INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-25 06:03 UTC

CVE-2026-5426: Hardcoded ASP.NET Machine Keys Enable Unauthenticated RCE in KnowledgeDeliver LMS

CVE VULNERABILITY | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0220
Type	CVE Vulnerability
CVE ID	CVE-2026-5426
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0007 (22th percentile)
Affected Products	KnowledgeDeliver LMS (Digital Knowledge); ASP.NET/IIS-based deployments with shared hardcoded machine keys
Published	2026-05-25T14:00:00+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

A critical unauthenticated remote code execution vulnerability (CVE-2026-5426) in KnowledgeDeliver LMS allows attackers to fully compromise servers without any login credentials, by exploiting hardcoded cryptographic keys shipped identically across all customer installations. Attackers have already weaponized this flaw to deploy web shells, tamper with JavaScript served to end users, and deliver Cobalt Strike implants to anyone visiting the compromised LMS. Any organization running KnowledgeDeliver LMS on ASP.NET/IIS infrastructure should treat this as an active compromise scenario, not a future patch cycle.

Technical Analysis

CVE-2026-5426 (CVSS 9.5 Critical) affects KnowledgeDeliver LMS by Digital Knowledge, deployed on ASP.NET/IIS. The root cause is hardcoded, identical ASP.NET machine keys shipped across all customer instances (CWE-321: Use of Hard-coded Cryptographic Key; CWE-1188: Initialization of a Resource with an Insecure Default). Because machine keys are shared across deployments, any attacker with knowledge of the key can forge valid ViewState payloads. Deserialization of these forged payloads triggers insecure deserialization (CWE-502), yielding unauthenticated RCE at the IIS worker process privilege level. Observed

post-exploitation activity includes: in-memory web shell deployment (T1505.003), client-side JavaScript tampering targeting end users (T1565.001), Cobalt Strike BEACON delivery via browser (T1204.002), and C2 over HTTP (T1071.001). The attack pattern mirrors documented machine key abuse against Sitecore and Microsoft-documented IIS ViewState deserialization cases. No authenticated access is required at any stage. Confirmed patch status and affected version range in vendor advisory from Digital Knowledge and NVD entry at https://nvd.nist.gov/vuln/detail/CVE-2026-5426.

