SICHERHEITS-AUDIT — RUNDE 11

NetMon Roots v1.1.20

Code Security Review — Juli 2026

Datum	29. Juli 2026
Analysierte Version	v1.1.20
Schwerpunkt	UFW-sysctl-Override-Feature, neues Laufzeitmess-Modul, Radio-Refactoring, Ist-Stand-Pruefung der Radio-Senken
Kritische Findings	0
Hohe Findings	0
Mittlere Findings	0
Haertungen umgesetzt	8 + 26 neue Regressionstests
Gesamt-Testsuite	2562 bestanden, 2 uebersprungen
Analysewerkzeug	Claude Fable 5 (Anthropic)
Projektverantwortlicher	Sven Froehlich

0 KRITISCHE — 0 HOHE — 0 MITTLERE FINDINGS (8 HAERTUNGEN UMGESETZT)

1. Zusammenfassung

Die elfte Sicherheitsanalyse pruefte den kompletten Diff seit dem Runde-10-Endstand (Commit 445279ac, 20. Juli 2026) bis zum aktuellen Arbeitsstand einschliesslich uncommitteter Aenderungen: 215 Commits und 468 Dateien, davon jedoch nur rund 3.750 Zeilen Produktivcode in 22 Modulen — der Rest sind Tests und Refactoring-Dokumentation. Sicherheitsrelevant neu waren vor allem das noch unveroeffentlichte UFW-sysctl-Override-Feature, das /etc/ufw/sysctl.conf als root umschreibt, sowie ein neues Laufzeitmess-Modul und dessen Einbindung in den zentralen Kommandopfad.

Diese Runde brachte kein Finding oberhalb der Melde-Schwelle von 8/10. Drei Kandidaten wurden gemeldet, alle drei im neuen UFW-sysctl-Feature. Zwei davon gingen an unabhaengige Validierungs-Sub-Agenten mit dem ausdruecklichen Auftrag, sie zu widerlegen: einer wurde widerlegt (2/10), einer von MEDIUM auf LOW herabgestuft (6/10). Der dritte ist eine reine Defense-in-Depth-Inkonsistenz (3/10). Behoben wurden alle drei trotzdem.

Bemerkenswert an dieser Runde ist eine strukturelle Erkenntnis: Der Radio-Diff-Agent stellte fest, dass die eigentlich scharfen Radio-Senken — mpv-/ffmpeg-Start, ICY-Metadaten, Aufnahme-Dateinamen, Playlist-Parsing — gar nicht in den geaenderten Dateien liegen, sondern in radio_controller.py und netmon_radio_core.py. Diese sind seit mehreren Runden unveraendert und fielen damit systematisch durch jeden diff-basierten Scope. Sie wurden in dieser Runde erstmals im Ist-Stand geprueft — mit vier weiteren Haertungen als Ergebnis. Fuer kommende Runden gilt daher: lange unveraenderte Senken periodisch im Ist-Stand pruefen, nicht nur den Diff.

Insgesamt wurden acht Defense-in-Depth-Haertungen umgesetzt und mit 26 neuen Regressionstests abgesichert. Die Gesamtsuite lief mit 2562 bestandenen und 2 uebersprungenen Tests — keine Regression. Alle Haertungen der Runden 7 bis 10 wurden gegengeprueft und sind intakt. Die Kindersicherung bleibt als bewusste Design-Entscheidung dokumentiert; die postinst-DNS-Policy und die Nur-Hash-Schnellpruefung auf der Downloadseite bleiben als Runde-9-Hinweise offen.

2. Methodik

Phase 1 - Diff-Recon

Bestimmung des Rundenscopes gegen den Runde-10-Endstand 445279ac, Trennung von Produktivcode, Tests und Dokumentation. Gegenpruefung der Vorrunden-Haertungen auf Regressionen: die xdg-open-Allowlist und der Update-URL-Host-Pin aus Runde 9, der killswitch-erhaltende Firewall-Recover aus Runde 9, die Sysctl-Allowlist und der suiteweite Reboot-Guard aus Runde 10 — alle intakt. Vor dem ersten Testlauf wurde bestaetigt, dass der Reboot-Guard aus Runde 10 aktiv ist (Betriebsregel seit dem Host-Neustart-Vorfall).

