



NCA-24.261026 – National CERT Advisory

Multiple Security Vulnerabilities in Node.js Allow Remote Code Execution, TLS Authentication Bypass and Unauthorized Access

The Node.js Project released security updates on 21 October 2026 addressing ten vulnerabilities (four high, three medium and three low severity) affecting Node.js 22.x, 24.x and 26.x. The flaws could enable remote code execution (RCE), memory corruption, TLS certificate validation bypass, DNS manipulation, HTTP response misassociation, predictable cryptographic identifiers and unauthorized filesystem access. Of particular concern is CVE-2026-111497, a high-severity out-of-bounds write in the Brotli decompression component that may allow arbitrary code execution, application crashes or disclosure of process memory through maliciously crafted compressed data.

Organizations should identify affected Node.js installations and apply the fixed releases (22.23.4, 24.22.1, 26.12.1 or later) immediately, prioritizing internet-facing services, API gateways, backend systems and cloud-hosted applications. Imperva reported demonstrating RCE against an official Node.js 24.22.0 build and Red Hat documented public proof-of-concept code causing memory corruption; widespread active exploitation has not been established. Threat intelligence update: 26 October 2026.

DOCUMENT SUMMARY

Field	Details
Advisory ID	NCA-24.261026
Threat Type	Multiple software vulnerabilities / Remote code execution / TLS authentication bypass / Unauthorized access
Severity	High (Immediate patching required)
Attack Vector	Maliciously crafted input (Brotli-compressed data, hostnames, IP address representations) processed through HTTP, DNS and TLS functions
Authentication Required	Not specified in the vendor bulletin
User Interaction	Not specified in the vendor bulletin; exposed applications must process attacker-controlled input
Risk Rating	High (Remote code execution, memory disclosure, TLS validation bypass and service disruption)
Scope & Applicability	Government, CII, financial institutions and other organizations running Node.js 22.x, 24.x or 26.x, including servers, containers, CI/CD pipelines and third-party applications embedding Node.js

IMPACT ANALYSIS

Successful exploitation of the identified vulnerabilities could result in:

- Remote Code Execution:** Unauthorized execution of code through applications processing crafted Brotli-compressed data.
- Information Disclosure:** Exposure of process memory, sensitive application data or filesystem metadata.
- Service Disruption:** Application crashes, service instability or denial of service.
- TLS Authentication Bypass:** Improper acceptance of certificates for unintended IP endpoints, potentially enabling server impersonation under appropriate conditions.
- DNS Manipulation:** Redirection of DNS requests through inconsistent hostname validation and resolution.



National Cyber Emergency Response Team

Government of Pakistan



6. **Security Control Bypass:** Circumvention of certain filesystem restrictions and validation controls.
7. **Potential secondary risks:** Unintended HTTP response acceptance, predictable identifiers across application instances and discrepancies in ZIP security scanning, with greater significance in multi-tenant, untrusted-input and sensitive-credential environments.

THREAT CHARACTERISTICS

Characteristic	Details
Threat Category	Multiple vulnerabilities (10 CVEs) in the Node.js runtime
Threat Status	Fixes released 21 October 2026; RCE demonstrated by researchers (Imperva) and public proof-of-concept code for memory corruption (Red Hat); widespread active exploitation not established
Initial Access	Attacker-controlled input processed by vulnerable applications: crafted Brotli streams, malformed hostnames, embedded NUL characters and non-canonical IPv4 addresses
Target Scope	Internet-facing services, API gateways, backend systems, cloud-hosted and multi-tenant applications
Attack Complexity	Not rated; DNS and TLS flaws depend on suitable application and network conditions
Privileges Required	Not specified; RCE would run with the privileges of the affected application
Affected Systems	Node.js 22.x before 22.23.4; 24.x before 24.22.1; 26.x before 26.12.1
Technical Disclosure	10 CVEs (4 High, 3 Medium, 3 Low); severities are vendor-assigned and CVSS scores are not provided in the vendor bulletin

