INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:37 UTC

Critical Unpatched RCE Vulnerability in GNU InetUtils Telnetd (CVE-2026-32746)

CVE VULNERABILITY | CRITICAL | CVSS 9.8

SCC Item ID	SCC-CVE-2026-0017
Type	CVE Vulnerability
CVE ID	CVE-2026-32746
Severity	CRITICAL
CVSS Base Score	9.8
EPSS Score	0.0006 (18th percentile)
Affected Products	GNU InetUtils telnetd, all versions (unpatched as of reporting date 2026-03-04)
Published	2026-03-18

Executive Summary

A critical, unpatched remote code execution vulnerability (CVE-2026-32746) in GNU InetUtils telnetd allows unauthenticated attackers to execute arbitrary code with elevated privileges over TCP port 23. All versions of GNU InetUtils telnetd are affected, and no patch is available as of March 4, 2026. Any organization running telnetd on internet-facing or internally accessible systems faces immediate risk of full system compromise without requiring credentials. Note: CVSS 9.8 score and CWE classifications are estimated from secondary sources; NVD record was not confirmed accessible at analysis time.

Technical Analysis

CVE-2026-32746 affects GNU InetUtils telnetd, all versions, unpatched as of 2026-03-04. The vulnerability permits unauthenticated remote code execution via TCP port 23 with elevated privileges. The precise root cause has not been confirmed from primary sources in this session. CWE assignments are estimated based on reported impact class: CWE-119 (improper restriction of memory buffer operations) and CWE-787 (out-of-bounds write) are consistent with unauthenticated RCE in legacy network daemons; CWE-284 (improper access control) aligns with the privilege escalation component. CVSS base score of 9.8 is inferred from impact description, not confirmed from NVD. EPSS score is 0.00057 (17.7th percentile) as of analysis date, reflecting low observed exploitation activity but not discounting the critical impact. MITRE ATT&CK relevance: T1190 (Exploit Public-Facing Application) is the primary initial access vector; post-exploitation may involve T1059 (Command and Scripting Interpreter) for shell access. Not listed in CISA KEV as of analysis date. Secondary source URLs from The Hacker News and Security Affairs were identified but not actively fetched and verified in

this session, treat source-derived specifics with medium confidence until NVD record is confirmed.

Action Checklist

1. Step 1, Immediate containment: Disable telnetd on all systems where it is running. If telnetd cannot be disabled, block TCP port 23 at the perimeter firewall and any internal network boundaries. Prefer disabling the service entirely; firewall rules alone do not eliminate risk from internal exposure.
2. Step 2, Inventory: Identify all systems running GNU InetUtils telnetd across your environment. Query configuration management databases, run authenticated network scans for port 23 (open and listening), and cross-reference with asset inventory. Include cloud, OT, and legacy infrastructure.
3. Step 3, Detection sweep: Review firewall and network logs for inbound and outbound TCP port 23 traffic over the past 30 days. Flag any sessions originating from external IPs or unexpected internal hosts. Check for anomalous process spawning from telnetd parent processes.
4. Step 4, Patch monitoring: No patch is available as of 2026-03-04. Monitor the GNU InetUtils project advisory page and NVD entry (https://nvd.nist.gov/vuln/detail/CVE-2026-32746) for patch availability. Establish a patch deployment SLA before a fix is released so response is immediate upon availability.
5. Step 5, Communication and long-term controls: Notify relevant stakeholders of exposure status and containment actions taken. Use this event to drive a formal policy deprecating telnetd in favor of SSH across all systems. Document any systems where telnetd cannot be immediately removed and apply compensating controls with a defined remediation deadline.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	If any system running telnetd is internet-facing or handles sensitive data (PII, credentials, patient records, financial data), or if Step 3 detects exploitation attempts or unauthorized port 23 connections, escalate immediately to CISO and consider engaging external IR firm for forensic analysis and containment verification.
Recovery Notes	After containment is confirmed on all systems (Step 1 complete, Step 3 sweep shows no active exploitation), conduct a forensic review of any systems that had port 23 accessible for more than 48 hours to rule out prior compromise. Restore telnetd-free baselines to affected systems and perform integrity verification (file checksums, process baseline comparison). Re-enable monitoring for port 23 access with alerting set to 'critical' and configure automated response rules (alert + auto-block source IP) for any future port 23 connection attempts.
Forensic Artifacts	Firewall/NetFlow logs (inbound and outbound TCP port 23 traffic, 30-day lookback) | Process accounting logs (/var/log/account/pacct on Linux, or Get-EventLog Security on Windows showing 4688 Process Creation events) | User authentication logs (/var/log/auth.log, /var/log/secure on Linux; Security Event Log 4624/4625 on Windows) | Network packet captures (.pcap files from TCP port 23 sessions, including payload for command analysis) | Syslog and auditd logs (auditctl -m EXECVE for child processes spawned from telnetd parent, /var/log/messages for service start/stop events)