Action Checklist

1. Step 1: Containment, Immediately restrict external network access to all KnowledgeDeliver LMS instances at the perimeter. If internet-facing, place behind a WAF with ViewState validation rules or take offline until patched. Identify all IIS application pools running KnowledgeDeliver and verify machine key values in web.config are not hardcoded (validationKey/decryptionKey attributes). Document all affected hosts per CIS 1.1 asset inventory.
2. Step 2: Detection, Review IIS logs for anomalous POST requests to pages processing ViewState, particularly large or malformed __VIEWSTATE parameters. Search SIEM for outbound connections from IIS worker processes (w3wp.exe) to external IPs, a strong indicator of post-exploitation. Hunt for new or modified .aspx/.ashx files in the web root (D3-SFA: System File Analysis). Check browser-delivered JavaScript files served by the LMS for unauthorized modifications (T1565.001). Correlate with AU-6 audit record review. Alert on w3wp.exe spawning cmd.exe, powershell.exe, or mshta.exe (T1059.003, T1059.001).
3. Step 3: Eradication, Apply the vendor-supplied patch from Digital Knowledge when available; confirm the patched version eliminates hardcoded machine keys and generates unique keys per deployment. If patch is not yet available, immediately rotate all machine keys to cryptographically unique values per deployment by generating new randomized validationKey and decryptionKey values in web.config (minimum 64 hex characters each). Remove any discovered web shells and reverse any JavaScript modifications. Revoke and rotate all service account credentials associated with the IIS application pool (D3-CRO: Credential Rotation).
4. Step 4: Recovery, After key rotation or patching, validate no web shells remain using file integrity monitoring against a known-good baseline (D3-SFA). Confirm IIS application pools restart cleanly with new machine key values. Monitor outbound network from IIS hosts for 30 days post-remediation for Cobalt Strike BEACON callback patterns (malleable C2, periodic beaconing intervals). Re-enable external access only after validation. Reference NIST IR controls for post-incident monitoring.
5. Step 5: Post-Incident, This incident exposes a hardcoded credential/key management gap (CWE-321, CWE-1188). Conduct a full audit of all web applications for hardcoded cryptographic material. Implement NIST AC-6 (Least Privilege) for IIS application pool accounts to limit RCE blast radius. Enforce NIST SI-4 continuous monitoring on web server processes. Add CIS 7.1 and CIS 7.2 vulnerability and remediation process reviews specifically covering third-party LMS and web application vendors. Consider mandatory unique-key-per-deployment requirements in vendor procurement standards.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate immediately to CISO, legal counsel, and privacy officer if any evidence of web shell access to the LMS database, JavaScript tampering affecting end users (potential session/credential harvesting from students or staff), or Cobalt Strike BEACON callbacks is confirmed — triggering breach notification assessment under applicable data protection regulations (FERPA, GDPR, state breach laws) given the PII/PHI exposure scope of an LMS platform.
Recovery Notes	Before re-enabling external access, confirm machine key rotation is applied to every KnowledgeDeliver deployment in the environment (not just the primary instance) and validate via a controlled __VIEWSTATE replay test that the old hardcoded keys are fully invalidated. Monitor all IIS hosts running KnowledgeDeliver for a minimum of 30 days post-remediation using Sysmon process creation and network connection events, specifically hunting for w3wp.exe child processes and periodic outbound connections consistent with Cobalt Strike BEACON malleable C2 profiles that may indicate a pre-existing implant survived eradication. Re-audit the LMS JavaScript delivery pipeline at day 7 and day 30 using hash comparison against the post-remediation baseline to detect any re-compromise or persistent supply-chain-style tampering targeting end-user browsers.
Forensic Artifacts	IIS W3C access logs (default path: %SystemDrive%\inetpub\logs\LogFiles\W3SVC*\ex*.log) — filter for POST requests to KnowledgeDeliver page handlers with cs-bytes exceeding 10,000 bytes, which indicates a crafted ViewState deserialization payload exploiting the hardcoded machineKey (CVE-2026-5426 attack vector) | Windows Security Event Log Event ID 4688 (Process Creation) with ParentProcessName=w3wp.exe and NewProcessName matching cmd.exe, powershell.exe, mshta.exe, or certutil.exe — these child process spawns are the direct forensic result of successful ASP.NET ViewState deserialization RCE via the hardcoded machine keys | KnowledgeDeliver web root file system artifacts: any .aspx or .ashx files with NTFS creation timestamps post-dating the initial deployment, located in non-standard subdirectories of the IIS site root — these are attacker-placed web shells enabled by the unauthenticated RCE and confirmed by YARA scanning with webshell signature rules | KnowledgeDeliver JavaScript bundle files (e.g., files under /Scripts, /assets, or /Content) with SHA-256 hash mismatches against vendor-distributed or baseline copies — evidence of T1565.001 data manipulation used to deliver malicious JavaScript to LMS end users (students, staff) visiting the compromised platform | w3wp.exe process memory dump (captured via ProcDump before app pool recycling): contains decoded ViewState deserialization payloads, reflective DLL injection artifacts, and potential Cobalt Strike BEACON configuration data (malleable C2 profile, sleep timer, jitter value, C2 host/URI) attributable to the post-exploitation Cobalt Strike deployment described in the threat summary