Phase 2 - Analyse-Sub-Agenten

Vier parallele Analyse-Sub-Agenten mit getrennten Scopes: (a) die sicherheitskritischen Feature-Controller inklusive des uncommitteten UFW-sysctl-Features, (b) Core-Module, Services und Skripte mit Schwerpunkt auf dem zentralen Kommandopfad, (c) der Radio-Diff und (d) — ausgeloest durch einen Abdeckungshinweis aus (c) — die Radio-Senken im Ist-Stand. Der Radio-Diff wurde nicht zeilenweise gesichtet, sondern ueber einen normalisierten Inhaltsdiff (beide Fassungen whitespace-normalisiert, sortiert, verglichen) als reines Refactoring belegt.

Phase 3 - False-Positive-Validierung

Je ein unabhaengiger Validierungs-Sub-Agent pro belastbarem Kandidat, mit dem ausdruecklichen Auftrag, den Kandidaten zu widerlegen statt ihn zu bestaetigen. Beide arbeiteten empirisch: einer extrahierte das Original-Paket ufw_0.36.2-6_all.deb, um den behaupteten Auslieferungsinhalt zu pruefen, der andere testete das sysctl-Fehlerverhalten an temporaeren Kopien (procps-ng 4.0.4) und schrieb einen Proof-of-Concept. Ergebnis: 2/10 (widerlegt) und 6/10 (von MEDIUM auf LOW herabgestuft). In beiden Faellen erwiesen sich zentrale Teilbehauptungen der Melder als falsch — dokumentiert in Abschnitt 3.

Phase 4 - Haertung & Regressionscheck

Acht Defense-in-Depth-Haertungen umgesetzt und mit 26 neuen Regressionstests abgesichert. Jede Haertung wurde vor der Umsetzung am echten Code gegengeprueft, nicht auf Zuruf des meldenden Agenten uebernommen. Gesamtsuite danach: 2562 bestanden, 2 uebersprungen.

3. Kandidaten & Validierung

Kein Kandidat erreichte die Melde-Schwelle von 8/10; es gibt in dieser Runde also keine bestaetigten Findings. Der Vollstaendigkeit halber ist dokumentiert, was gemeldet, wie es gegengeprueft und wie es eingestuft wurde — einschliesslich der Teilbehauptungen, die sich als falsch erwiesen.

K1 — stderr-Kontamination beim Read-Modify-Write von /etc/ufw/sysctl.conf [LOW, Konfidenz 6/10, gehaertet]

`netmon/features/security_audit_controller.py — _apply_ufw_sysctl_overrides() / _read_existing_sysctl_dropin().`

Das neue Feature liest /etc/ufw/sysctl.conf per 'sudo -n cat' ueber run_cmd, veraendert den Inhalt und schreibt ihn als root zurueck. run_cmd fuehrt jedoch stderr mit stdout zusammen: eine sudo-Warnung wie 'unable to resolve host ...' erscheint mit Exit-Code 0 im vermeintlichen Dateiinhalt, passiert die Filter unveraendert und landet als Muellzeile in einer Root-Konfigdatei. WIDERLEGT wurde die behauptete Auswirkung: Der Melder gab an, die Haertungswerte koennten dadurch beim Boot ausfallen. Empirisch geprueft (procps-ng 4.0.4, temporaere Kopien) warnt sysctl bei einer Syntax-Muellzeile, verarbeitet die Folgezeilen aber weiter und liefert rc=0; ufw-init ignoriert Fehler ohnehin per '|| true'. Kein Haertungsverlust. Uebersehen hatte der Melder ausserdem, dass ein Zurueckschreiben nur bei tatsaechlicher Aenderung stattfindet, und dass dasselbe Muster bereits committed existiert — neu eingefuehrt wurde es also nicht. Der Inhalt der sudo-Warnungen ist zudem nicht angreiferkontrolliert (Hostname, PAM, Audit); es gibt keinen Injektionsvektor.

