INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-20 06:54 UTC

CVE-2026-35275: Vulnerability in the Oracle VM VirtualBox product of Oracle Virtualization (component: Shared Folder...

CVE VULNERABILITY | HIGH | CVSS 7.5

SCC Item ID	SCC-CVE-2026-0340
Type	CVE Vulnerability
CVE ID	CVE-2026-35275
Severity	HIGH
CVSS Base Score	7.5
EPSS Score	0.0012 (2th percentile)
Affected Products	Oracle VM VirtualBox 7.2.8 (Shared Folders component)
Published	2026-06-17T10:40:19.767
Discovery Source	Nvd

Executive Summary

CVE-2026-35275 is a high-severity vulnerability in Oracle VM VirtualBox 7.2.8 that allows a low-privileged local attacker to break out of a guest virtual machine and access or modify critical data on the host system or other guest VMs. The flaw resides in the Shared Folders component and requires no user interaction, though exploitation demands high attack complexity. Organizations running VirtualBox 7.2.8 in virtualized infrastructure, including development, testing, or production environments, face risk of VM escape, data exposure, and cross-VM integrity compromise.

Technical Analysis

CVE-2026-35275 affects Oracle VM VirtualBox 7.2.8, specifically the Shared Folders component. The vulnerability is classified under CWE-284 (Improper Access Control) and carries a CVSS 3.1 Base Score of 7.5 with vector CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:N. Attack vector is local; privilege requirement is low (guest OS user); no user interaction is required. The scope change (S:C) indicates the vulnerability crosses the VM boundary, enabling impact on the host or sibling guest VMs. Confidentiality and Integrity impacts are both rated High; Availability is not affected. EPSS score is 0.0012 (2.1 percentile), indicating low current exploitation probability. The CVE is not listed in the CISA Known Exploited Vulnerabilities catalog as of this report. Mapped MITRE ATT&CK techniques include T1611 (Escape to Host), T1565 (Data Manipulation), and T1083 (File and

Directory Discovery). No public exploit code has been confirmed in the source data. Source authority: NVD (T1).

Action Checklist

1. Step 1: Identification & Containment. Identify all hosts running Oracle VM VirtualBox 7.2.8. Restrict guest OS accounts to the minimum necessary privilege level; audit which guest users have access to Shared Folders. Disable or unmount Shared Folder mappings on VMs where the feature is not operationally required until patching is complete. Reference: NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).
2. Step 2: Detection. Query endpoint and hypervisor logs for anomalous cross-VM file access events, unexpected reads or writes to host-side Shared Folder mount points, and privilege escalation events within guest OS sessions. Monitor for T1611 (Escape to Host) behavioral patterns: unexpected host process spawning correlated with guest activity, and unusual file system access from VirtualBox worker processes on the host. Reference: NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 8.2 (Collect Audit Logs).
3. Step 3: Eradication. Apply Oracle's official patch for VirtualBox 7.2.8 addressing CVE-2026-35275 as released via Oracle's Critical Patch Update. Confirm the patched version is installed across all hypervisor hosts. Remove or formally document exceptions for any Shared Folder configurations that cannot be immediately patched. Reference: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management).
4. Step 4: Recovery. After patching, re-enable Shared Folder mappings only for VMs with a verified operational need, applying least-privilege access controls to folder permissions. Validate that no unauthorized files were written to host-side directories during the exposure window. Re-run asset inventory to confirm all VirtualBox instances are at the patched version. Reference: NIST AC-6 (Least Privilege), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory).
5. Step 5: Post-Incident. Review the Shared Folders feature usage policy across the VM estate; restrict the feature by default and require documented approval to enable it. Evaluate whether VM isolation controls adequately prevent low-privileged guest users from reaching sensitive host resources. Map residual gaps to NIST AC-4 (Information Flow Enforcement) and D3-UAP (User Account Permissions) countermeasures. Conduct a broader audit of hypervisor configurations for similar scope-change risk patterns.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and legal counsel if Event ID 4663 or VBox.log analysis confirms unauthorized writes to host-side Shared Folder paths during the exposure window, or if any guest VM is identified as multi-tenant or hosting PII/PHI data subject to breach notification obligations under HIPAA, GDPR, or applicable state law.