Per-Action IR Details

Step 1: Containment — Immediately restrict external network access to all KnowledgeDeliver LMS instances at the perimeter. If internet-facing, place behind a WAF with ViewState validation rules or take offline until patched. Identify all IIS application pools running KnowledgeDeliver and audit machine key configurations in web.config for hardcoded machineKey values (validationKey/decryptionKey attributes). Document all affected hosts per CIS 1.1 asset inventory.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST SC-7 (Boundary Protection), NIST CM-6 (Configuration Settings), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Block port 80/443 to KnowledgeDeliver hosts at the perimeter firewall or host-based Windows Firewall using: `netsh advfirewall firewall add rule name='Block KDL External' protocol=TCP dir=in localport=80,443

action=block`. Enumerate all web.config files recursively for hardcoded machineKey attributes: `Get-ChildItem -Path C:\inetpub -Recurse -Filter web.config | Select-String -Pattern 'machineKey' | Select-Object Path, LineNumber, Line`. Cross-reference IIS application pools via `%windir%\system32\inetsrv\appcmd list apppool /processModel.userName:*` to identify service accounts in use.

Evidence: Before isolating the host, capture: (1) Full copy of all web.config files from the KnowledgeDeliver IIS site root and any virtual directories, preserving the hardcoded validationKey/decryptionKey values as evidence of the root cause; (2) IIS W3C access logs from `%SystemDrive%\inetpub\logs\LogFiles\W3SVC*` covering the maximum available retention window, with focus on POST requests containing oversized __VIEWSTATE parameters (typically >10KB, indicating a crafted deserialization payload); (3) Live netstat snapshot before isolation: `netstat -ano > c:\ir\netstat_snapshot.txt` to capture any active C2 connections from w3wp.exe PIDs; (4) Windows Security Event Log (EVTX) exported from the affected host before any reboot or pool recycling.

Step 2: Detection — Review IIS logs for anomalous POST requests to pages processing ViewState, particularly large or malformed __VIEWSTATE parameters. Search SIEM for outbound connections from IIS worker processes (w3wp.exe) to external IPs — a strong indicator of post-exploitation. Hunt for new or modified .aspx/.ashx files in the web root (D3-SFA: System File Analysis). Check browser-delivered JavaScript files served by the LMS for unauthorized modifications (T1565.001). Correlate with AU-6 audit record review. Alert on w3wp.exe spawning cmd.exe, powershell.exe, or mshta.exe (T1059.003, T1059.001).

NIST Phase: Detection Analysis

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

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

Compensating: Deploy Sysmon (config with SwiftOnSecurity baseline) and enable Event ID 1 (Process Create) to catch w3wp.exe spawning cmd.exe, powershell.exe, or mshta.exe. Run this PowerShell query against Sysmon logs: `Get-WinEvent -LogName 'Microsoft-Windows-Sysmon/Operational' | Where-Object {$_.Id -eq 1 -and $_.Message -match 'w3wp.exe' -and $_.Message -match 'cmd.exe|powershell.exe|mshta.exe'}`. For IIS log analysis without a SIEM, use Log Parser 2.2: `LogParser -i:W3C "SELECT cs-uri-stem, cs-bytes, c-ip, time-taken FROM ex*.log WHERE cs-method='POST' AND cs-bytes > 10000 ORDER BY cs-bytes DESC"` to surface large ViewState POST requests. For JavaScript tampering detection, compute SHA-256 hashes of all .js files in the LMS web root and diff against a known-good copy or Git baseline: `Get-FileHash -Path C:\inetpub\wwwroot\KnowledgeDeliver\*.js -Algorithm SHA256 | Export-Csv js_hashes.csv`.

