INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-15 19:02 UTC

Edge Password Manager Stored Credentials in Cleartext Memory at Startup, Fix Now Rolling Out

SECURITY ANALYSIS | **MEDIUM** | CVSS 5.0

SCC Item ID	SCC-STY-2026-0134
Type	Security Analysis
Severity	MEDIUM
CVSS Base Score	5.0
Affected Products	Microsoft Edge, all channels (Stable, Beta, Dev, Canary, Extended Stable) prior to build 148
Published	2026-05-15T10:49:39
Discovery Source	Rss

Executive Summary

Microsoft Edge's built-in password manager loaded all saved credentials into process memory in cleartext at browser startup, exposing them to any process running under the same user account or with administrative privileges. Microsoft initially defended the behavior as intentional before reversing course and committing to a fix beginning with build 148. The incident raises a pointed question for security leaders: browser-native credential storage has been marketed as a convenience feature, but this case demonstrates that it can introduce credential exposure risk equivalent to storing passwords in a plaintext file. Notably, no CVE identifier has been assigned to this exposure, suggesting Microsoft's handling prioritized rapid fix deployment over formal vulnerability tracking.

Technical Analysis

The vulnerability centers on a design decision in Edge's password manager: during browser initialization, all saved credentials were decrypted and loaded into process memory in cleartext, regardless of whether the user had opened the password manager UI or autofilled any credential. This behavior persisted for the full browser session.

The attack surface is straightforward. Any process operating under the same user context, including malware that has achieved user-level code execution without elevation, could read Edge's memory space and extract plaintext credentials. A publicly released proof-of-concept demonstrated the technique, and the attack aligns with MITRE ATT&CK T1555.003 (Credentials from Password Stores: Credentials from Web Browsers) and T1552.001 (Unsecured Credentials: Credentials in Files), with T1003 (OS Credential Dumping) applicable where an attacker has administrative access.

The CWE mapping is precise: CWE-316 (Cleartext Storage of Sensitive Information in Memory) is the primary weakness, with CWE-312 (Cleartext Storage of Sensitive Information) as a supporting classification. The issue is not a buffer overflow or injection flaw; it is a fundamental architectural choice about when credentials are decrypted and how long they remain accessible in memory.

Microsoft's initial characterization of the behavior as intentional is significant. It suggests the exposure was not an oversight introduced by a developer error but a deliberate product decision that went through design review. The public proof-of-concept and external pressure, not an internal audit, triggered the policy reversal. Microsoft has committed to remediation across all supported Edge channels, beginning with build 148, but no CVE identifier has been assigned based on available source data as of this writing.

All Edge channels are affected prior to build 148: Stable, Beta, Dev, Canary, and Extended Stable. Organizations that have deployed Edge at scale and permitted or encouraged use of the built-in password manager face meaningful credential exposure risk on any endpoint where user-level malware has executed, or where an administrator's session could be accessed by an unauthorized party.

Sources: BleepingComputer reporting on the disclosure and fix rollout; Microsoft Edge Stable Channel release notes (learn.microsoft.com/en-us/deployedge/microsoft-edge-relnote-stable-channel).

Action Checklist

1. Step 1: Assess exposure, determine whether Edge is deployed in your environment and whether the built-in Edge password manager is enabled or in active use by employees; query your MDM or browser management policy for password manager configuration status.
2. Step 2: Verify patch status, confirm all Edge installations across Stable, Beta, Dev, Canary, and Extended Stable channels are at build 148 or later; prioritize endpoints with elevated-privilege users or access to sensitive systems.
3. Step 3: Review credential storage policy, audit whether enterprise policy permits use of browser-native password managers; if no policy exists, create one; evaluate whether a dedicated enterprise password manager with stronger memory protections is appropriate for your risk profile.
4. Step 4: Assess EDR and memory protection coverage, verify that your endpoint detection and response tooling monitors for credential-scraping behavior (T1555.003, T1003) and that any process other than msedge.exe attempting to read Edge's memory triggers alerting on endpoints running Edge.
5. Step 5: Communicate findings and monitor, brief security leadership on the credential exposure window that existed prior to build 148 and track Microsoft's official release notes for confirmation that remediation is complete across all channels.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to incident response if Sysmon Event ID 10 or EDR telemetry shows any non-Edge process accessing msedge.exe memory with read permissions (GrantedAccess 0x1010 or 0x1410) during the pre-build-148 exposure window, or if Edge-stored credentials for privileged accounts (admin, service, VPN, PAM) are found to have been accessible; also escalate if regulatory notification obligations apply due to employee or customer PII stored in the Edge password manager.

