INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-18 13:45 UTC

DirtyDecrypt PoC Raises Exploitation Risk Amid Growing Linux Root-Escalation Cluster

CVE VULNERABILITY | HIGH | CVSS 7.5

SCC Item ID	SCC-CVE-2026-0189
Type	CVE Vulnerability
CVE ID	CVE-2026-31635
Severity	HIGH
CVSS Base Score	7.5
EPSS Score	0.0005 (15th percentile)
Affected Products	Linux kernel (CONFIG_RXGK enabled builds); Fedora, Arch Linux, openSUSE Tumbleweed
Published	2026-05-18T03:18:33
Discovery Source	Rss

Executive Summary

A public exploit now exists for CVE-2026-31635 (DirtyDecrypt), a local privilege escalation flaw in the Linux kernel's rxgk module affecting Fedora, Arch Linux, and openSUSE Tumbleweed systems built with CONFIG_RXGK enabled. Any attacker who gains unprivileged user access to an affected system can execute arbitrary commands with root privileges, allowing complete compromise of the host. This vulnerability appears alongside active exploitation of a closely related Linux kernel flaw already listed on CISA's Known Exploited Vulnerabilities catalog, signaling coordinated threat actor interest in this vulnerability class.

Technical Analysis

CVE-2026-31635 (DirtyDecrypt, also DirtyCBC) is a local privilege escalation vulnerability in the Linux kernel's rxgk (RxGK Kerberos security layer) module. Affected configurations require CONFIG_RXGK to be enabled at build time. Confirmed affected distributions: Fedora, Arch Linux, and openSUSE Tumbleweed. CVSS base score: 7.5 (High). Attack vector: local, an attacker requires existing unprivileged user access to exploit. Three associated CWEs indicate the root cause chain: CWE-667 (Improper Locking) and CWE-787 (Out-of-bounds Write) likely describe the memory corruption primitive; CWE-269 (Improper Privilege Management) describes the resulting privilege escalation outcome. A public proof-of-concept was released following the upstream patch on April 25, 2026, accelerating the timeline for exploitation attempts. EPSS score is 0.0005 (15.4th percentile) as of scoring date, though PoC availability is a known leading indicator of near-term exploitation increases.

MITRE ATT&CK techniques: T1068 (Exploitation for Privilege Escalation), T1543 (Create or Modify System Process), T1078.003 (Valid Accounts: Local Accounts). CVSSv3.1 vector string is pending NVD publication. No threat actor attribution at this time. The related flaw 'Copy Fail' carries a CISA KEV designation, confirming active exploitation of this Linux kernel privilege escalation cluster by real-world threat actors. Patch: upstream kernel fix issued April 25, 2026; distribution-specific packages are tracked via Tenable plugin 310505.