Evidence: Capture before analysis: (1) IIS W3C logs — filter on cs-method=POST with cs-bytes >10000 directed at KnowledgeDeliver page handlers (e.g., /Default.aspx, /Course.aspx, or any page using ScriptManager/UpdatePanel which processes __VIEWSTATE); (2) Windows Security Event Log Event ID 4688 (Process Creation) with audit process creation enabled, filtering on ParentProcessName containing w3wp.exe and NewProcessName containing cmd.exe, powershell.exe, mshta.exe, or certutil.exe — these indicate successful ViewState deserialization RCE; (3) Sysmon Event ID 3 (Network Connection) for w3wp.exe making outbound connections to non-internal IP ranges, consistent with Cobalt Strike BEACON callbacks; (4) File system timestamps (`dir /T:C /O:D /S C:\inetpub\wwwroot\KnowledgeDeliver\*.aspx *.ashx`) to identify web shells created after the earliest suspected exploitation date; (5) KnowledgeDeliver JavaScript assets (e.g., LMS bundle .js files in /Scripts or /assets) compared to vendor-distributed checksums for evidence of T1565.001 supply-chain-style tampering targeting end users.

Step 3: Eradication — Apply the vendor-supplied patch from Digital Knowledge when available; confirm the patched version eliminates hardcoded machine keys and generates unique keys per deployment. If patch is not yet available, immediately rotate all machine keys to cryptographically unique values per deployment by generating new randomized validationKey and decryptionKey values in web.config (minimum 64 hex characters each). Remove any discovered web shells and reverse any JavaScript modifications. Revoke and rotate all service account credentials associated with the IIS application pool (D3-CRO: Credential Rotation).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST CM-6 (Configuration Settings), NIST IA-5 (Authenticator Management), NIST AC-2 (Account Management), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Generate cryptographically unique machine keys using PowerShell (no external tooling required): `$rng = [System.Security.Cryptography.RNGCryptoServiceProvider]::new(); $vk = New-Object byte[] 64; $dk = New-Object byte[] 32; $rng.GetBytes($vk); $rng.GetBytes($dk); [BitConverter]::ToString($vk).Replace('-',''); [BitConverter]::ToString($dk).Replace('-','')`. Replace validationKey (128 hex chars) and decryptionKey (64 hex chars) in web.config with these values, ensuring each KnowledgeDeliver deployment has a distinct key pair. For web shell removal, scan web root with ClamAV (freshclam updated) using the webshell signature database, and cross-validate with YARA rules from Neo23x0/signature-base (specifically webshell.yar): `yara64.exe webshells.yar C:\inetpub\wwwroot\KnowledgeDeliver\ -r`. Revoke IIS app pool service account password via AD: `Set-ADAccountPassword -Identity -Reset -NewPassword (Read-Host -AsSecureString)` and disable the old credential in all relevant configuration stores.

Evidence: Preserve before eradication: (1) Forensic copy (hash-verified) of any discovered web shells — capture full file content, NTFS creation/modification timestamps, and owner SID via `Get-Item | Select-Object FullName, CreationTimeUtc, LastWriteTimeUtc, @{n='Owner';e={(Get-Acl $_.FullName).Owner}}`; (2) Backup of the original web.config containing the hardcoded machineKey values as primary evidence of CVE-2026-5426 root cause — this is chain-of-custody evidence if regulatory notification is required; (3) Memory dump of the running w3wp.exe process (before recycling the app pool) using ProcDump: `procdump -ma c:\ir\w3wp_memdump.dmp` — this may contain decoded ViewState payloads, shellcode, or Cobalt Strike reflective DLL artifacts in process memory; (4) Full directory listing with timestamps of the KnowledgeDeliver web root before any file removal to establish the forensic baseline of attacker-placed artifacts.