Recovery Notes	After patching to the Oracle CPU-addressed version, re-enable Shared Folders only with the `--readonly` flag where write access is not operationally required, directly mitigating the guest-to-host write vector. Monitor host-side Shared Folder target directories and VBoxSVC process activity via Sysmon Event ID 1 and AU Event ID 4663 for a minimum of 30 days post-recovery to detect any delayed persistence mechanisms that may have been staged during the exposure window. Validate the patched version across the full VirtualBox estate using the osquery or PowerShell registry query documented in Step 3 before closing the incident record.
Forensic Artifacts	VirtualBox VM log files at %USERPROFILE%\.VirtualBox\Machines\\Logs\VBox.log — record all Shared Folder mount, unmount, and guest-initiated file operation events; primary artifact for reconstructing exploitation attempts against the CVE-2026-35275 Shared Folders component | Windows Security Event Log Event ID 4663 (An attempt was made to access an object) filtered on the host-side Shared Folder directory path — documents any unauthorized file reads or writes from VirtualBox worker processes to host filesystem during the exposure window | Windows Security Event Log Event ID 4688 (Process Creation) filtered on ParentProcessId matching VBoxHeadless.exe or VBoxSVC.exe PID — a guest-initiated VM escape via CVE-2026-35275 would manifest as unexpected host-side process spawning under a VBox parent process | Memory image of VBoxHeadless.exe and VBoxSVC.exe processes at time of suspected exploitation — the Shared Folders component vulnerability operates within the VBoxSVC service address space; heap analysis may reveal guest-injected shellcode or ROP chain artifacts specific to this exploit | Host-side Shared Folder target directory file hash baseline (SHA-256) compared against post-exposure state — files written to the host filesystem from a guest VM exploiting CVE-2026-35275 would appear as new or modified entries outside the expected change window, distinguishable from legitimate host-initiated writes by timestamp and originating process audit records

Per-Action IR Details

Step 1: Containment — Identify all hosts running Oracle VM VirtualBox 7.2.8. Restrict guest OS accounts to the minimum necessary privilege level; audit which guest users have access to Shared Folders. Disable or unmount Shared Folder mappings on VMs where the feature is not operationally required until patching is complete. Reference: NIST AC-6 (Least Privilege), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts).

NIST Phase: Containment

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

Controls: NIST AC-6 (Least Privilege), CIS 5.4 (IG1/IG2/IG3) — Restrict Administrator Privileges to Dedicated Administrator Accounts

Compensating: On each VirtualBox 7.2.8 host, enumerate all VMs with Shared Folders enabled using: `VBoxManage list vms` followed by `VBoxManage showvminfo | grep -i 'shared folder'`. For Windows hosts, run this as a PowerShell loop: `foreach ($vm in (& 'C:\Program Files\Oracle\VirtualBox\VBoxManage.exe' list vms)) { & VBoxManage showvminfo $vm | Select-String 'Shared Folder' }`. Unmount non-essential folders immediately with `VBoxManage sharedfolder remove --name ` — no SIEM required.

Evidence: Before disabling any Shared Folder mappings or modifying guest accounts, capture volatile state: (1) On the host, run `VBoxManage list runningvms` to record all active VMs at time of containment. (2) On each guest suspected of anomalous activity, acquire current network connections via `netstat -ano` (Windows guest) or `ss -tulnp` (Linux guest) and running process list via `tasklist /v` or `ps auxf`. (3) On the host OS, capture the VirtualBox process tree — specifically `VBoxSVC.exe`, `VBoxHeadless.exe`, and `VBoxSDS.exe` child processes — using `Get-CimInstance Win32_Process | Where-Object {$_.Name -like 'VBox*'} | Select-Object ProcessId,ParentProcessId,CommandLine`. These are destroyed the moment a VM is powered off or a Shared Folder is unmounted.

Step 2: Detection — Query endpoint and hypervisor logs for anomalous cross-VM file access events, unexpected reads or writes to host-side Shared Folder mount points, and privilege escalation events within guest OS sessions. Monitor for T1611 (Escape to Host) behavioral patterns: unexpected host process spawning correlated with guest activity, and unusual file system access from VirtualBox worker processes on the host. Reference: NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 8.2 (Collect Audit Logs).

NIST Phase: Detection Analysis

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

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 8.2 (IG1/IG2/IG3) — Collect Audit Logs

Compensating: Deploy Sysmon on VirtualBox hosts using SwiftOnSecurity's config; create a custom rule targeting Event ID 1 (Process Create) where ParentImage matches `VBoxHeadless.exe` or `VBoxSVC.exe` and Image is anything outside `C:\Program Files\Oracle\VirtualBox\`. On Linux hosts, use auditd with: `auditctl -w /proc/sys/kernel -p rwa -k vbox_escape` and watch for writes to `/proc` from the `vboxsf` kernel module context. Monitor the host-side Shared Folder path (default Windows: `\\vboxsvr\`) for file writes using `auditpol /set /subcategory:'File System' /success:enable /failure:enable` and filter Event ID 4663 (file object access) on those paths.