Recovery Notes	After confirming build 148+ deployment across all channels, instruct users who had credentials stored in the Edge password manager to rotate passwords for any sensitive accounts — particularly those used to access financial systems, VPNs, email, or administrative consoles — since cleartext credential exposure to same-user-context processes cannot be ruled out without confirmed absence of scraping activity. Monitor for credential-reuse attacks (anomalous authentication attempts, impossible travel, off-hours logins) against those accounts for a minimum of 30 days post-patch using Windows Security Event ID 4625 (Failed Logon) and 4768/4769 (Kerberos ticket requests) as baseline indicators. Verify that Edge Group Policy disabling the password manager (if adopted) has propagated correctly by re-running the registry audit from Step 1 on a sample of endpoints two weeks post-deployment.
Forensic Artifacts	Sysmon Event ID 10 (ProcessAccess) logs targeting msedge.exe — specifically entries where SourceImage is not a Microsoft-signed Edge helper process and GrantedAccess includes read flags (0x1010, 0x1410), covering the period from Edge installation date through build 148 deployment; this is the primary artifact for determining whether the cleartext credential exposure was exploited | Edge Local State and Login Data SQLite files at %LOCALAPPDATA%\Microsoft\Edge\User Data\Default\Login Data — preserve a forensic copy before any credential rotation; these files reveal which credentials were stored in the password manager and therefore which accounts were exposed in cleartext memory at each startup | Windows Security Event ID 4688 (Process Creation) logs filtered for known memory-scraping utilities (procdump.exe, comsvcs.dll, taskmgr.exe used atypically, or unsigned executables) launched under the same user session as a running Edge process during the vulnerable window | Edge process memory dump (if a suspicious cross-process read was detected) — capture with `procdump -ma msedge.exe edge_memdump.dmp` before patching or restarting the browser on a suspect endpoint; strings analysis of the dump will confirm whether cleartext credentials were present and which account names/domains were exposed | Registry export of HKLM\SOFTWARE\Policies\Microsoft\Edge and HKCU\SOFTWARE\Policies\Microsoft\Edge at time of assessment — documents whether PasswordManagerEnabled was set to 0 (disabled via policy) or 1/absent (enabled), establishing whether organizational policy controls were in place to limit exposure scope before this vulnerability was publicly disclosed

Per-Action IR Details

Step 1: Assess exposure — determine whether Edge is deployed in your environment and whether the built-in Edge password manager is enabled or in active use by employees; query your MDM or browser management policy for password manager configuration status.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection & Analysis: Scoping and impact estimation for adverse events affecting credential stores

Controls: NIST IR-4 (Incident Handling) — requires determining scope of incidents affecting organizational systems, NIST SI-4 (System Monitoring) — inventory of active credential storage mechanisms on monitored endpoints, NIST RA-3 (Risk Assessment) — assess likelihood and impact of cleartext credential exposure in Edge process memory, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — enumerate all endpoints running any Edge channel prior to build 148, CIS 2.1 (Establish and Maintain a Software Inventory) — identify Edge version and channel per endpoint to determine pre-148 exposure window

Compensating: Run the following PowerShell one-liner across managed endpoints via PSRemoting or a startup script to enumerate Edge installs and PasswordManager policy status: `Get-ItemProperty 'HKLM:\SOFTWARE\Policies\Microsoft\Edge' -Name 'PasswordManagerEnabled' -ErrorAction SilentlyContinue; Get-ItemProperty 'HKCU:\SOFTWARE\Microsoft\Edge\BLBeacon' -Name 'version' -ErrorAction SilentlyContinue`. For unmanaged endpoints, use osquery: `SELECT name, version FROM programs WHERE name LIKE '%Microsoft