Step 4: Recovery — After key rotation or patching, validate no web shells remain using file integrity monitoring against a known-good baseline (D3-SFA). Confirm IIS application pools restart cleanly with new machine key values. Monitor outbound network from IIS hosts for 30 days post-remediation for Cobalt Strike BEACON callback patterns (malleable C2, periodic beaconing intervals). Re-enable external access only after validation. Reference NIST IR controls for post-incident monitoring.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-3 (Malicious Code Protection), NIST SI-4 (System Monitoring), NIST CP-10 (System Recovery and Reconstitution), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Establish a file integrity baseline of the KnowledgeDeliver web root post-remediation using built-in PowerShell: `Get-ChildItem -Recurse C:\inetpub\wwwroot\KnowledgeDeliver\ | Get-FileHash -Algorithm SHA256 | Export-Csv C:\ir\webroot_baseline_postpatch.csv`. Schedule a daily comparison task to diff against this baseline. For Cobalt Strike BEACON detection without EDR, configure Sysmon Event ID 3 (Network Connection) alerts for w3wp.exe making periodic outbound connections at regular intervals to non-RFC1918 addresses — BEACON default intervals are 60s with ±50% jitter; use Wireshark/tshark capture scheduled via Task Scheduler: `tshark -i -f 'src host and not dst net 10.0.0.0/8 and not dst net 192.168.0.0/16' -w C:\ir\outbound_capture_%DATE%.pcap`. Review captures with Zeek or NetworkMiner for malleable C2 URI patterns typical of Cobalt Strike profiles.

Evidence: Capture for recovery validation: (1) IIS Application Event Log entries confirming successful app pool restart with new machine key (Event Source: ASP.NET, confirming no MachineKey-related errors on startup); (2) Post-rotation test — attempt a replay of a previously captured crafted __VIEWSTATE payload against the now-rotated keys and confirm HTTP 500 or MAC validation failure in IIS logs, proving the rotation invalidated the attacker's key material; (3) 30-day rolling Sysmon Event ID 3 captures filtered to w3wp.exe outbound connections, preserved as evidence of absence of BEACON callbacks; (4) File integrity comparison report (pre- vs. post-remediation hash diff) for all .aspx, .ashx, .js, and .config files in the KnowledgeDeliver web root, confirming no attacker-placed files persisted through remediation.

Step 5: Post-Incident — This incident exposes a hardcoded credential/key management gap (CWE-321, CWE-1188). Conduct a full audit of all web applications for hardcoded cryptographic material. Implement NIST

AC-6 (Least Privilege) for IIS application pool accounts to limit RCE blast radius. Enforce NIST SI-4 continuous monitoring on web server processes. Add CIS 7.1 and CIS 7.2 vulnerability and remediation process reviews specifically covering third-party LMS and web application vendors. Consider mandatory unique-key-per-deployment requirements in vendor procurement standards.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST AC-6 (Least Privilege), NIST SI-4 (System Monitoring), NIST SI-2 (Flaw Remediation), NIST RA-3 (Risk Assessment), NIST SA-11 (Developer Testing and Evaluation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Audit all IIS-hosted application web.config files for hardcoded machineKey, connectionString passwords, and appSettings secrets using a recursive PowerShell scan: `Get-ChildItem -Path C:\inetpub -Recurse -Include web.config,*.config | Select-String -Pattern 'machineKey|password=|connectionString|validationKey|decryptionKey' | Export-Csv C:\ir\config_audit.csv`. Reduce IIS app pool service account privileges to a dedicated low-privilege local or AD account with no local admin rights — verify with `whoami /priv` run as the app pool identity. Implement Sysmon rule (EventID 1) as a permanent detective control alerting on w3wp.exe spawning any child process, deployed via Group Policy. Add MITRE ATT&CK T1059.001/T1059.003 Sigma rules to log review workflow for ongoing detection of web server process abuse.

Evidence: Document for lessons learned and regulatory review: (1) The original web.config with hardcoded machineKey values (hash-verified, access-controlled) establishing the CWE-321/CWE-1188 root cause for any breach notification analysis; (2) Timeline reconstruction from IIS logs showing the earliest POST request with an oversized __VIEWSTATE parameter that could represent initial exploitation — this determines breach notification window for any PII/PHI processed by the KnowledgeDeliver LMS; (3) Inventory of all student/user accounts and PII stored in the LMS database, establishing data-at-risk scope given that post-exploitation JavaScript tampering (T1565.001) may have enabled client-side credential or session token harvesting from end users; (4) Evidence of Cobalt Strike implant deployment (if confirmed) — memory forensics artifacts, network BEACON captures, and any lateral movement Event IDs (4624 logon type 3, 4648 explicit credential use) from the IIS host to internal systems, for full scope determination.