Bewertung: Unterhalb der Melde-Schwelle, daher kein Finding. Trotzdem gehaertet: ein neues Plausibilitaetsgate (_implausible_sysctl_config_line) bricht den Lauf mit klarer Meldung ab, sobald der gelesene Inhalt eine Zeile enthaelt, die weder leer noch Kommentar noch 'key = value' ist — an beiden Lesestellen, also auch am bereits bestehenden Drop-in-Pfad. Bewusst Abbruch statt stillem Verwerfen: eine unerwartete Zeile in einer Root-Konfigdatei darf nicht kommentarlos geloescht werden.

K2 — Neue Schreibsenke ohne die Allowlist ihrer Schwester-Senke [LOW, Konfidenz 3/10, gehaertet]

`netmon/features/security_audit_controller.py — _build_ufw_sysctl_override_content().`

Die in Runde 10 eingezogene Schluessel-Allowlist sitzt bewusst direkt an der Schreibsenke (_render_sysctl_dropin_content) — als zweite Verteidigungslinie fuer den Fall, dass ein Spec auf anderem Weg als ueber die Transfer-Import-Validierung hereinkommt. Die neue Senke schreibt nach /etc/ufw/sysctl.conf, eine Datei, die laut eigenem Header /etc/sysctl.conf und /etc/sysctl.d ueberschreibt — sie wirkt also eher staerker als die Drop-in-Datei, trug aber keine dieser Pruefungen. Heute nicht ausnutzbar: der einzige Aufrufer uebergibt eine Modulkonstante, und es existiert kein Datenfluss von Transferpaket, Remote-JSON oder Dateinamen zu dieser Senke; das wurde explizit nachverfolgt. Gemeldet als Defense-in-Depth-Regression gegen eine dokumentierte Schutzentscheidung, nicht als Exploit.

Bewertung: Gehaertet: Die neue Senke prueft jetzt dieselbe SECURITY_BASELINE_ALLOWED_SYSCTL_KEYS-Allowlist und dieselbe Wert-Validierung wie ihre Schwester-Senke. Ein Eintrag wie 'kernel.core_pattern = |/tmp/payload' oder ein Wert mit eingebettetem Zeilenumbruch wird verworfen, statt als root persistiert zu werden.

K3 — 'Sysctl-Re-Apply dreht Kernelhaertung still zurueck' [WIDERLEGT, Konfidenz 2/10]

`netmon/features/security_audit_controller.py — _apply_ufw_sysctl_overrides() / apply_security_baseline_profile().`

Gemeldet wurde, das unbedingte Neuladen von /etc/ufw/sysctl.conf am Ende des UFW-Schritts koenne NetMons eigene Kernelhaertung ueberschreiben — konkret rp_filter. Widerlegt an drei Stellen. Erstens empirisch: der Validierer extrahierte das Original-Paket ufw_0.36.2-6_all.deb und wies nach, dass /etc/ufw/sysctl.conf ueberhaupt kein rp_filter enthaelt — weder gesetzt noch auskommentiert. Die tragende Praemisse war falsch. Zweitens: die einzige reale Ueberschneidung mit der NetMon-Baseline ist log_martians — und genau dafuer existiert der beanstandete Code. Die Paketdatei liefert log_martians=0 gegen NetMons Soll 1; das neue Override korrigiert das, bevor neu geladen wird. Der Melder hatte die Gegenmassnahme als Bug gemeldet. Drittens: eine nachgelagerte Wert-Verifikation existiert sehr wohl — nach allen Schritten laeuft ein erneuter Baseline-Check, der die Live-Kernelwerte liest. Eine stille Rueckdrehung ist strukturell ausgeschlossen. Hinzu kommt ein Denkfehler in der Richtung der Abweichung: rp_filter=1 (strict) ist strenger als NetMons bewusstes 2 (loose). Selbst im hypothetischen Fall wuerde die Sicherheit steigen und allein NetMons Funktionsabsicht (VPN, Multi-Interface) leiden — und das waere in der UI als 'Bewusste Abweichung' sichtbar, nicht still.

