound connections to external IPs initiated by the inference server process). Check SIEM for T1190 and T1059 indicators on hosts running these components.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Version discovery without SIEM: run 'pip show lerobot' or 'pip3 show lerobot' on each host, or query container images with 'docker run --rm pip show lerobot'. For process spawn detection without EDR, deploy Sysmon with EventID 1 (Process Create) configured to alert on python.exe or python3 spawning sh, bash, cmd.exe, or curl as child processes — this captures the post-exploit shell spawn that pickle RCE would trigger. Use the Sigma rule equivalent: parent_image contains 'python' AND image contains ('sh','bash','curl','wget','nc'). On Linux hosts, enable auditd with: 'auditctl -a always,exit -F arch=b64 -S execve -F ppid=' to capture all child process executions from the PolicyServer parent process. Query pip package version across fleet via osquery: 'SELECT name, version FROM python_packages WHERE name = "lerobot";'

Evidence: Capture pip freeze output ('pip freeze | grep -i lerobot') from all AI/ML inference hosts before any updates to establish installed version at time of detection. Pull Python process execution history from auditd logs or Sysmon EventID 1 records filtered on PolicyServer's PID as parent — pickle deserialization RCE will surface as an unexpected child process (shell, interpreter, or network utility) spawned directly by the PolicyServer Python process. Review /var/log/syslog or journalctl output for the PolicyServer service for Python exceptions, unexpected import errors, or traceback output that may indicate a failed or probing exploit attempt. Check network flow logs (NetFlow, VPC Flow Logs, or pcap from an in-path sensor) for inbound gRPC traffic (TCP 50051 or configured port) originating from IPs outside the authorized robot client IP list — especially high-frequency or large-payload connections consistent with pickle payload delivery.

Step 3: Eradication — No vendor patch is currently available. Apply compensating controls: enforce strict network-layer access control lists (ACLs) so only authorized robot clients and orchestration systems can reach the gRPC endpoint. Require mutual TLS (mTLS) on gRPC channels if your deployment supports configuration of transport security. Do not deserialize pickle data from untrusted sources; if operationally feasible, evaluate switching serialization format pending the v0.6.0 release.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SC-8 (Transmission Confidentiality and Integrity), NIST SC-23 (Session Authenticity), NIST CM-6 (Configuration Settings), NIST CM-7 (Least Functionality), CIS 4.2 (Establish and

Maintain a Secure Configuration Process for Network Infrastructure), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Because no LeRobot patch exists as of CVE-2026-25874 disclosure, the primary compensating control is network enforcement: configure iptables or nftables with an allowlist permitting only authorized robot client IPs to reach the gRPC port ('iptables -A INPUT -p tcp --dport 50051 -s -j ACCEPT; iptables -A INPUT -p tcp --dport 50051 -j DROP'). For mTLS without enterprise PKI, generate self-signed CA and client/server certificates using openssl and configure gRPC channel credentials in LeRobot's Python gRPC initialization if the framework exposes channel security options. As a serialization-layer compensating control, wrap the PolicyServer's pickle.loads call with a HMAC-signed payload validator if code modification is operationally feasible — reject any payload whose signature does not match a pre-shared key between authorized clients and the server. Document all compensating controls as risk acceptance entries per NIST 800-61r3 §3.4 until v0.6.0 is available.

Evidence: Before applying ACLs, extract the full PolicyServer configuration file (typically a YAML or JSON config in the LeRobot deployment directory) to document the current gRPC binding address and port, authentication settings (or absence thereof), and any existing TLS configuration. If exploitation is suspected, preserve a memory snapshot of the PolicyServer process using 'gcore ' or LiME kernel module before killing and re-configuring — pickle-deserialized payloads that establish persistence may have injected code into the Python interpreter's memory space. Capture current iptables/nftables ruleset ('iptables -L -n -v') as a pre-change baseline. Check for new cron entries ('crontab -l -u '), new systemd unit files in /etc/systemd/system/, and new files in /tmp or world-writable directories — these are common post-exploitation persistence mechanisms a pickle RCE payload would deploy.

Step 4: Recovery — After network controls are in place, validate that gRPC endpoints are no longer reachable from untrusted network segments using an internal port scan or service probe. Review process execution logs on affected hosts for signs of prior exploitation (unexpected child processes, new user accounts, modified files). Monitor inference server behavior baselines for anomalies post-containment. When LeRobot 0.6.0 is released, apply the update and verify the gRPC channel now enforces authentication before restoring full network access.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST SI-2 (Flaw Remediation), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CM-3 (Configuration Change Control), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Validate network isolation from an untrusted vantage point using nmap: 'nmap -p 50051 -Pn ' from a host outside the authorized ACL — connection refused or filtered confirms the control is effective. For behavior baseline monitoring without an EDR, configure auditd to log all execve, connect, and open syscalls for the PolicyServer process user account and pipe to a local file for daily diff review. Use AIDE (Advanced Intrusion Detection Environment) to establish a filesystem integrity baseline of the LeRobot installation directory and Python site-packages immediately after containment: 'aide --init && cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db'. When v0.6.0 ships, verify the fix by reviewing the LeRobot GitHub release diff for the specific commit addressing pickle deserialization authentication and confirm 'pip show lerobot' reports version 0.6.0 or later before restoring gRPC access.