Vulnerability Overview

CVE Identifier	Vulnerability Description	Severity
CVE-2026-111497	Brotli decoder out-of-bounds write potentially enabling remote code execution	High (8.8) 22.x, 24.x, 26.x
CVE-2026-105482	DNS hostname truncation through embedded NUL characters	High (7.5) 22.x, 24.x, 26.x
CVE-2026-105483	TLS certificate hostname verification bypass involving fullwidth IP digits	High (7.4) 24.x, 26.x
CVE-2026-106069	TLS certificate validation bypass involving non-canonical IPv4 addresses	High (7.4) 22.x, 24.x, 26.x
CVE-2026-105481	HTTP keep-alive connection reuse permits acceptance of unsolicited responses	Medium (5.3) 22.x, 24.x, 26.x
CVE-2026-106072	Improper certificate validation for non-canonical IPv4 hostnames	Medium (5.3) 22.x, 24.x, 26.x
CVE-2026-106074	Predictable cryptographic identifiers due to cached randomness in snapshots	Medium (4.8) 22.x, 24.x, 26.x
CVE-2026-106070	ZIP archive validation inconsistency involving attacker-controlled content	Low (3.3) 22.x, 24.x, 26.x
CVE-2026-106071	Filesystem permission over-grant through wildcard matching	Low (3.1) 22.x, 24.x, 26.x

National Cyber Emergency Response Team (NCERT)

Government of Pakistan

Pak Secretariat, L Block, Islamabad, Pakistan

+92-51-9203422 | info@pkcert.gov.pk | <https://pkcert.gov.pk/advisories>



National Cyber Emergency Response Team

Government of Pakistan



CVE-2026-106073	Filesystem permission bypass through fs.realpathSync()	Low (3.1)	22.x, 24.x, 26.x
-----------------	--	-----------	------------------

Technical Notes

- Brotli (CVE-2026-111497):** Specially crafted Brotli data triggers improper handling of an invalid block switch, causing the decoder to write beyond its allocated buffer. Applications decompressing attacker-controlled Brotli streams are particularly exposed.
- DNS and TLS (CVE-2026-105482, -105483, -106069):** Inconsistent hostname normalization, non-standard IPv4 representations and embedded NUL characters may allow redirected DNS-dependent requests, circumvented domain restrictions or improperly authenticated TLS connections, notably in applications using domain allowlists, server-side URL retrieval and certificate-based trust.
- Dependency & Trust Store Updates:**
Alongside Node.js core fixes, this update upgrades bundled dependencies including c-ares, OpenSSL, SQLite, and Undici. Additionally, it revokes the compromised eMudhra Intermediate CA from the default root certificate trust store.

INDICATORS OF EXPOSURE / TARGETING

The vendor bulletin documents software vulnerabilities rather than a malicious campaign and provides no attacker IP addresses, domains or file hashes. The following are investigative indicators, not confirmed indicators of compromise, and should be correlated with application behavior, process telemetry and network evidence. Conduct behavior-based hunting and investigate:

- Unexpected decompression errors involving Brotli-encoded content in application logs.
- Repeated Node.js crashes, segmentation faults or abnormal process termination.
- Unusual child processes initiated by Node.js services.
- Unexpected outbound connections following processing of externally supplied compressed content.
- Abnormal or non-canonical hostnames and IP address representations in TLS and DNS logs.
- Unexpected attempts to access filesystem paths outside the application's authorized scope.

REMEDIATION ACTIONS

1. Immediate Security Measures

Action Category	Specific Actions
Apply Security Updates	Upgrade to Node.js v22.23.4, v24.22.1 or v26.12.1 or later vendor-supported versions. Download only from the official Node.js portal (nodejs.org) or trusted system package repositories.
Inventory & Exposure Assessment	Inventory Node.js runtimes across servers, cloud platforms, container images, CI/CD pipelines, development workstations and third-party applications. Prioritize applications processing untrusted Brotli-compressed content, including HTTP clients, API integrations and decompression workflows.
Input Validation & TLS Security	Reject malformed hostnames, embedded NUL characters and ambiguous IP address representations. Apply strict hostname validation and never disable TLS certificate verification.
Interim Compensating Controls	Until patching is complete, disable unnecessary Brotli decompression for untrusted inputs where feasible; apply request-size limits, decompression safeguards and network access restrictions. These reduce exposure but do not replace patching.