Evidence: This step is analytical and does not itself alter live state; however, if analysis confirms active exploitation, capture before any containment action: (1) VirtualBox host-side log files at `%USERPROFILE%\.VirtualBox\VBoxSVC.log` and `%USERPROFILE%\.VirtualBox\Machines\\Logs\VBox.log` — these record Shared Folder mount/unmount events and guest-initiated file operations. (2) Windows Security Event Log: Event ID 4663 filtered on access to the host-side Shared Folder directory, and Event ID 4688 (Process Creation) for any child processes spawned under a VBox worker PID. (3) Linux hosts: `/var/log/kern.log` entries referencing `vboxsf` module activity and `dmesg` output for unexpected kernel module loads. (4) Memory image of the VBoxHeadless process for the suspect VM — use ProcDump (`procdump.exe -ma `) to capture heap state that may contain guest-injected shellcode prior to process termination.

Step 3: Eradication — Apply Oracle's official patch for VirtualBox 7.2.8 addressing CVE-2026-35275 as released via Oracle's Critical Patch Update. Confirm the patched version is installed across all hypervisor hosts. Remove or formally document exceptions for any Shared Folder configurations that cannot be immediately patched. Reference: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 7.3 (IG1/IG2/IG3) — Perform Automated Operating System Patch Management, CIS 7.4 (IG1/IG2/IG3) — Perform Automated Application Patch Management

Compensating: For teams without automated patch management, download the Oracle CPU-patched VirtualBox installer directly from https://www.virtualbox.org/wiki/Downloads (validate SHA-256 checksum published in the Oracle advisory before executing). Verify installed version post-patch on Windows with: `Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Where-Object {$_.DisplayName -like '*VirtualBox*'} | Select-Object DisplayVersion`. On Linux: `vboxmanage --version`. Maintain a patch exception register in a simple spreadsheet for any host that cannot be patched immediately, documenting the compensating control (Shared Folders disabled) and a target remediation date.

Evidence: Before applying the Oracle CPU patch to any VirtualBox 7.2.8 host suspected of prior exploitation, capture: (1) Full memory image of the host using WinPmem (Windows) or LiME kernel module (Linux) — the exploit targets the Shared Folders component which operates in the VBoxSVC service context; post-patch, live memory artifacts from exploitation are destroyed. (2) File system snapshot of the host-side Shared Folder paths — hash all files present using `Get-FileHash -Algorithm SHA256` (PowerShell) or `sha256sum -r /*` (Linux) to establish a pre-patch baseline for later integrity comparison. (3) Copy and preserve VBox.log files for all running VMs at `%USERPROFILE%\.VirtualBox\Machines\\Logs\VBox.log` before the patch installer overwrites them. The patch installation process may restart VBoxSVC and rotate these logs.

Step 4: Recovery — After patching, re-enable Shared Folder mappings only for VMs with a verified operational need, applying least-privilege access controls to folder permissions. Validate that no unauthorized files were written to host-side directories during the exposure window. Re-run asset inventory to confirm all VirtualBox instances are at the patched version. Reference: NIST AC-6 (Least Privilege), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AC-6 (Least Privilege), CIS 1.1 (IG1/IG2/IG3) — Establish and Maintain Detailed Enterprise Asset Inventory

Compensating: Re-enable Shared Folders selectively using `VBoxManage sharedfolder add --name --hostpath --readonly` where write access is not required — the `--readonly` flag eliminates the guest-to-host write vector exploited by CVE-2026-35275. For integrity validation of host-side Shared Folder directories during the exposure window, compare current file hashes against the pre-patch baseline captured in Step 3 using a PowerShell diff: `Compare-Object (Import-Csv baseline_hashes.csv) (Get-FileHash \* -Algorithm SHA256 | Select-Object Hash,Path)`. Re-inventory patched VirtualBox versions using osquery: `SELECT name, version FROM programs WHERE name LIKE '%VirtualBox%';`

Evidence: This step restores functionality and does not primarily destroy forensic evidence; however, before re-enabling any Shared Folder mapping, confirm: (1) The host-side target directory has been integrity-checked against the pre-patch hash baseline — any SHA-256 mismatch on files in the Shared Folder path during the exposure window is a candidate indicator of host-side write exploitation. (2) Windows Security Event Log Event ID 4663 audit records for the Shared Folder host path have been exported and preserved covering the full exposure window (VirtualBox 7.2.8 installation date through patch application date). (3) Any guest VM snapshots taken during the exposure window are preserved in their current state before the VM is resumed or modified — snapshots may contain volatile state from the time of potential exploitation.