Edge%';` combined with `SELECT * FROM registry WHERE path LIKE 'HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Edge\PasswordManagerEnabled';`

Evidence: Before scoping, capture the current Edge Group Policy registry hive at `HKLM\SOFTWARE\Policies\Microsoft\Edge` and `HKCU\SOFTWARE\Policies\Microsoft\Edge` — specifically the `PasswordManagerEnabled` DWORD value — to document whether the password manager was enforced on or off via policy at the time of the exposure window. Also preserve the Edge installation version from `HKCU\SOFTWARE\Microsoft\Edge\BLBeacon\version` per endpoint to establish which hosts ran vulnerable builds.

Step 2: Verify patch status — confirm all Edge installations across Stable, Beta, Dev, Canary, and Extended Stable channels are at build 148 or later; prioritize endpoints with elevated-privilege users or access to sensitive systems.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: Eliminating the vulnerability from affected systems and verifying remediation completeness

Controls: NIST SI-2 (Flaw Remediation) — identify, report, and correct the Edge password manager cleartext memory flaw across all channels, NIST CM-6 (Configuration Settings) — enforce build 148+ as the minimum acceptable Edge configuration across all channels, CIS 7.3 (Perform Automated Operating System Patch Management) — apply Edge updates through automated patch management, monthly or more frequent, CIS 7.4 (Perform Automated Application Patch Management) — enforce Edge application updates across Stable, Beta, Dev, Canary, and Extended Stable channels via MDM or WSUS

Compensating: Without MDM, use a PowerShell script to query Edge version on all domain-joined hosts via PSRemoting: `Invoke-Command -ComputerName (Get-ADComputer -Filter *).Name -ScriptBlock { (Get-ItemProperty 'HKCU:\SOFTWARE\Microsoft\Edge\BLBeacon').version }`. Flag any host returning a version string below 148.x. For manual channel verification, check `%LOCALAPPDATA%\Microsoft\Edge\Application\` (Stable) and `%LOCALAPPDATA%\Microsoft\EdgeBeta\Application\`, `%LOCALAPPDATA%\Microsoft\EdgeDev\Application\`, `%LOCALAPPDATA%\Microsoft\EdgeCanary\Application\` for the version folder name. Prioritize hosts where the logged-in user is a local admin or has access to PAM, finance, HR, or VPN systems.

Evidence: Before applying or confirming the patch, preserve a point-in-time record of Edge version strings from all four channel paths on each endpoint. Capture `msedge.exe` file version via `(Get-Item 'C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe').VersionInfo.ProductVersion` to establish a defensible pre-remediation baseline. On elevated-privilege endpoints, also capture a process list snapshot with `Get-Process msedge | Select-Object Id, StartTime, Path` to document whether Edge was running and credentials were actively loaded in memory at the time of assessment.

Step 3: Review credential storage policy — audit whether enterprise policy permits use of browser-native password managers; if no policy exists, create one; evaluate whether a dedicated enterprise password manager with stronger memory protections is appropriate for your risk profile.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing policies and controls to prevent recurrence and reduce future credential exposure risk

Controls: NIST IR-8 (Incident Response Plan) — update IR plan to include browser-native credential storage as a credential exposure scenario requiring policy definition, NIST AC-3 (Access Enforcement) — enforce policy-based restrictions on which credential storage mechanisms are authorized for organizational use, NIST IA-5 (Authenticator Management) — manage browser-stored credentials as authenticators subject to organizational protection requirements, NIST CM-6 (Configuration Settings) — configure Edge Group Policy to set `PasswordManagerEnabled` to disabled if browser-native storage is not authorized, CIS 5.2 (Use Unique Passwords) — browser password manager policy must ensure credentials stored in Edge are unique per service, reducing blast radius if cleartext memory is read, CIS 4.6 (Securely Manage Enterprise Assets and Software) — manage Edge credential storage configuration through version-controlled GPO or MDM policy baseline