National Cyber Emergency Response Team (NCERT)

Government of Pakistan

Pak Secretariat, L Block, Islamabad, Pakistan

+92-51-9203422 | info@pkcert.gov.pk | <https://pkcert.gov.pk/advisories>



National Cyber Emergency Response Team

Government of Pakistan



Action Category	Specific Actions
Runtime Hardening	Enforce least-privilege execution, container isolation, filesystem restrictions and network segmentation.
Remediation Validation	Verify installed runtime versions, rebuild and redeploy container images, restart services and perform regression and security testing. Remove vulnerable versions from deployment pipelines and inventories.
Monitoring and Escalation	Enable the monitoring below, preserve evidence and activate incident response if exploitation is suspected.

2. Vendor Remediation Timeline

The Node.js Project and independent researchers reported the following:

Date	Response Area	Vendor-Reported Measures
21 October 2026	Security release	Security releases 22.23.4, 24.22.1 and 26.12.1 published, addressing ten vulnerabilities and updating dependencies (c-ares, OpenSSL, SQLite, Undici) and the bundled certificate trust store.
21 October 2026	Trust-store update	Updates address trust-store concerns associated with the compromised eMudhra intermediate certification authority.
After release	Independent research	Imperva demonstrated RCE against official Node.js 24.22.0; patched 24.22.1 rejected the same input. Red Hat documented public proof-of-concept code causing memory corruption and crashes.

Remediation Clarification

1. Installing the fixed releases is the confirmed remediation; interim controls reduce exposure but do not replace patching.
2. CVE applicability differs across release lines. End-of-life versions are unsupported and should be migrated to maintained releases.
3. Downstream vendor-packaged builds may have backported fixes with different version numbering; confirm status with the vendor.
4. Reviewed sources do not establish widespread active exploitation; organization-specific exposure and compromise status require local validation.

3. Patch and Technical Disclosure Status

Technical Item	Disclosed Status	Customer Implication
Node.js 22.x (LTS)	Vulnerable < 22.23.4	Upgrade to v22.23.4 or later
Node.js 24.x (Active)	Vulnerable < 24.22.1	Upgrade to v24.22.1 or later
Node.js 26.x (Current)	Vulnerable < 26.12.1	Upgrade to v26.12.1 or later
CVSS scores	Not provided; severities are vendor-assigned	Prioritize by exposure and system criticality
Certificate trust store	Updated, including eMudhra CA concerns	Deploy trust-store updates alongside runtime patches
Packaged builds	May carry backported fixes with different versions	Confirm fix status with the package vendor

National Cyber Emergency Response Team (NCERT)

Government of Pakistan

Pak Secretariat, L Block, Islamabad, Pakistan

+92-51-9203422 | info@pkcert.gov.pk | <https://pkcert.gov.pk/advisories>



National Cyber Emergency Response Team

Government of Pakistan



4. Logging and Operational Controls

Control Area	Specific Actions
Application Logging	Review Brotli decompression errors and application crash records.
Process & Endpoint Telemetry	Monitor Node.js crashes, segmentation faults and unusual child processes.
DNS, TLS & Network Logging	Log outbound connections and non-canonical hostnames or IP address representations.
Filesystem Monitoring	Monitor attempts to access paths outside the application's authorized scope.
Evidence Preservation	Preserve relevant logs and telemetry before restarting or redeploying affected services.
Post-Patch Validation	Confirm fixed versions across all inventories and that no vulnerable versions remain in deployment pipelines.

ACTION SUMMARY & RESPONSE PRIORITIES

Action Type	Specific Steps
Patch Management	Upgrade to v22.23.4, v24.22.1, v26.12.1 or later.
Exposure Assessment	Inventory runtimes; prioritize internet-facing services and untrusted Brotli processing.
Input & TLS Hardening	Reject malformed hostnames, NUL characters and ambiguous IPs; keep certificate verification on.
Interim Controls	Restrict untrusted Brotli decompression and apply size limits until patched.
Threat Hunting	Investigate crashes, memory faults, child processes and anomalous DNS/TLS activity.
Runtime Hardening	Apply least privilege, container isolation and network segmentation.
Remediation Validation	Verify versions, rebuild images, restart services and test.