Step 5: Post-Incident — Review the Shared Folders feature usage policy across the VM estate; restrict the feature by default and require documented approval to enable it. Evaluate whether VM isolation controls adequately prevent low-privileged guest users from reaching sensitive host resources. Map residual gaps to NIST AC-4 (Information Flow Enforcement) and D3-UAP (User Account Permissions) countermeasures. Conduct a broader audit of hypervisor configurations for similar scope-change risk patterns.

NIST Phase: Post Incident

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

Controls: NIST AC-4 (Information Flow Enforcement), CIS 4.6 (IG1/IG2/IG3) — Securely Manage Enterprise Assets and Software, CIS 7.1 (IG1/IG2/IG3) — Establish and Maintain a Vulnerability Management Process

Compensating: Codify the Shared Folders restriction policy as a VirtualBox baseline configuration managed via version-controlled VM configuration files (.vbox XML). Use a Sigma rule targeting Sysmon Event ID 1 for `VBoxManage.exe` with CommandLine containing `sharedfolder add` without `--readonly` to detect policy violations going forward — this gives a 2-person team ongoing detection coverage without a SIEM. For the broader hypervisor audit, script a check across all `.vbox` files: `Select-String -Path C:\Users\*\.VirtualBox\Machines\*\*.vbox -Pattern 'SharedFolder'` to enumerate all persisted Shared Folder configurations estate-wide.

Evidence: Post-incident activity does not alter live compromise state; preserve for lessons-learned and potential regulatory disclosure: (1) Consolidated VBox.log files from all VMs active during the exposure window — these are the primary record of Shared Folder operations and should be archived to an immutable evidence store. (2) The pre- and post-patch file hash comparison report from Step 4, documenting whether any host-side files were modified during the window. (3) A full export of Windows Security Event Log Event ID 4688 and 4663 records from all VirtualBox hosts covering the exposure window, preserved in EVTX format with chain-of-custody documentation per NIST 800-61r3 §4 requirements for potential regulatory or legal use.

Detection Guidance

Focus detection on the host-side VirtualBox process layer and Shared Folder mount points. Key indicators: (1) VBoxSharedFolders or vboxsf kernel module generating file access events to host directories outside expected guest-mapped paths; (2) guest OS processes with low-privilege context triggering host-side file write events, correlate guest session user SID/UID against host file system audit logs (NIST AU-3, AU-6); (3) unexpected file creation or modification timestamps on host-side Shared Folder roots during periods of low administrative activity. For SIEM queries, filter VirtualBox host process logs for cross-boundary I/O events flagged by the guest-host interface. Behavioral indicator aligned to T1083: enumeration of host directory structures from within a guest session. Aligned to T1611: VirtualBox worker process on host spawning child processes not consistent with normal hypervisor operation. No confirmed IOC hashes, IPs, or domains are available in the source data for this CVE. Reference: NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs), D3-SFA (System File Analysis), D3-LAM (Local Account Monitoring).

Framework Mappings

MITRE-ATTACK

- **T1611** — Escape to Host
- **T1565** — Data Manipulation
- **T1083** — File and Directory Discovery

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

NIST-800-53R5

- **AC-3** — Access Enforcement

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control

ISO-27001-2022

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1611	Escape to Host	Privilege-Escalation

Technique ID	Technique Name	Tactic
T1565	Data Manipulation	Impact
T1083	File and Directory Discovery	Discovery

Sources

Source	URL	Tier
nvd	https://nvd.nist.gov/vuln/detail/CVE-2026-35275	T1
CVE-2026-35275 - Vulnerability Details - OpenCVE	https://app.opencve.io/cve/CVE-2026-35275	T3
Kwetsbaarheden verholpen in Oracle VM VirtualBox - Threat Radar	https://radar.offseq.com/threat/kwetsbaarheden-verholpen-in-oracle-...	T3
CVE-2026-10275: OpenSC Buffer Overflow Vulnerability	https://www.sentinelone.com/vulnerability-database/cve-2026-10275/	T3
Vulnerability in the Oracle VM VirtualBox product of... - GitHub	https://github.com/advisories/GHSA-49g5-fpx8-324c	T3

Generated 2026-06-20 06:54 UTC by TJS Security Command Center