Detection Guidance

Primary detection targets: (1) IIS access logs, hunt for POST requests with unusually large __VIEWSTATE parameters (>10KB) or requests containing base64-encoded binary data in ViewState fields. (2) Process telemetry, alert on w3wp.exe spawning child processes including cmd.exe, powershell.exe, cscript.exe, or mshta.exe; this is high-confidence post-exploitation evidence. (3) File integrity, baseline all .aspx, .ashx, and .js files in the LMS web root; new or modified files after the known exploitation window are suspect (D3-SFA). (4) Network, flag outbound HTTP/HTTPS connections from IIS worker processes to external IPs, particularly periodic beaconing patterns consistent with Cobalt Strike BEACON (T1071.001, T1001). (5) JavaScript delivery, compare hash values of LMS-served JavaScript files against known-good versions; browser-delivered malicious JS (T1565.001) is a confirmed TTP in this campaign. (6) Payload encoding, monitor for obfuscated or base64-encoded payloads in ViewState and request parameters (T1027). Log sources: IIS access logs, Windows Security Event Log (Event IDs 4688 for process creation with command-line auditing enabled), EDR process telemetry, network flow data, and web application firewall logs. NIST AU-2 and AU-12 should be verified to confirm these event types are being captured. CIS 8.2 log collection compliance should be confirmed across all IIS hosts.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://cloud.google.com/blog/topics/threat-intelligence/knowledgedeliver-viewstate-deserialization-vulnerability/`	Google Cloud Threat Intelligence report on KnowledgeDeliver exploitation — check for IOCs published in this report; URL is source-provided but should be validated before access	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1071.001** — Web Protocols
- **T1027** — Obfuscated Files or Information
- **T1059.003** — Windows Command Shell
- **T1505.003** — Web Shell
- **T1176** — Software Extensions
- **T1565.001** — Stored Data Manipulation
- **T1222.001** — Windows File and Directory Permissions Modification
- **T1001** — Data Obfuscation
- **T1190** — Exploit Public-Facing Application
- **T1059** — Command and Scripting Interpreter
- **T1059.001** — PowerShell
- **T1204.002** — Malicious File

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **CM-2** — Baseline Configuration
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **16.10** — Apply Secure Design Principles in Application 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
T1071.001	Web Protocols	Command-And-Control
T1027	Obfuscated Files or Information	Defense-Evasion
T1059.003	Windows Command Shell	Execution
T1505.003	Web Shell	Persistence
T1176	Software Extensions	Persistence
T1565.001	Stored Data Manipulation	Impact
T1222.001	Windows File and Directory Permissions Modification	Defense-Evasion
T1001	Data Obfuscation	Command-And-Control
T1190	Exploit Public-Facing Application	Initial-Access
T1059	Command and Scripting Interpreter	Execution
T1059.001	PowerShell	Execution
T1204.002	Malicious File	Execution

Sources

Source	URL	Tier
Threat Intelligence	https://cloud.google.com/blog/topics/threat-intelligence/knowledged...	T3
CVE-2026-5426 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-5426	T1
CVE-2026-5426: Digital Knowledge KnowledgeDeliver RCE	https://www.sentinelone.com/vulnerability-database/cve-2026-5426/	T3
CVE-2026-5426: KnowledgeDeliver deployments before February ...	https://cve.imfht.com/detail/CVE-2026-5426?lang=en	T3

Source	URL	Tier
Hard-coded ASP.NET/IIS machineKey value in Digital... · CVE-2026 ...	https://github.com/advisories/GHSA-g88c-8gfj-6c98	**T3**

Generated 2026-05-25 06:03 UTC by TJS Security Command Center