MONITORING & DETECTION REQUIREMENTS

#	Monitoring Activity
1	Monitor for unexpected Node.js process terminations and abnormal memory faults.
2	Review application logs for unexpected Brotli decompression errors.
3	Monitor suspicious compressed-data processing, especially from external sources.
4	Monitor for unauthorized child processes originating from Node.js applications.
5	Monitor anomalous outbound DNS/TLS communications and unexpected outbound connections after processing external compressed content.
6	Review TLS and DNS logs for abnormal or non-canonical hostnames and IP address representations.
7	Monitor filesystem access attempts outside the application's authorized scope.
8	Correlate alerts with application behavior, process telemetry and network evidence before concluding that exploitation has occurred.

DISASTER RECOVERY & INCIDENT RESPONSE REQUIREMENTS

Phase	Requirement Area	Actions / Controls
1	Incident Identification	Validate suspicious crashes or anomalies using process, application and network telemetry; confirm Node.js version and exposure.
2	Immediate Containment	Isolate affected hosts or containers, restrict untrusted input and network access, and apply patches or interim controls.
3	Evidence Preservation	Preserve application, process, DNS/TLS and network logs before restarting or redeploying services.
4	Secure Recovery	Upgrade to fixed releases, rebuild and redeploy images, restart services and perform regression and security testing.
5	Post-Incident Actions	Assess possible disclosure of sensitive memory or credentials, rotate secrets if exposure is suspected, strengthen runtime hardening and complete reporting through established procedures.

STRATEGIC PROTECTIVE MEASURES (CII & HIGH-VALUE TARGETS)

1. Maintain an inventory of Node.js runtimes, container images, CI/CD pipelines and embedded Node.js components.
2. Track official Node.js security announcements, apply security releases promptly and migrate end-of-life versions to maintained releases.
3. Apply least privilege, container isolation and network segmentation to Node.js workloads.
4. Preserve TLS verification, enforce strict hostname validation and deploy trust-store updates, including those for the eMudhra intermediate CA.
5. Apply request-size and decompression safeguards to services handling untrusted data and remove vulnerable versions from deployment pipelines.

INCIDENT REPORTING

Organizations identifying related exploitation, crashes or suspicious activity should activate their incident-response procedures and report relevant incidents to National CERT Pakistan:

- **Incident Reporting Portal:** <https://pkcert.gov.pk/report-incident/>
- **Email:** cert@pkcert.gov.pk
- **UAN:** +92 519203412

CALL TO ACTION – KEY IMPERATIVES

#	Requirement
1	Upgrade Node.js to v22.23.4, v24.22.1 or v26.12.1 or later.
2	Inventory all Node.js runtimes, containers and embedded components.
3	Prioritize internet-facing services and applications processing untrusted Brotli data.
4	Apply interim controls on untrusted compressed input until patching is complete.
5	Enforce strict hostname validation and keep TLS certificate verification enabled.
6	Monitor for crashes, unusual child processes and anomalous DNS/TLS activity.
7	Deploy certificate trust-store updates alongside runtime patches.



National Cyber Emergency Response Team

Government of Pakistan



#	Requirement
8	Verify remediation and escalate suspected exploitation to National CERT.

REFERENCES

1. Node.js Project: October 21, 2026 Security Releases – <https://nodejs.org/en/blog/vulnerability/october-2026-security-releases>
2. Node.js Project: Node.js 22.23.4 Security Release – <https://nodejs.org/en/blog/release/v22.23.4>
3. Node.js Project: Node.js 24.22.1 Security Release – <https://nodejs.org/en/blog/release/v24.22.1>
4. Node.js Project: Node.js 26.12.1 Security Release – <https://nodejs.org/en/blog/release/v26.12.1>
5. Red Hat: CVE-2026-111497 Brotli Security Bulletin – <https://access.redhat.com/security/vulnerabilities/RHSB-2026-015>
6. Imperva Threat Research: Node2Shell: From Brotli Data to Remote Code Execution – <https://www.imperva.com/blog/node2shell-from-brotli-data-to-remote-code-execution/>

