INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-13 16:29 UTC

Critical Marimo Python Notebook RCE Vulnerability (CVE-2026-39987) Exploited Within 10 Hours of Disclosure

CVE VULNERABILITY | **CRITICAL** | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0034
Type	CVE Vulnerability
CVE ID	CVE-2026-39987
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0270 (86th percentile)
Affected Products	Marimo (open-source reactive Python notebook), specific version range not confirmed from available sources
Published	1 day ago
Discovery Source	Serper

Executive Summary

A critical pre-authentication remote code execution vulnerability (CVE-2026-39987) in Marimo, an open-source reactive Python notebook platform, was actively exploited in the wild within approximately 10 hours of public disclosure. No credentials or user interaction are required to exploit the flaw, meaning any internet-exposed Marimo instance is at immediate risk of full system compromise. Organizations running Marimo for data science, ML, or AI development workflows should treat this as an emergency response item.

Technical Analysis

CVE-2026-39987 is a critical pre-authentication RCE vulnerability in Marimo, an open-source reactive Python notebook. The vulnerability is classified under CWE-306 (Missing Authentication for Critical Function) and CWE-94 (Improper Control of Code Generation), indicating the flaw allows unauthenticated attackers to inject and execute arbitrary Python code on the host system in a single request. MITRE ATT&CK mapping covers T1190 (Exploit Public-Facing Application) for initial access and T1059.006 (Command and Scripting Interpreter: Python) for execution. CVSS base score is 9.8 (Critical). EPSS score is 0.027 at the 85.9th percentile, indicating elevated exploitation probability relative to the broader CVE population. Exploitation in the wild was observed within approximately 10 hours of public disclosure per reporting from Sysdig and The Hacker News.

IMPORTANT CONFIDENCE NOTE: Authoritative NVD and OSV records were not available in the source data at time of writing. Specific affected version ranges have not been confirmed from primary sources. Technical details in this summary are derived from vendor advisory reporting and security research blogs (Sysdig, Endor Labs original analysis; Rescana, SecurityAffairs, The Hacker News secondary aggregation). Verify against NVD (https://nvd.nist.gov/vuln/detail/CVE-2026-39987) and the official Marimo advisory before taking patch action. CISA KEV inclusion: not confirmed as of configuration date.

Action Checklist

1. Step 1: Containment, Immediately identify all Marimo instances in your environment. Take internet-exposed Marimo servers offline or place them behind a firewall rule blocking external access. Marimo notebooks are often run locally or on internal infrastructure, but any instance bound to a public or shared network interface is at immediate risk. Do not wait for patch confirmation before restricting access.
2. Step 2: Detection, Check for unexpected process spawning from the Marimo server process (e.g., shell children of the Marimo Python process). Review web server or application logs for anomalous POST requests to Marimo endpoints, particularly requests that require no session token or authentication header. Look for unexpected outbound connections from Marimo host systems. Sysdig's blog post on this CVE may contain behavioral indicators, validate against that source directly.
3. Step 3: Eradication, Apply the patch or upgrade to the fixed Marimo version. Check https://github.com/marimo-team/marimo/releases and https://nvd.nist.gov/vuln/detail/CVE-2026-39987 for the confirmed remediation version. Do not rely on secondary reporting alone for patch version.
4. Step 4: Recovery, After patching, verify the Marimo process is running the updated version. Audit host systems for indicators of prior compromise: new user accounts, modified cron jobs, dropped files, or persistence mechanisms. Monitor outbound network traffic from Marimo hosts for 7 days post-remediation. Restore affected systems from known-clean backups if compromise is suspected.
5. Step 5: Post-Incident, Review your inventory process for open-source developer tooling. Marimo and similar notebook platforms (Jupyter, etc.) are frequently deployed without formal change control. Establish a policy requiring network isolation or authentication enforcement for any notebook or REPL-style tool. Map this vulnerability to NIST CSF ID.AM-2 (software asset inventory) and PR.AC-3 (remote access management) as control gaps.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO and legal/privacy counsel immediately if forensic artifacts indicate successful pre-authentication RCE (child process spawning, new accounts, dropped files, or outbound C2 connections) on any Marimo host with access to sensitive data, ML model weights, API keys, or cloud credentials stored in notebooks or environment variables, as this may trigger breach notification obligations under applicable regulations.