Per-Action IR Details

Step 1, Immediate containment: Disable telnetd on all systems where it is running. If telnetd cannot be disabled, block TCP port 23 at the perimeter firewall and any internal network boundaries. Prefer disabling the service entirely; firewall rules alone do not eliminate risk from internal exposure.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.2.3

Controls: NIST IR-4(1) — Incident Handling Implementation, CIS 6.1 — Establish network segmentation, NIST AC-2(7) — Privileged access management

Compensating: On Linux: `systemctl disable inetd` or `systemctl disable telnetd`; remove telnetd from `/etc/inetd.conf` (comment out the telnet line). On systems without systemd, edit `/etc/services` to comment the telnet line, then restart inetd via `killall -HUP inetd`. Verify with `netstat -tlnp | grep :23` (should return empty). For Windows: `sc config telnet start=disabled && sc stop telnet`. At firewall: add ACL rule `deny tcp any any eq 23` or equivalent in your firewall CLI/UI, test with `telnet localhost 23` from local workstation (should timeout or refuse).

Evidence: Before disabling: (1) Run `netstat -tlnp` or `ss -tlnp` and capture full output showing all listening ports and process IDs. (2) Run `ps aux | grep -i telnet` and capture process names, user context, and command-line arguments. (3) Check `/proc/[PID]/cmdline` for each telnetd process to confirm full invocation. (4) Verify service configuration state with `systemctl status telnetd` or `chkconfig telnetd` output. (5) Capture firewall rule baseline with `iptables -L -n -v` (Linux) or `netsh advfirewall show allprofiles` (Windows) before rule additions.

Step 2, Inventory: Identify all systems running GNU InetUtils telnetd across your environment. Query configuration management databases, run authenticated network scans for port 23 (open and listening), and cross-reference with asset inventory. Include cloud, OT, and legacy infrastructure.

NIST Phase: Preparation

Reference: NIST 800-61r3 §3.1.1

Controls: NIST CM-2 — Baseline configuration, CIS 1.1 — Asset management, NIST IA-4 — Identifier management

Compensating: Use free/open-source tools: (1) Nmap for port enumeration: `nmap -p 23 --open -Pn ` for each subnet; save output with `-oX results.xml`. (2) Grep system configs manually: `grep -r telnet /etc/inetd.conf /etc/xinetd.d/ 2>/dev/null` on each Linux host (use `for` loop over SSH targets). (3) Windows: `Get-Service | Where-Object {$_.Name -like '*telnet*'}` via PowerShell across WinRM-enabled hosts. (4) Cross-reference with `/etc/services` entries. (5) Create CSV: hostname, OS, port 23 status (open/closed/filtered), last scanned date. Store in shared drive with read-only access for audit.

Evidence: Before scanning: (1) Document baseline network topology and subnets in a diagram or text file. (2) Export current CMDB or asset inventory to CSV (hostname, IP, OS, last patched date, criticality). (3) Capture current firewall rules with `iptables-save > fw_baseline.txt` (Linux) or GPO export (Windows). (4) Document known systems where telnetd is expected (if any) and their business justification. (5) Take a baseline port 23 availability scan one week prior to distribute to stakeholders as a reference point.