Bewertung: Kein Sicherheits-Finding, keine Codeaenderung noetig. Das Entfernen der beanstandeten Zeile wuerde im Gegenteil die log_martians-Gegenmassnahme wirkungslos machen.

4. Weitere Beobachtungen & Design-Entscheidungen

Scope-Luecke geschlossen — die Radio-Senken lagen nie in einem Diff

Der Radio-Diff-Agent stellte fest, dass die sicherheitsrelevanten Radio-Pfade (mpv-/ffmpeg-Start, ICY-Metadaten, Aufnahme-Dateinamen, Playlist-Parsing) nicht in den geaenderten Dateien liegen, sondern in radio_controller.py und netmon_radio_core.py. Beide sind seit Runden unveraendert und fielen damit systematisch durch jeden diff-basierten Scope. Sie wurden in dieser Runde erstmals im Ist-Stand geprueft — mit vier Haertungen als Ergebnis. Konsequenz fuer kommende Runden: lange unveraenderte Senken periodisch im Ist-Stand pruefen.

Aufnahme-Dateinamen aus ICY-Metadaten — sauber

Der naheliegendste Verdacht im Radio-Modul bestaetigte sich nicht: Stationsname und ICY-Titel laufen beide durch eine Slug-Funktion, die alles ausser Kleinbuchstaben und Ziffern ersetzt. Pfad-Traversal, Newlines, Null-Bytes, Shell-Metazeichen und fuehrende Bindestriche sind damit konstruktiv ausgeschlossen. Ebenso sauber: die Stream-URL-Schema-Allowlist wird auf allen Lade- und Speicherpfaden erzwungen, der EQ-Filter ist auf ganzzahlige Werte geklemmt, und der projectM-ZIP-Import ist gegen Zip-Slip abgesichert.

3.000 Zeilen Radio-Refactoring — ohne Verhaltensaenderung belegt

Statt den grossen Refactoring-Diff zeilenweise zu sichten, wurde ein normalisierter Inhaltsdiff gefahren. Ergebnis: die entfernten Zeilen sind ausnahmslos State-Initialisierungen und Geometrie-Ausdruecke, die 1:1 in neue Hilfsfunktionen gewandert sind. Die Whitelist beim Laden der Sprachdatei — ohne die ein Pfad-Traversal ueber die Konfigdatei moeglich waere — hat das Refactoring nachweislich ueberlebt.

Core und Services — keine Detektionsluecke durch die Refactorings

Der zentrale Kommandopfad core/command.py hat als einzige Aenderung einen Mess-Wrapper erhalten; Argumentaufbau, Timeouts und Env-Handling sind byte-identisch, und der Reboot-Guard aus Runde 10 wird nicht umgangen. Im Repository-Guard wurden alle Detektionszweige gegen die Vorversion abgeglichen und sind vollstaendig erhalten — kein untrusted Repository und kein Schluessel kann durch den Refactor unbemerkt durchrutschen. Das Support-Bundle leakt keine Keys, VPN-Configs oder PSKs.

UFW-Konflikthinweis im rp_filter-Zweig — geprueft, vorhanden

Bei der Widerlegung von K3 wurde zusaetzlich behauptet, den erklaerenden Hinweis '/etc/ufw/sysctl.conf setzt denselben Wert anders' erzeuge nur der sysctl-Zweig der Baseline-Auswertung, nicht der rp_filter-Zweig — eine Diagnose-Luecke also. Auch das hat der Nachtest widerlegt. Der rp_filter-Zweig ruft nicht den Low-Level-Helfer _sysctl_ufw_override_value() auf, nach dem gesucht wurde, sondern den Wrapper _sysctl_ufw_override_hint(), und haengt dessen Ergebnis an die Begruendung an; beide rp_filter-Specs tragen das dazu noetige config_source. End-to-End nachgestellt (Drop-in setzt 2, ufw-Datei setzt 1, Kernel meldet 1) liefert die Auswertung Status 'Bewusste Abweichung' samt ufw-Hinweis. Keine Luecke, kein Handlungsbedarf.