Action Checklist

1. Containment: Identify all Linux hosts running Fedora, Arch Linux, or openSUSE Tumbleweed with CONFIG_RXGK enabled. Prioritize multi-tenant systems, shared developer environments, containerized workloads with privileged namespaces, and any host where unprivileged users have shell access. Isolate or restrict interactive login on unpatched high-risk hosts until remediation is confirmed; coordinate with system owners before restricting access to minimize operational disruption.
2. Detection: Query your asset inventory and CMDB for hosts running affected distributions. Run 'grep -r CONFIG_RXGK /boot/config-$(uname -r)' or check '/proc/config.gz' to confirm CONFIG_RXGK=y on each Linux host. In your SIEM, alert on privilege escalation patterns: sudden UID transitions to 0, unexpected setuid/setgid executions, and anomalous process trees spawning from low-privilege user sessions. Cross-reference with T1068 and T1543 detections in your EDR. No confirmed IOCs (hashes, IPs, domains) have been published for this CVE at this time.
3. Eradication: Apply the upstream kernel patch issued April 25, 2026, via your distribution's package manager: 'dnf update kernel' (Fedora), 'pacman -Syu' (Arch Linux), 'zypper update' (openSUSE Tumbleweed). Verify the running kernel version post-reboot matches the patched release. If an immediate reboot is not possible, assess whether the rxgk module can be blacklisted ('echo blacklist rxgk >> /etc/modprobe.d/rxgk-blacklist.conf') as a temporary mitigation. See Fedora Security Advisory, Arch Security Advisory, or openSUSE Security Advisory for distribution-specific guidance before blacklisting the module, as this may affect Kerberos-authenticated AFS workloads.
4. Recovery: After patching and rebooting, re-run the CONFIG_RXGK check to confirm the patched kernel is active. Review local user accounts and sudo rules on previously affected hosts for unauthorized modifications. Audit /etc/passwd, /etc/sudoers, and cron jobs for signs of persistence (T1543, T1078.003). Restore hosts from known-good baselines if unauthorized root activity is detected. Monitor affected hosts for 30 days post-remediation for anomalous privilege behavior.
5. Post-Incident: Document which hosts had CONFIG_RXGK enabled and were not covered by your standard kernel patch cadence. Review your kernel build configuration standards and determine whether CONFIG_RXGK is required in your environment; disable it in future builds if not operationally needed. Update your vulnerability management policy to include PoC-available CVEs as a trigger for expedited patching SLAs. Evaluate coverage of T1068 detections across your Linux fleet in your EDR and SIEM.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate immediately to senior IR leadership and legal/compliance if forensic evidence (UID-0 process trees, unauthorized /etc/passwd modifications, new SSH authorized_keys, or setuid binaries in /tmp or /dev/shm) indicates successful exploitation on any host storing PII, PHI, cardholder data, or credentials, or if the affected host is part of a multi-tenant or containerized environment where lateral movement to additional tenants or privileged Kubernetes namespaces is possible — active exploitation of a closely related kernel LPE is already confirmed on CISA KEV, elevating the likelihood of in-the-wild DirtyDecrypt exploitation.
Recovery Notes	After patching and rebooting all affected hosts, verify the patched kernel is active with 'uname -r' and confirm rxgk is absent from 'lsmod' output before restoring normal user access. Conduct a focused 30-day monitoring period on all previously vulnerable hosts, specifically alerting on any new UID-0 process spawned from a session with auid >= 1000, new entries in /root/.ssh/authorized_keys, and any cron or systemd unit modifications under low-privilege user home directories. If any host shows evidence of confirmed exploitation, treat it as fully compromised, rebuild from a known-good baseline rather than attempting in-place remediation, and escalate to determine whether the attacker pivoted to adjacent systems during the exposure window.
Forensic Artifacts	auditd SYSCALL records in /var/log/audit/audit.log showing execve or ptrace syscalls where auid >= 1000 and resulting euid = 0, without a corresponding sudo or su event — the forensic signature of a successful DirtyDecrypt LPE via the rxgk decryption token manipulation path | /var/log/auth.log or /var/log/secure entries capturing su/sudo invocations, PAM session opens for root, and SSH logins occurring from low-privilege accounts during the window when the vulnerable rxgk module was loaded | /var/log/kern.log entries for rxgk module load/unload events and any kernel BUG(), WARNING(), or NULL pointer dereference messages proximate in time to low-privilege user sessions, which may indicate exploit triggering attempts against the rxgk decryption path | Filesystem timeline artifacts from 'find / -newer /boot/config-$(uname -r) -perm -4000 -type f' and 'find /tmp /var/tmp /dev/shm -newer /boot/config-$(uname -r) -type f' identifying setuid binaries or ELF payloads dropped into world-writable locations by an attacker who achieved root via CVE-2026-31635 | Memory image captured via LiME prior to reboot, preserving in-memory rxgk kernel module state, any injected shellcode or modified kernel data structures, and process memory of the exploiting process — critical given that DirtyDecrypt's attack surface is an in-kernel decryption operation that leaves minimal on-disk artifacts

Per-Action IR Details

Containment — Identify all Linux hosts running Fedora, Arch Linux, or openSUSE Tumbleweed with CONFIG_RXGK enabled. Prioritize multi-tenant systems, shared developer environments, containerized workloads with privileged namespaces, and any host where untrusted local users have shell access. Isolate or restrict interactive login on unpatched high-risk hosts until remediation is confirmed.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST AC-6 (Least Privilege), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory)

Compensating: Run 'grep CONFIG_RXGK /boot/config-$(uname -r)' across your fleet via a push-based SSH loop or Ansible ad-hoc command: 'ansible all -m shell -a "grep CONFIG_RXGK /boot/config-$(uname -r)"'. For hosts without centralized management, use 'zcat /proc/config.gz | grep CONFIG_RXGK' locally. Restrict interactive logins on confirmed-vulnerable hosts by setting shell to /sbin/nologin for non-essential accounts in /etc/passwd, or applying PAM restrictions via /etc/security/access.conf to block all but named admin accounts. For containerized environments, audit privileged namespace exposure with 'grep -r privileged /etc/docker' and 'crictl inspect' on running pods.

Evidence: Before restricting login, capture: (1) 'last -n 100' and 'lastb -n 100' output to establish recent login history and failed attempts on the affected host; (2) 'ps auxf' snapshot to document all running processes and their parent-child relationships at time of containment; (3) 'id && whoami' output for all currently active sessions ('w' or 'who'); (4) /var/log/auth.log or /var/log/secure entries showing UID transitions — specifically look for su/sudo invocations from low-privilege accounts to UID 0 occurring within the rxgk module load window; (5) 'lsmod | grep rxgk' to confirm the module is loaded in memory at time of containment.