Step 3, Detection sweep: Review firewall and network logs for inbound and outbound TCP port 23 traffic over the past 30 days. Flag any sessions originating from external IPs or unexpected internal hosts. Check for anomalous process spawning from telnetd parent processes.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2.1

Controls: NIST SI-4 — Information system monitoring, CIS 8.6 — Centralized logging, NIST AU-2 — Audit events

Compensating: Without SIEM: (1) Export firewall logs (vendor-specific format) and parse with grep/awk: `grep ':23' firewall.log | grep -E 'ACCEPT|ALLOW' > port23_traffic.txt`. (2) For Linux auditd: `ausearch -m NETFILTER_PKT -ts 30-days-ago | grep 'dport=23' > audit_port23.log`. (3) Check process accounting on systems with telnetd: `lastcomm | grep telnetd` for spawned processes. (4) Use `tcpdump` on network tap or span port to capture live traffic: `tcpdump -i eth0 'tcp port 23' -w port23_capture.pcap`. (5) Check auth logs for telnet login attempts: `grep -i telnet /var/log/auth.log /var/log/secure | head -50`.

Evidence: Preserve before analysis: (1) Full packet captures from tap/span port for any port 23 sessions: save `.pcap` files with timestamps. (2) Firewall logs with full headers (source IP, dest IP, port, action, timestamp, user/account if

available) from the past 30 days; export to immutable storage (write-once, read-many). (3) Process accounting logs (`/var/log/account/pacct` on Linux). (4) User login logs (`/var/log/wtmp`, `/var/log/btmp` on Linux; Security Event Log 4624/4625 on Windows). (5) Any proxy or application logs that may show telnet-over-HTTP tunneling attempts. Hash all log files with SHA-256 for integrity verification.

Step 4, Patch monitoring: No patch is available as of 2026-03-04. Monitor the GNU InetUtils project advisory page and NVD entry (https://nvd.nist.gov/vuln/detail/CVE-2026-32746) for patch availability. Establish a patch deployment SLA before a fix is released so response is immediate upon availability.

NIST Phase: Preparation

Reference: NIST 800-61r3 §3.1.3

Controls: NIST SI-2 — Software and firmware updates, CIS 2.3 — Address unauthorized software, NIST SC-3 — Security testing and evaluation

Compensating: Set up free monitoring: (1) Subscribe to NVD RSS feed (https://nvd.nist.gov/feeds/json/cve/1.1/nvdCve-1.1-recent.json) and filter locally for CVE-2026-32746 updates using a cron script with curl + jq. (2) Monitor GNU InetUtils GitHub releases page (https://github.com/inetutils/inetutils/releases) with weekly manual check or free GitHub Actions notification. (3) Create a shared Google Sheet tracking 'Patch Status' with columns: CVE ID, Vendor, Expected Release Date, Actual Release Date, Testing Completion Date, Deployment Date. (4) Establish SLA: '0-24 hours from public patch release to deployment on test systems; 1-5 business days to production.' Communicate SLA to leadership in writing. (5) Assign a single owner (SOC manager or patch team lead) to verify patch release and trigger deployment workflow.

Evidence: Before patch release: (1) Document current GNU InetUtils version on all affected systems: `telnetd --version` or `telnet -v`. (2) Create a pre-patch snapshot of system state (disk image or filesystem hash) for comparison post-patch. (3) Document current firewall rules and network ACLs that will be modified if telnetd is re-enabled post-patch. (4) Establish a change control request template and approval chain. (5) Record baseline performance metrics (CPU, memory, network latency) on test systems to compare post-patch and detect regressions.