Kindersicherung — weiterhin bewusste Design-Entscheidung

Die PBKDF2-gestuetzte Kindersicherung im Secure-Browser bleibt wie in Runde 8 dokumentiert: Alltagshuerde gegen Gelegenheitszugriff im selben Konto, keine harte Grenze gegen technisch versierte Nutzer mit Kontozugriff (No-Root-Design). Keine Codeaenderung, unveraendert gueltig.

postinst-DNS & Nur-Hash-Schnellpruefung — Policy-Hinweise weiter offen

Die beiden Policy-Hinweise aus Runde 9 bleiben unveraendert bestehen: das root-postinst biegt aktive NM-DNS bei der Installation auf 8.8.8.8 (Opt-in empfohlen), und die Schnellpruefung auf der Downloadseite empfiehlt nur sha256sum -c statt der minisign-Vollpruefung. Kein ausnutzbarer Befund — als Design-Entscheidungen des Projektverantwortlichen offen.

5. Umgesetzte Haertungen (Runde 11)

Acht Haertungen, keine davon durch ein bestaetigtes Finding erzwungen — alle als Defense in Depth umgesetzt, jede mit Regressionstests abgesichert.

Allowlist an der UFW-sysctl-Schreibsenke — security_audit_controller.py

_build_ufw_sysctl_override_content() prueft Schluessel und Werte jetzt gegen dieselbe Allowlist und dieselben Format-Regeln wie die Drop-in-Senke. Unerlaubte Schluessel und Werte mit Zeilenumbruechen oder Shell-Metazeichen werden verworfen.

Plausibilitaetsgate gegen stderr-Kontamination — security_audit_controller.py

_implausible_sysctl_config_line() bricht jeden Read-Modify-Write ab, dessen gelesener Inhalt eine Zeile enthaelt, die weder leer noch Kommentar noch 'key = value' ist — an beiden Lesestellen. Verhindert, dass eine sudo-Warnung mit Exit-Code 0 als root in eine Root-Konfigdatei zurueckgeschrieben wird.

'--'-Optionsterminator vor dem mpv-Medienargument — radio_controller.py

mpv wertet Optionen an jeder Position aus. Ein Playlist-Eintrag namens '--script=payload.lua' waere als Option geladen worden statt als Dateiname. Das Medienargument steht jetzt hinter einem '--'; der EQ-Filter wird korrekt davor einsortiert.

Handover-Payload durchlaeuft die URL-Allowlist — radio_controller.py

Die einzige Stelle im Modul, an der ein Stations-Datensatz ohne normalize_station_entry() in den App-Zustand gelangte — und damit die Stream-URL-Schema-Allowlist umging, die ueberall sonst erzwungen wird. Ein file://-Eintrag im Uebergabe-Payload wird jetzt verworfen.

mpv-IPC-Socket in ein privates Verzeichnis — radio_controller.py

Wer auf diesen Socket schreiben darf, darf mpv steuern — und mpv kennt ein 'run'-Kommando. Der Socket lag bisher direkt im State-Verzeichnis (typisch 0755). Er liegt jetzt in XDG_RUNTIME_DIR, ersatzweise in einem eigens angelegten 0700-Unterverzeichnis.

ffmpeg-Protokoll-Allowlist ohne 'file' — radio_controller.py

Die Aufnahme erhaelt jetzt ein explizites '-protocol_whitelist http,https,tcp,tls,crypto'. Ohne diese Grenze waehlt ffmpeg den Demuxer nach Inhalt — eine als Stream ausgelieferte HLS-/concat-Playlist konnte damit lokale Dateien in die Aufnahme ziehen.

Wert-seitige Redaction im Messlog — core/performance_monitor.py

Die Redaction pruefte bisher nur den Metadaten-Schluessel; ein Secret unter einem harmlosen Schluessel waere im Klartext im Log gelandet. Werte laufen jetzt zusaetzlich durch dieselbe Redaction wie die Diagnosedaten — rekursiv auch in Listen und verschachtelten Dicts, ebenso der Fehlertext. Kein damaliger Aufrufer war betroffen; die Haertung schliesst die Falle fuer kuenftige.