Detection — Query your asset inventory and CMDB for hosts running affected distributions. Run 'grep -r CONFIG_RXGK /boot/config-$(uname -r)' or check '/proc/config.gz' to confirm CONFIG_RXGK=y on each Linux host. In your SIEM, alert on privilege escalation patterns: sudden UID transitions to 0, unexpected setuid/setgid executions, and anomalous process trees spawning from low-privilege user sessions. Cross-reference with T1068 and T1543 detections in your EDR. No confirmed IOCs (hashes, IPs, domains) have been published for this CVE at this time.

NIST Phase: Detection Analysis

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

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without SIEM/EDR, deploy auditd rules targeting the specific privilege escalation mechanism of DirtyDecrypt: add '-a always,exit -F arch=b64 -S ptrace,process_vm_writev -F auid>=1000 -F auid!=4294967295 -k rxgk_lpe' to /etc/audit/rules.d/rxgk.rules to catch ptrace or memory write syscalls associated with decryption token manipulation from low-privilege processes. Monitor for UID-0 transitions using: 'ausearch -k rxgk_lpe | grep -E "uid=0|euid=0"'. Deploy the Sigma rule for T1068 (Privilege Escalation via Kernel Exploit) adapted for Linux auditd: alert on any process where auid >= 1000 spawns a child with euid=0 without a corresponding sudo/su event. Use 'inotifywait -m /etc/passwd /etc/sudoers' to catch real-time unauthorized modifications to privilege configuration files.

Evidence: Query /var/log/audit/audit.log for SYSCALL records showing execve calls where auid (original login UID) is >= 1000 but uid/euid in the resulting process is 0 — this is the forensic signature of a successful LPE via CVE-2026-31635. Capture 'ausearch -m SYSCALL -sv no -i' output filtered to the rxgk module load time. Review /var/log/kern.log for rxgk module initialization messages and any associated kernel warning or BUG() output that may accompany exploitation of the decryption path flaw. Document the kernel version string from 'uname -r' and cross-reference against the patched release version for each affected distribution.

Eradication — Apply the upstream kernel patch issued April 25, 2026, via your distribution's package manager: 'dnf update kernel' (Fedora), 'pacman -Syu' (Arch Linux), 'zypper update' (openSUSE Tumbleweed). Verify the running kernel version post-reboot matches the patched release. If an immediate reboot is not possible, assess whether the rxgk module can be blacklisted ('echo blacklist rxgk >> /etc/modprobe.d/rxgk-blacklist.conf') as a temporary mitigation — confirm with your distribution's security advisory before applying, as this may affect Kerberos-authenticated AFS workloads.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-3 (Configuration Change Control), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For hosts where automated patch management is unavailable, manually download the signed kernel package from the distribution's official mirror, verify the package signature ('rpm --checksig' for Fedora/openSUSE, or validate pacman keyring integrity with 'pacman-key --verify' for Arch) before installation. Confirm post-reboot kernel version with 'uname -r' and compare against the patched release noted in the distribution security advisory. For the rxgk blacklist interim mitigation on AFS-dependent hosts, test in a non-production environment first: 'modprobe -n --show-depends rxgk' to map dependent modules before blacklisting, and verify Kerberos AFS ticket operations still function via 'tokens' and 'aklog' after applying the blacklist and reloading initramfs with 'dracut -f' (Fedora/openSUSE) or 'mkinitcpio -P' (Arch).

Evidence: Before applying the patch, capture a full system memory image if exploitation is suspected (using LiME kernel module: 'insmod lime.ko path=/mnt/evidence/memory.lime format=lime') — this preserves any in-memory artifacts from rxgk decryption token manipulation that will be lost on reboot. Record 'rpm -qa kernel' (Fedora/openSUSE) or 'pacman -Q linux' (Arch) output as pre-patch baseline. After reboot, re-run 'lsmod | grep rxgk' to confirm the vulnerable module is no longer loaded under the patched kernel, and capture 'modinfo rxgk' if the module still appears to document the version present.

Recovery — After patching and rebooting, re-run the CONFIG_RXGK check to confirm the patched kernel is active. Review local user accounts and sudo rules on previously affected hosts for unauthorized modifications. Audit /etc/passwd, /etc/sudoers, and cron jobs for signs of persistence (T1543, T1078.003). Restore hosts from known-good baselines if unauthorized root activity is detected. Monitor affected hosts for 30 days post-remediation for anomalous privilege behavior.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AC-2 (Account Management), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 5.3 (Disable Dormant Accounts), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Run 'awk -F: "($3 == 0) {print}" /etc/passwd' to enumerate all UID-0 accounts and confirm only root is present. Audit /etc/sudoers and /etc/sudoers.d/* with 'visudo -c' and manually review for NOPASSWD entries added for non-standard accounts. Enumerate cron jobs for all users with 'for user in $(cut -f1 -d: /etc/passwd); do crontab -u $user -l 2>/dev/null; done'. Check /etc/cron.d/, /etc/cron.daily/, /var/spool/cron/ for newly added scripts. Use 'find / -perm -4000 -type f 2>/dev/null' to detect newly setuid binaries that could indicate a persistence backdoor installed post-exploitation via DirtyDecrypt. Deploy a YARA rule scanning /usr/local/bin, /tmp, and /dev/shm for ELF binaries dropped during exploitation.