Recovery Notes	Before returning any Marimo instance to service, confirm the patched version via 'pip show marimo' and validate that the instance is bound only to loopback or an authenticated reverse proxy — not directly to a public interface. Monitor all Marimo hosts for 7 days post-remediation using Sysmon Event ID 1 (Process Creation) and Event ID 3 (Network Connection) filters scoped to the Marimo parent process, and review outbound connection logs daily for beaconing patterns (regular interval connections to external IPs) that would indicate a persistent implant survived patching. If any host cannot be forensically cleared, treat it as compromised and rebuild from a known-clean image rather than patching in place.
Forensic Artifacts	Marimo/uvicorn HTTP access logs: POST requests to Marimo kernel execution API endpoints (e.g., /api/kernel/run, /api/kernel/instantiate) originating from external IPs with no session token or authentication header — the pre-auth nature of CVE-2026-39987 means legitimate exploitation requests will be structurally indistinguishable from unauthenticated API calls except by source IP and payload content | Sysmon Event ID 1 (Process Creation) records: child processes (sh, bash, curl, wget, python -c, nc, ncat) where ParentImage or ParentCommandLine references the Marimo or uvicorn process — this is the primary indicator of successful RCE payload execution via CVE-2026-39987 | Sysmon Event ID 3 (Network Connection) records: outbound TCP connections initiated by the Marimo/Python process to external IPs on non-standard ports, particularly shortly after inbound POST requests to the Marimo API — indicative of a reverse shell or C2 callback following exploit success | File system artifacts in /tmp, /var/tmp, and the Marimo working directory: dropped scripts, compiled ELF binaries, or Base64-encoded payloads written by the exploiting process, identifiable by 'find /tmp /var/tmp -newer -type f' and cross-referenced against the exploitation window | Python environment modifications: unauthorized entries in sitecustomize.py, usercustomize.py, or .pth files in the Marimo Python environment's site-packages directory, which an attacker with RCE could use to establish persistence that survives a Marimo version upgrade without a full environment rebuild

Per-Action IR Details

Step 1: Containment — Immediately identify all Marimo instances in your environment. Take internet-exposed Marimo servers offline or place them behind a firewall rule blocking external access. Marimo notebooks are often run locally or on internal infrastructure, but any instance bound to a public or shared network interface is at immediate risk. Do not wait for patch confirmation before restricting access.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected systems to prevent further exploitation while preserving evidence

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run 'netstat -tlnp | grep python' or 'ss -tlnp | grep marimo' on Linux hosts to identify Marimo processes bound to 0.0.0.0 or external interfaces (default port 2718). On Windows, use 'netstat -ano | findstr :2718'. Apply an immediate host firewall rule: 'iptables -I INPUT -p tcp --dport 2718 -j DROP' (Linux) or equivalent Windows Defender Firewall inbound block rule. Use nmap from a jump host to scan your /16 for open port 2718 to find unregistered instances: 'nmap -p 2718 --open 10.0.0.0/16'.

Evidence: Before isolation, capture a full network connection state snapshot ('ss -tlnp > /tmp/marimo_connections_$(date +%s).txt'), the running process tree ('ps auxf > /tmp/process_tree_$(date +%s).txt'), and netflow or firewall logs showing all inbound connections to port 2718 for the prior 24 hours. Record the Marimo process PID and its current working directory ('ls -la /proc//cwd') before taking the instance offline. This preserves evidence of any pre-containment exploitation activity.

Step 2: Detection — Check for unexpected process spawning from the Marimo server process (e.g., shell children of the Marimo Python process). Review web server or application logs for anomalous POST requests to Marimo endpoints, particularly requests that require no session token or authentication header. Look for unexpected outbound connections from Marimo host systems. Sysdig's blog post on this CVE may contain behavioral indicators — validate against that source directly.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate log sources, identify attack scope, and characterize exploitation activity

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysmon (configuration with ProcessCreate event ID 1) and filter for processes where ParentImage matches the Marimo Python executable path (e.g., '*marimo*' or '*uvicorn*'). Sysmon Event ID 1 with ParentCommandLine containing 'marimo' spawning cmd.exe, sh, bash, curl, wget, or python -c is a high-confidence exploitation indicator. For web log analysis, grep Marimo access logs for POST requests lacking Authorization or Cookie headers: 'grep "POST" access.log | grep -v "Authorization\|Cookie"'. Use Wireshark or tcpdump to capture outbound traffic from the Marimo host: 'tcpdump -i eth0 -w /tmp/marimo_capture.pcap not dst net 10.0.0.0/8'. For process ancestry on Windows, query Sysmon logs via PowerShell: 'Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" | Where-Object {$_.Message -match "marimo" -and $_.Id -eq 1}'.

