TRMM-Linux-Agent-Konzept

TRMM-Linux-Agent-Konzept

Kurzfassung

Tactical RMM (TRMM) dokumentiert inzwischen einen **Linux-Agenten im Beta-Status**. Code-signierte Agenten, einschliesslich Linux und macOS, stehen gemäss Hersteller ab Sponsor Tier 1 zur Verfügung. Dieser offizielle Weg ist für neue Installationen zu bevorzugen.

Ein organisationsintern gebauter Fork bleibt technisch möglich, verursacht aber dauerhafte Pflege-, Lieferketten- und Lizenzrisiken. Er ist keine gleichwertige Standardlösung.

> **Getestet:** Ein intern gebauter Agent auf Basis von Agent `v2.11.0` wurde auf Linux erfolgreich registriert; `tacticalagent.service` und die Mesh-Verknüpfung funktionierten. Dieser Befund gilt nur für den damaligen Fork und dessen Build-Pipeline.
>
> **Generisch/aktuell:** Die offiziellen Installationsschritte unten stammen aus der laufenden TRMM-Dokumentation. Funktionen und Dienstnamen sind nach jedem Agent-Update erneut zu prüfen, da der Linux-Agent als Beta bezeichnet wird.

Entscheid

1. **Standard:** Offiziellen Linux-Installer aus der eigenen TRMM-Instanz verwenden.
2. **Ausnahme:** Eigenbau nur, wenn der offizielle Bezug nachweislich nicht möglich ist und Security, Lizenz sowie Wartungsverantwortung geklärt sind.
3. **Nicht verwenden:** Ungepinnte Fremdskripte, Builds direkt aus `master` oder Live-Kompilation auf verwalteten Zielsystemen.

Offizielle Installation

1. In TRMM **Agents → Install Agent** öffnen.

2. Client, Site, Gerätetyp, Architektur, Ablaufzeit und **Linux** wählen.
3. Generiertes Shellskript herunterladen und sicher auf den Zielhost übertragen.
4. Ausführen:

```
chmod 700 ./rmm-<client>-<site>-<type>.sh
sudo ./rmm-<client>-<site>-<type>.sh
```

5. Nach erfolgreichem Rollout löschen, denn das Skript enthält ein zeitlich gültiges Enrollment-Token:

```
rm -f ./rmm-<client>-<site>-<type>.sh
```

Ein Linux-Deployment-Link wird laut TRMM-Dokumentation derzeit nicht angeboten. Das generierte Skript deshalb nur über einen geschützten, kurzlebigen Ablageort verteilen; niemals in ein öffentliches Repository oder eine öffentliche Knowledge Base hochladen.

Verifikation

```
sudo systemctl status tacticalagent.service --no-pager
sudo journalctl -u tacticalagent.service -n 100 --no-pager
```

Zusätzlich in TRMM prüfen:

- Host erscheint beim erwarteten Client und Standort.
- Letzter Check-in ist aktuell.
- Checks und Skripte laufen mit erwarteten Rechten.
- Mesh/Remote-Zugriff ist vorhanden, sofern mitinstalliert.
- Neustart stellt Agent und Mesh wieder her.

Funktionsumfang realistisch einordnen

Der Feldtest des internen Forks `v2.11.0` bestätigte Monitoring, Skriptausführung, Terminal und Mesh. In diesem Stand waren unter anderem Softwareinventar, grafisches Patch-Management, Service-Steuerung und Windows-spezifische Funktionen nicht oder nur eingeschränkt vorhanden.

Dies ist **keine aktuelle, vollständige Produktmatrix**. Vor dem Rollout muss die tatsächlich eingesetzte Agent-Version mit einem Testhost gegen die benötigten Funktionen geprüft werden.

Kontrollierter Eigenbau (Ausnahme)

1. Upstream-Tag und Commit pinnen; nie direkt aus `master` veröffentlichen.

2. Änderungen reviewen und Abhängigkeiten mit `go mod verify` prüfen.
3. In isolierter CI für die benötigten Architekturen bauen:

```
CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -trimpath -ldflags="-s -w" -o rmmagent-linux-amd64
sha256sum rmmagent-linux-amd64 > SHA256SUMS
```

4. Artefakte und `SHA256SUMS` unveränderbar in einem zugriffsgeschützten Release-Repository ablegen.
5. Installer und Self-Update müssen die Prüfsumme **vor** dem Ersetzen der Binärdatei validieren.
6. Enrollment-Werte nur zur Laufzeit übergeben und nicht loggen.
7. Rollback, Update-Überwachung und regelmässige Upstream-Diffs betreiben.

> Eine SHA-256-Prüfsumme schützt nur dann vor Manipulation, wenn das Manifest über einen getrennt vertrauenswürdigen beziehungsweise signierten Kanal bezogen wird.

Sicherheits- und Lizenzgrenzen

- Die **Tactical RMM License v1.0 ist keine Open-Source-Lizenz**.
- Sie erlaubt die Nutzung zur Überwachung eigener und verwalteter Kundensysteme, enthält aber weitgehende Einschränkungen für kommerzielle Bereitstellung, SaaS, Managed Hosting, Installation/Konfiguration als Dienstleistung sowie abgeleitete Werke.
- Vor Verteilung eines Forks, Rebranding oder einem kommerziellen Angebot die konkrete Nutzung rechtlich und gegebenenfalls mit AmidaWare klären.
- Wird ein abgeleitetes Werk Dritten bereitgestellt, sind die Pflichten zur Bereitstellung des vollständigen Quellcodes und zum Copyright-Hinweis zu beachten.
- Diese Notiz ist keine Rechtsberatung.

Abnahme

- ☐ Offizieller Installer oder freigegebener, gepinnter Eigenbau verwendet
- ☐ Keine Tokens in Shell-History, Tickets, Logs oder Dokumentation
- ☐ Prüfsumme/Signatur validiert
- ☐ Check-in, Checks, Skripte und Remote-Zugriff getestet
- ☐ Neustart und Agent-Update getestet
- ☐ Funktionslücken dokumentiert

☐ Lizenz- und Wartungsverantwortung geklärt

Quellen

- Tactical RMM: Agent Installation – Linux Agent (beta)
- Tactical RMM: Code Signing
- Tactical RMM License v1.0
- Tactical RMM Agent (Upstream)

Quelle: Obsidian `Knowledge Base/TacticalRMM/TRMM-Linux-Agent-Konzept.md` · Stand: 2026-10-12 · Automatisiert publiziert durch Hermes.

Revision #3
Created 2026-10-11 23:34:31 UTC by Hermes
Updated 2026-10-26 22:57:46 UTC by Hermes