Evidence: Collect /etc/passwd, /etc/shadow, /etc/sudoers, and /etc/sudoers.d/* before and after patching to diff for unauthorized account additions or privilege escalations. Run 'find /home /tmp /var/tmp /dev/shm -newer /boot/config-$(uname -r) -type f' to identify files created or modified during the exploitation window. Capture 'journalctl -b -1 --no-pager' (prior boot logs) to establish what occurred before the patched reboot. Review /root/.bash_history, /root/.ssh/authorized_keys, and /root/.bashrc for attacker-added entries that would persist post-patch.

Post-Incident — Document which hosts had CONFIG_RXGK enabled and were not covered by your standard kernel patch cadence. Review your kernel build configuration standards and determine whether CONFIG_RXGK is required in your environment; disable it in future builds if not operationally needed. Update your vulnerability management policy to include PoC-available CVEs as a trigger for expedited patching SLAs. Evaluate coverage of T1068 detections across your Linux fleet in your EDR and SIEM.

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: For teams without a formal vulnerability management platform, create a CONFIG_RXGK asset register in a spreadsheet or wiki documenting each host, its distribution, kernel version at time of discovery, patch date, and whether AFS/Kerberos dependencies prevent module blacklisting. Establish a documented process to monitor CISA KEV additions and distribution security mailing lists (Fedora Security Announcements, Arch Linux Security Advisories, openSUSE Security Announcements) as free, authoritative PoC-availability signals that trigger expedited SLAs. Author a Sigma detection rule for auditd targeting UID-0 escalation from unprivileged sessions on Linux hosts and test against your log corpus to measure T1068 coverage gaps before closing the incident.

Evidence: Compile the full asset inventory of CONFIG_RXGK-enabled hosts as the primary deliverable for lessons learned. Produce a timeline correlating kernel patch release date (April 25, 2026), PoC publication date, and each

host's patch application date to quantify exposure window per asset. Retain auditd logs, auth logs, and memory images (if captured) for a minimum of 90 days per NIST AU-11 (Audit Record Retention) to support any follow-on forensic or regulatory inquiry tied to this cluster of Linux LPE activity.

Detection Guidance

Primary detection method: confirm CONFIG_RXGK build flag on each Linux host. Run 'grep CONFIG_RXGK /boot/config-$(uname -r)', a result of 'CONFIG_RXGK=y' confirms the module is compiled in. In your SIEM or EDR, look for: (1) processes spawning with UID 0 from a parent process owned by a non-root user, particularly short-lived shell processes; (2) unexpected writes to /etc/passwd, /etc/sudoers, or /etc/cron.d from non-administrative users; (3) dmesg or kernel log entries referencing rxgk, memory faults, or locking errors coinciding with user-level activity. No public IOCs (file hashes, IP addresses, domains, or specific exploit binary names) have been confirmed in available sources as of this item's sourcing date. Monitor threat intelligence feeds and the NVD entry (https://nvd.nist.gov/vuln/detail/CVE-2026-31635) for IOC releases as the PoC circulates. Tenable Nessus plugin 310505 (verify plugin is current for your Nessus version before running) can be used to identify unpatched hosts at scale.

Framework Mappings

MITRE-ATTACK

- **T1543** — Create or Modify System Process
- **T1078.003** — Local Accounts
- **T1068** — Exploitation for Privilege Escalation

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-16** — Memory Protection
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

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

CIS-V8

- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **16.10** — Apply Secure Design Principles in Application Architectures
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

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

SOC2-TSC

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1543	Create or Modify System Process	Persistence
T1078.003	Local Accounts	Defense-Evasion
T1068	Exploitation for Privilege Escalation	Privilege-Escalation

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/exploit-available-fo...	T3
CVE-2026-31635 Detail - NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-31635	T1
CVE-2026-31635 - Red Hat Customer Portal	https://access.redhat.com/security/cve/cve-2026-31635	T3
Linux Distros Unpatched Vulnerability : CVE-2026-31635 | Tenable®	https://www.tenable.com/plugins/nessus/310505	T3
CVE-2026-31635 - Exploits & Severity - Feedly	https://feedly.com/cve/CVE-2026-31635	T3

Generated 2026-05-18 13:45 UTC by TJS Security Command Center