Evidence: Capture Marimo application logs (typically in the working directory or '$HOME/.marimo/logs/'), web access logs showing requests to Marimo's HTTP API endpoints (POST /api/kernel/run or equivalent kernel execution endpoints), Sysmon Event ID 1 (Process Creation) records showing child processes of the Marimo/uvicorn parent, Sysmon Event ID 3 (Network Connection) records for outbound connections initiated by the Marimo process, and bash/zsh history files for any accounts active on the host during the exploitation window ('cat /home/*/.bash_history', 'cat /root/.bash_history').

Step 3: Eradication — Apply the patch or upgrade to the fixed Marimo version once confirmed in the official Marimo GitHub advisory or NVD record. Specific version upgrade path was not available in source data at time of writing — check https://github.com/marimo-team/marimo/releases and https://nvd.nist.gov/vuln/detail/CVE-2026-39987 for the confirmed remediation version. Do not rely on secondary reporting alone for patch version.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the vulnerability and any attacker-introduced artifacts before returning systems to service

Controls: NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-6 (Configuration Settings), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Before patching, verify the current installed version with 'pip show marimo | grep Version' and record it as baseline evidence. Upgrade using 'pip install --upgrade marimo' only after confirming the fixed version number from the official GitHub advisory at https://github.com/marimo-team/marimo/releases (search-retrieved URL — validate before use). Pin the confirmed safe version in requirements.txt or pyproject.toml to prevent accidental rollback: 'marimo>='. After upgrade, re-run 'pip show marimo' to confirm the version change. If the host is suspected compromised, do not patch in place — rebuild from a known-clean image and redeploy Marimo at the patched version.

Evidence: Before patching, capture a pip freeze snapshot ('pip freeze > /tmp/pip_freeze_pre_patch_$(date +%s).txt') to document the vulnerable environment state. Collect any webshells or dropped files in the Marimo working directory, temp directories ('/tmp', '/var/tmp'), and any directories writable by the Marimo process user. Hash all collected artifacts with SHA-256 ('sha256sum ') before removal. Check for unauthorized additions to requirements.txt or pyproject.toml that may indicate supply-chain persistence introduced during exploitation.

Step 4: Recovery — After patching, verify the Marimo process is running the updated version. Audit host systems for indicators of prior compromise: new user accounts, modified cron jobs, dropped files, or

persistence mechanisms. Monitor outbound network traffic from Marimo hosts for 7 days post-remediation. Restore affected systems from known-clean backups if compromise is suspected.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore systems to normal operation and verify the threat has been fully eliminated before returning to production

Controls: NIST IR-4 (Incident Handling), NIST CP-10 (System Recovery and Reconstitution), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-6 (Security and Privacy Function Verification), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

Compensating: Audit for persistence mechanisms specific to Python/notebook environments: check crontab entries ('crontab -l -u marimo; crontab -l -u root'), systemd timers ('systemctl list-timers'), new accounts added after the exploitation window ('awk -F: "$3 >= 1000" /etc/passwd' and compare against a known-good baseline), and Python sitecustomize.py or usercustomize.py modifications that could enable persistent code execution. Check for PYTHONSTARTUP, PYTHONPATH, and LD_PRELOAD environment variable manipulation in /etc/environment, /etc/profile.d/, and the Marimo service unit file. Use osquery to enumerate recently modified files: 'SELECT * FROM file WHERE directory = "/tmp" AND mtime > (strftime("%s","now") - 86400);'. For 7-day outbound monitoring without a SIEM, deploy tcpdump as a cron job logging to a remote syslog server.

Evidence: Collect '/etc/passwd' and '/etc/shadow' diffs against pre-incident baseline, full crontab listings for all users, contents of '/etc/profile.d/' and any '.bashrc'/'.profile' modifications, a recursive listing of files modified within 48 hours of the exploitation window ('find / -newer /tmp/marimo_connections_.txt -not -path "/proc/*" -not -path "/sys/*" 2>/dev/null'), and a memory dump if the Marimo process was still running at detection time (use 'avml' or 'LiME' on Linux for volatile memory acquisition before process termination).