Compensating: Disable the Edge password manager organization-wide via Group Policy without MDM: set `HKLM\SOFTWARE\Policies\Microsoft\Edge\PasswordManagerEnabled` to `0` (DWORD) via a GPO Computer

Configuration preference or a login script. Verify propagation with `gpresult /r` on a test endpoint. Document the policy decision, the rationale tied to this cleartext memory exposure, and the approved alternative (e.g., KeePass with a shared team vault, Bitwarden self-hosted) in a one-page credential storage policy memo signed by the CISO or equivalent.

Evidence: Before drafting or modifying policy, retrieve and preserve the current state of all Edge password manager-related Group Policy registry keys: `HKLM\SOFTWARE\Policies\Microsoft\Edge\PasswordManagerEnabled`, `PasswordManagerBlocklist`, and `PasswordProtectionLoginURLs`. Export the full Edge policy hive with `reg export 'HKLM\SOFTWARE\Policies\Microsoft\Edge' edge_policy_baseline.reg`. This establishes the pre-remediation policy posture and supports any regulatory documentation requirement showing the organization's configuration at the time of the exposure window.

Step 4: Assess EDR and memory protection coverage — verify that your endpoint detection and response tooling monitors for credential-scraping behavior (T1555.003, T1003) and that process memory access by non-Edge processes triggers alerting on endpoints running Edge.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection & Analysis: Verify detection capability covers the specific attack vector (cross-process memory read of Edge credential store) and assess whether exploitation occurred during the exposure window

Controls: NIST SI-4 (System Monitoring) — monitor for unauthorized process memory access targeting msedge.exe, specifically ReadProcessMemory API calls from non-Edge processes, NIST AU-2 (Event Logging) — ensure logging captures process creation and cross-process memory access events relevant to T1555.003 and T1003, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — review Sysmon and Windows Security logs for evidence of credential scraping from Edge process memory during the pre-build-148 exposure window, CIS 8.2 (Collect Audit Logs) — confirm audit logging is enabled on all endpoints that ran Edge prior to build 148 and that logs cover the exposure window

Compensating: Deploy Sysmon with a configuration that captures Event ID 10 (ProcessAccess) targeting `msedge.exe` as the target image: add a rule matching `TargetImage` contains `msedge.exe` with `GrantedAccess` of `0x1010` or `0x1410` (read and query access flags commonly used by credential scrapers). Use the following Sigma rule stub for offline log analysis: `title: Cross-Process Read of Edge Password Manager, detection: filter: TargetImage|endswith: '\msedge.exe', EventID: 10, GrantedAccess: '0x1010|0x1410'`. Parse historical Sysmon logs with `Get-WinEvent -LogName 'Microsoft-Windows-Sysmon/Operational' | Where-Object {$_.Id -eq 10 -and $_.Message -like '*msedge*'}` to check for scraping attempts during the vulnerable window.

Evidence: Collect Sysmon Event ID 10 (ProcessAccess) logs targeting `msedge.exe` from all endpoints covering the period prior to Edge build 148 deployment. Also collect Windows Security Event ID 4688 (Process Creation) logs filtered for known credential-dumping tools (e.g., Mimikatz, procdump, comsvcs.dll) or any process that called `ReadProcessMemory` against the Edge process. If a commercial EDR is present, pull the process memory access telemetry for `msedge.exe` and flag any source process outside of `msedge.exe`, `MicrosoftEdgeUpdate.exe`, or Microsoft-signed Edge helper processes.