Step 5, Communication and long-term controls: Notify relevant stakeholders of exposure status and containment actions taken. Use this event to drive a formal policy deprecating telnetd in favor of SSH across all systems. Document any systems where telnetd cannot be immediately removed and apply compensating controls with a defined remediation deadline.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §3.4

Controls: NIST IR-6 — Incident reporting, CIS 18.1 — Third-party software updates, NIST CA-7 — Continuous monitoring, NIST CP-3 — Contingency planning

Compensating: Create a formal remediation plan: (1) Draft a 'Telnetd Deprecation Policy' stating telnetd is prohibited except where documented business justification exists (e.g., OT systems). (2) For each exception system, implement mandatory compensating controls: restrict SSH port 22 to specific source IPs only via `iptables -A INPUT -p tcp --dport 22 -s -j ACCEPT`; log all telnetd sessions to syslog with `auditctl -w /usr/sbin/telnetd -p x`; enforce session timeout with PAM config in `/etc/pam.d/login`. (3) Maintain an 'Exceptions Tracking' spreadsheet with columns: Hostname, Justification, Control Owner, Review Date (quarterly), Target Remediation Date. (4) Document and publish remediation steps on a shared wiki with a 'Last Updated' timestamp. (5) Schedule monthly review meeting with asset owners to track progress.

Evidence: Before communication: (1) Compile incident summary report with timeline, affected systems count, exposure duration if known, containment actions taken, and residual risk. (2) Create a 'Current State' inventory showing systems with telnetd disabled vs. systems with compensating controls vs. systems where telnetd remains active. (3) Document any detected exploitation attempts (from Step 3 evidence) to include in briefing. (4) Prepare risk assessment matrix: 'Systems with no telnetd and no SSH exposure = Green, Systems with compensating controls = Yellow, Systems with uncontrolled telnetd = Red.' (5) Preserve all stakeholder communications (emails, meeting minutes) in incident folder for post-incident review.

Detection Guidance

Primary detection focus: identify systems with telnetd running and listening on TCP port 23, then look for anomalous activity. Specific guidance: (1) Network scan, run an authenticated scan (e.g., nmap -sV -p 23) to confirm which hosts have port 23 open and which process is listening. Filter for GNU InetUtils telnetd specifically in banner responses. (2) Firewall and NetFlow logs, query for any TCP/23 connections in the past 30 days. Flag inbound connections from external IP space and any outbound connections that could indicate reverse shell activity. (3) System process logs, on Linux hosts, check for telnetd processes: ps aux | grep telnetd. Review /var/log/auth.log or equivalent for telnet session establishment events. (4) Behavioral indicators, look for child processes spawned from telnetd (e.g., /bin/sh, /bin/bash launched with telnetd as parent), which would indicate successful exploitation and shell access consistent with T1059. (5) SIEM query pattern (generic): event_source=syslog AND process_name=telnetd AND child_process IN (/bin/sh, /bin/bash, /bin/dash). Note: no confirmed IOCs (hashes, IPs, or signatures) associated with active exploitation of this CVE have been identified in available sources at time of analysis.

Framework Mappings

MITRE-ATTACK

- **T1059** — Command and Scripting Interpreter
- **T1078** — Valid Accounts
- **T1190** — Exploit Public-Facing Application

CIS-V8

- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

ISO-27001-2022

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

NIST-800-53R5

- **AC-6** — Least Privilege

SOC2-TSC

- **CC6.3** — Authorizes, modifies, or removes access

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059	Command and Scripting Interpreter	Execution
T1078	Valid Accounts	Defense-Evasion
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
	https://securityaffairs.com/189620/hacking/researchers-warn-of-unpa...	T3
Critical Unpatched Telnetd Flaw (CVE-2026-32746) Enables ...	https://thehackernews.com/2026/03/critical-telnetd-flaw-cve-2026-32...	T3
An unpatched critical telnetd bug (CVE-2026-32746) lets attackers ...	https://www.facebook.com/thehackernews/posts/%EF%B8%8F-warning-an-u...	T3
Critical Telnetd Flaw Enables Unauthenticated RCE via Port 23	https://www.reddit.com/r/pwnhub/comments/1rwzf8p/critical_telnetd_f...	T3
Researchers warn of unpatched, critical Telnetd flaw affecting all ...	https://community.opentextcybersecurity.com/vulnerability-vault-228...	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-32746	T1

Generated 2026-03-29 18:37 UTC by TJS Security Command Center