Evidence: Conduct a post-containment process ancestry review: query auditd EXECVE records or Sysmon EventID 1 logs for the 48-72 hours preceding detection, filtering on the PolicyServer process account as the UID — any child processes (bash, sh, python -c, curl, wget, nc) spawned from that account that are not part of normal LeRobot operation are indicators of prior successful exploitation. Check /etc/passwd and /etc/shadow modification timestamps, and review 'last' and 'lastlog' output for new or unexpected login activity under the service account or root. Run 'find / -newer /path/to/lerobot/install -type f -not -path "/proc/*" -not -path "/sys/*"' to surface files created or modified after LeRobot installation — attacker-dropped webshells, reverse shell scripts, or persistence implants will appear here.

Step 5: Post-Incident — Conduct an inventory audit of all AI/ML inference components in your environment that use pickle serialization or unauthenticated inter-service communication. This vulnerability exposes a control gap in secure-by-default configuration for AI inference infrastructure. Add LeRobot and similar

open-source robotics/AI serving frameworks to your continuous vulnerability monitoring scope. Evaluate whether your software supply chain review process for AI/ML libraries includes assessment of serialization mechanisms and authentication requirements on network-exposed endpoints.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity (Lessons Learned)

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST SA-11 (Developer Testing and Evaluation), NIST SI-2 (Flaw Remediation), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Identify all Python-based services in the environment that call pickle.loads or use torch.load (which internally invokes pickle) by running a grep across deployed codebases: 'grep -r "pickle.loads\|torch.load\|joblib.load\|numpy.load" /opt /srv /app --include="*.py"'. Enumerate unauthenticated inter-service gRPC or HTTP endpoints using nmap service discovery: 'nmap -p 50051,8080,8443 --open -sV ' and flag any service returning a gRPC or HTTP response without requiring client credentials. Subscribe to Hugging Face's security advisories via their GitHub Security Advisories feed (github.com/huggingface/lerobot/security/advisories) and add to your vulnerability feed aggregator or manually review weekly. Add a checklist item to your AI/ML library intake process: does this library expose a network port? Does it use pickle, joblib, or torch serialization on that port? Is mutual authentication required by default?

Evidence: Compile a full lessons-learned artifact package including: (1) timeline of CVE-2026-25874 discovery to containment with timestamps, (2) list of all LeRobot versions and hosts identified in the asset inventory audit, (3) network flow evidence showing whether unauthorized gRPC connections occurred before containment (confirming or ruling out pre-containment exploitation), (4) results of the post-containment filesystem integrity scan (AIDE output or find-newer results), and (5) the grep output of all pickle.loads usages across the AI/ML service codebase. This package satisfies NIST IR-8 (Incident Response Plan) documentation requirements and feeds directly into the lessons-learned session.

Detection Guidance

Primary detection focus is on the gRPC endpoint exposed by LeRobot's PolicyServer. Query your asset management and container registry for any deployment of huggingface/lerobot or lerobot pip package at version 0.4.3 or below. In network logs, identify hosts with open gRPC ports (commonly TCP 50051 or custom-configured) associated with LeRobot processes and flag any inbound connections from IPs outside your authorized robot client or orchestration IP ranges. On the host, monitor for anomalous child process creation where the parent is the PolicyServer process, specifically watch for shell interpreters (bash, sh, cmd, powershell), network tools (curl, wget, nc), or credential-reading utilities spawned by the inference server. In SIEM, correlate T1190 (exploit of public-facing application) alerts on AI inference hosts with subsequent T1059 (scripting interpreter execution) events. Note: in-process RCE execution may evade child process-based detection; prioritize network-layer blocking over post-exploitation detection for this vulnerability. Because pickle deserialization RCE can result in in-process execution without a new child process, also monitor for unexpected outbound network connections and file writes from the PolicyServer process itself. No public IOCs (IPs, hashes, domains) associated with active exploitation of this CVE have been identified in the item data at time of analysis.

Framework Mappings

MITRE-ATTACK

- **T1059** — Command and Scripting Interpreter

- **T1565** — Data Manipulation
- **T1552** — Unsecured Credentials
- **T1021** — Remote Services
- **T1210** — Exploitation of Remote Services
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-17** — Remote Access
- **AC-3** — Access Enforcement
- **IA-2** — Identification and Authentication (Organizational Users)
- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-8** — Transmission Confidentiality and Integrity
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A02:2021** — Cryptographic Failures
- **A08:2021** — Software and Data Integrity Failures
- **A07:2021** — Identification and Authentication Failures

CIS-V8

- **3.10** — Encrypt Sensitive Data in Transit
- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

HIPAA-SECURITY

- **164.312(e)(1)** — Transmission Security

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059	Command and Scripting Interpreter	Execution
T1565	Data Manipulation	Impact
T1552	Unsecured Credentials	Credential-Access
T1021	Remote Services	Lateral-Movement
T1210	Exploitation of Remote Services	Lateral-Movement
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/04/critical-cve-2026-25874-leaves-hu...	T3
CVE-2026-25874 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-25874	T1
CVE-2026-25874 - Ubuntu	https://ubuntu.com/security/CVE-2026-25874	T3
CVE-2026-25874 - CRITICAL Vulnerability | SOC Defenders	https://socdefenders.ai/cves/CVE-2026-25874	T3
Threats Tagged 'cve-2026-25874'	https://radar.offseq.com/threats?tag=cve-2026-25874	T3

Generated 2026-04-28 18:49 UTC by TJS Security Command Center