Step 5: Communicate findings and monitor — brief security leadership on the credential exposure window that existed prior to build 148 and track Microsoft's official release notes for confirmation that remediation is complete across all channels.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons learned, leadership communication, and ongoing monitoring of vendor remediation status across all affected Edge channels

Controls: NIST IR-6 (Incident Reporting) — report findings to security leadership including scope of affected endpoints, exposure window duration, and patch completion status per Edge channel, NIST IR-5 (Incident Monitoring) — track open remediation items across all Edge channels (Stable, Beta, Dev, Canary, Extended Stable) until build 148+ is confirmed deployed, NIST SI-5 (Security Alerts, Advisories, and Directives) — monitor Microsoft Security Response Center (MSRC) release notes and Edge update history for channel-specific build 148 confirmation, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — document this cleartext credential exposure as a

vulnerability management event and track remediation to closure, CIS 7.2 (Establish and Maintain a Remediation Process) — record the Edge cleartext memory issue in the remediation tracking register with per-channel patch verification milestones

Compensating: For teams without a vulnerability management platform, maintain a simple tracking spreadsheet with columns: Hostname, Edge Channel, Version at Discovery, Version at Patch, Date Patched, Verified By. Set a weekly calendar reminder to check the Microsoft Edge release schedule at `https://docs.microsoft.com/en-us/deployedge/microsoft-edge-release-schedule` and the Edge Enterprise release notes page for build 148 confirmation per channel. Prepare a one-page leadership brief covering: (1) what credentials were exposed and to which process contexts, (2) how many endpoints were affected, (3) patch completion percentage per channel, and (4) any detection gaps identified in Step 4.

Evidence: Before closing this item, preserve the Microsoft Edge release note entry or MSRC advisory confirming the cleartext memory fix is included in build 148 for each affected channel — screenshot or PDF the official release note as a dated artifact. Retain the endpoint version audit from Step 2 alongside post-patch verification results as documentation of remediation completeness. If a regulatory notification obligation exists (e.g., stored credentials included employee PII or customer data), preserve the full timeline: date of Microsoft's public disclosure, date of internal discovery, date of patch deployment per endpoint group.

Detection Guidance

Hunt for evidence of credential scraping targeting Edge's process memory space. Key signals to investigate:

Process memory access: Look for processes other than msedge.exe reading from Edge's memory space (PID cross-referencing in EDR telemetry). On Windows, Sysmon Event ID 10 (ProcessAccess) with target process matching msedge.exe and source process originating from an unexpected parent is a strong indicator of credential scraping activity.

ATT&CK T1555.003 behavioral indicators: Any tool or script querying the Edge Local State file or the Login Data SQLite database at %LOCALAPPDATA%\Microsoft\Edge\User Data\Default\Login Data should be flagged. Legitimate Edge processes access this path; access from cmd.exe, PowerShell, Python interpreters, or unknown binaries is suspicious.

T1003 and credential dumping tools: Audit EDR and AV logs for execution of known credential-dumping utilities. Given the public proof-of-concept for this specific issue, security teams should treat any execution of memory-reading tools on endpoints with Edge installed as a priority alert until patching is confirmed complete.

Policy audit: Review Group Policy or Microsoft Edge management policy configurations for PasswordManagerEnabled. If this is set to true or unmanaged, document the population of affected endpoints.

Log sources to check: Windows Security Event Log (process creation, object access), Sysmon (Event IDs 1, 10, 11), EDR process telemetry, browser management policy audit logs.

Framework Mappings

MITRE-ATTACK

- **T1552.001** — Credentials In Files
- **T1555.003** — Credentials from Web Browsers
- **T1003** — OS Credential Dumping

NIST-800-53R5

- **AC-6** — Least Privilege
- **IA-5** — Authenticator Management
- **SI-4** — System Monitoring

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552.001	Credentials In Files	Credential-Access
T1555.003	Credentials from Web Browsers	Credential-Access
T1003	OS Credential Dumping	Credential-Access

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/microsoft/microsoft-edge-to-s...	T3
	https://www.bleepingcomputer.com/news/microsoft/microsoft-edge-to-s...	T3
	https://www.bleepingcomputer.com/news/microsoft/microsoft-edge-gets...	T3
	https://www.bleepingcomputer.com/news/microsoft/microsoft-bounty-pr...	T3
Microsoft Edge release notes for Stable Channel	https://learn.microsoft.com/en-us/deployedge/microsoft-edge-relnote...	T1

Generated 2026-05-15 19:02 UTC by TJS Security Command Center