Step 5: Post-Incident — Review your inventory process for open-source developer tooling. Marimo and similar notebook platforms (Jupyter, etc.) are frequently deployed without formal change control. Establish a policy requiring network isolation or authentication enforcement for any notebook or REPL-style tool. Map this vulnerability to NIST CSF ID.AM-2 (software asset inventory) and PR.AC-3 (remote access management) as control gaps.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned review, update detection capabilities, and improve controls to prevent recurrence

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 6.3 (Require MFA for Externally-Exposed Applications)

Compensating: Create a recurring osquery scheduled query to detect Marimo, Jupyter, Streamlit, and similar notebook processes bound to non-loopback interfaces: 'SELECT p.pid, p.name, p.cmdline, l.address, l.port FROM processes p JOIN listening_ports l ON p.pid = l.pid WHERE l.address != "127.0.0.1" AND l.address != "::1" AND (p.name LIKE "%marimo%" OR p.name LIKE "%jupyter%" OR p.name LIKE "%uvicorn%");'. Write a Sigma rule detecting child process spawning from Python notebook server processes (ParentImage matching uvicorn/marimo, ChildImage matching sh/bash/cmd/powershell) and deploy it against Sysmon logs. Subscribe to the Marimo GitHub repository security advisories (Watch > Security alerts) to reduce future detection lag below the 10-hour exploitation window observed in this incident.

Evidence: Compile the full incident timeline documenting: the Marimo version deployed, the date it was installed, whether it appeared in any software inventory prior to this incident, which network interfaces it was bound to, and how it was discovered (proactive scan vs. alert vs. external notification). This timeline directly evidences the ID.AM-2 and PR.AC-3 control gaps identified in the step and should feed the formal lessons-learned report required under NIST IR-8 (Incident Response Plan) plan maintenance obligations.

Detection Guidance

Detection specifics are limited by the secondary source quality of available reporting at time of writing. Based on the CWE classification (CWE-306, CWE-94) and ATT&CK techniques (T1190, T1059.006), monitor for: (1) unauthenticated HTTP requests to Marimo server endpoints, particularly those triggering cell execution without a valid session; (2) Python interpreter spawning unexpected child processes (shell, curl, wget, nc) from the Marimo process parent; (3) outbound connections from Marimo host systems to external IPs, especially on non-standard ports; (4) new files written to disk by the Marimo process outside expected notebook directories. For SIEM: correlate process creation events (Sysmon Event ID 1 on Windows; auditd execve on Linux) with the Marimo process as parent. Consult Sysdig's and Endor Labs' published writeups directly before operationalizing any detection rules in your SIEM, both contain request-level and behavioral IOC details not available in aggregated reporting.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://nvd.nist.gov/vuln/detail/CVE-2026-39987`	NVD authoritative record for CVE-2026-39987 — check for confirmed affected versions, CVSS vector, and patch details	HIGH
URL	`https://github.com/marimo-team/marimo/releases`	Official Marimo release page — source for confirmed patched version; validate before applying	HIGH

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1059.006** — Python

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **IA-2** — Identification and Authentication (Organizational Users)
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures
- **A03:2021** — Injection

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **16.10** — Apply Secure Design Principles in Application Architectures

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

ISO-27001-2022

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

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1059.006	Python	Execution

Sources

Source	URL	Tier
	https://www.rescana.com/post/critical-marimo-python-notebook-rce-vu...	T3
(consolidated)	https://securityaffairs.com/190623/hacking/cve-2026-39987-marimo-rc...	T3
Marimo RCE Flaw CVE-2026-39987 Exploited Within 10 Hours of ...	https://thehackernews.com/2026/04/marimo-rce-flaw-cve-2026-39987.html	T3
Root in One Request: Marimo's Critical Pre-Auth RCE (CVE-2026 ...	https://www.endorlabs.com/learn/root-in-one-request-marimos-critica...	T3
Marimo OSS Python Notebook RCE: From Disclosure to Exploitation ...	https://www.sysdig.com/blog/marimo-oss-python-notebook-rce-from-dis...	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-39987	T1

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-04-13 16:29 UTC by TJS Security Command Center