Symlink-sicherer, nutzerprivater Messlog — core/performance_monitor.py

Der Log wird jetzt mit O_NOFOLLOW | O_APPEND und Modus 0600 geoeffnet, die Rotation ruehrt symlink-verlinkte Pfade nicht an. Ein untergeschobener Symlink kann die Messwerte damit weder umlenken noch eine fremde Datei ueberschreiben.

26 neue Regressionstests

Elf in test_netmon_lynis_output.py (Allowlist an der UFW-Senke, Erkennung von stderr-Zeilen, Abbruch statt Zurueckschreiben, ufw-Konflikthinweis im rp_filter-Zweig), neun in test_netmon_radio_module.py (Optionsterminator, EQ-Reihenfolge, Protokoll-Allowlist, Socket-Rechte inklusive Fehlerfallback, Handover-Normalisierung) und sechs in test_performance_monitor.py (Wert-Redaction verschachtelt, Symlink-Verweigerung, Dateirechte). Gesamtsuite danach: 2562 bestanden, 2 uebersprungen.

6. Geprüfte Sicherheitskategorien

Kategorie	Ergebnis
Schreiben in root-eigene Konfig (/etc/ufw/sysctl.conf)	**Allowlist + Plausibilitaetsgate ergaenzt**
Argument-/Options-Injektion in externe Player (mpv, ffmpeg)	**'--' + Protokoll-Allowlist ergaenzt**
Fremddaten in Dateinamen (ICY-Metadaten, Stationsnamen)	**Kein Finding (Slug-Normalisierung)**
URL-Schema-Allowlists (Streams, Logos, Homepages)	**Handover-Luecke geschlossen**
Zentraler Kommandopfad (core/command.py)	**Kein Finding (nur Mess-Wrapper)**
Secret-Leaks in Logs, Support-Bundle und Diagnose	**Wert-Redaction ergaenzt**
Dateirechte, Symlink / TOCTOU an neuen Schreibpfaden	**O_NOFOLLOW + 0600 + 0700-Socketdir**
Repository-Guard, DNS-Service, Prozess-Guard	**Detektionszweige vollstaendig erhalten**
Unsichere Deserialisierung (pickle, eval, shell=True)	**Kein Finding**
TLS-Verifikation und Redirect-Handling	**Keine Abschaltung, kein http->file**
Autorisierung / Kindersicherung	**Bewusste Design-Entscheidung**
Regression der Runde-9/10-Haertungen	**Alle intakt**

Auditor & Methode

Auditor: Claude Fable 5 (Anthropic) — vier Analyse-Sub-Agenten + zwei unabhaengige False-Positive-Validierungs-Sub-Agenten

Projektverantwortlicher: Sven Froehlich

Datum: 29. Juli 2026

Codestand: Arbeitsstand v1.1.20 (Diff seit dem Runde-10-Endstand 445279ac, einschliesslich uncommitteter Aenderungen), ergaenzt um eine Ist-Stand-Pruefung der Radio-Senken — Runde 11

Methode: Statische Code-Analyse + manuelle Datenfluss-Verifikation + normalisierter Inhaltsdiff fuer die Refactoring-Masse + konfidenz-gescorte FP-Validierung mit empirischer Gegenpruefung (Paketextraktion, sysctl-Verhaltenstests) + Umsetzung von acht Haertungen mit 26 Regressionstests + Regressions-Gegenpruefung der Vorrunden-Haertungen (Gesamtsuite 2562 bestanden, 2 uebersprungen) — kein Laufzeit-Penetrationstest

Dieses Audit ersetzt keinen professionellen Penetrationstest durch eine zertifizierte Sicherheitsfirma. Die Analyse beschraenkt sich auf den angegebenen Codestand. Nicht erfasst: Laufzeitverhalten, Systembibliotheken, Netzwerkkonfiguration.