

Linux Agent Installation & Management

? Zweck

Dieses Dokument beschreibt die Installation, Aktualisierung und Deinstallation des Linux-Agents über das bereitgestellte `linuxrmm.sh`-Skript. Das Skript bindet Linux-Systeme an TacticalRMM / MeshCentral an und automatisiert die gesamte Einrichtung.

? Voraussetzungen

- Root- oder sudo-Zugriff auf das Zielsystem
- Unterstützte Distributionen: **Debian / Ubuntu** (getestet)
- Internetzugang zum TacticalRMM- und MeshCentral-Server
- Abhängigkeiten: `curl`, `wget`, `tar`, `systemd`

?? Parameterquellen in TacticalRMM

Da die Funktion „**Add Linux Agent**“ in unserer Umgebung derzeit **nicht aktiv** ist, müssen die Parameter **manuell** eingetragen werden.

Die folgenden Werte können direkt im TacticalRMM-Dashboard abgelesen werden:

Parameter	Beschreibung	Wo im TacticalRMM zu finden
<code><MESH_URL></code>	Download-Link des Mesh Agents (Linux Installer)	TacticalRMM → <i>Settings</i> → <i>Global Settings</i> → <i>MeshCentral</i> → Abschnitt Mesh Settings → Eintrag " Linux Agent Download URL "
<code><API_URL></code>	URL zur TacticalRMM API	TacticalRMM → <i>Settings</i> → <i>Global Settings</i> → Feld " Base API URL " (z. B. <code>https://rmm.example.com</code>)

Parameter	Beschreibung	Wo im TacticalRMM zu finden
<CLIENT_ID>	Eindeutige Kunden-ID	TacticalRMM → <i>Clients</i> → Kunde auswählen → Browser-Adresszeile (/clients/3) oder Tooltip „Client ID“
<SITE_ID>	Standort-ID des Kunden	TacticalRMM → <i>Clients</i> → <i>Sites</i> → Site auswählen → ID in der Adresszeile (/sites/1)
<AUTH_KEY>	Installationsschlüssel (Authentifizierung)	TacticalRMM → <i>Settings</i> → <i>API Keys</i> → Kategorie Agent Auth Keys → Neuen Key erstellen oder bestehenden verwenden
<AGENT_TYPE>	Typ der Installation (server / workstation)	Nach internem Standard definieren - dient der Unterscheidung im Dashboard

⚠ Hinweis:

Auth-Keys und API-URLs sind **sensibel** und dürfen **nicht** in öffentlichen oder gemeinsam genutzten Dokumenten gespeichert werden.
Alle Parameter können auch über Administrator-Zugriff direkt aus den TacticalRMM-Einstellungen exportiert werden.

?? Installation

? Syntax

```
sudo bash linuxrmm.sh install <MESH_URL> <API_URL> <CLIENT_ID> <SITE_ID> <AUTH_KEY> <AGENT_TYPE>
```

? Beispiel

```
sudo bash linuxrmm.sh install \  
"https://mesh.example.com/agent-linux64.sh" \  
"https://rmm.example.com" \  
3 1 "abcdefgh123456789" "server"
```

Das Skript lädt den passenden Agent (basierend auf der Systemarchitektur) herunter, konfiguriert ihn automatisch für den Mandanten und startet den Systemdienst.

? Update des Linux-Agents

```
sudo bash linuxrmm.sh update
```

Der laufende Dienst wird gestoppt, aktualisiert und automatisch neu gestartet.
Verbindungen zu TacticalRMM und MeshCentral bleiben erhalten.

? Deinstallation

? Syntax

```
sudo bash linuxrmm.sh uninstall <MESH_FQDN> <MESH_ID>
```

? Beispiel

```
sudo bash linuxrmm.sh uninstall mesh.example.com A1B2C3D4E5
```

Der lokale Dienst und alle Dateien werden entfernt.
Der Eintrag im TacticalRMM-Dashboard bleibt bestehen und kann dort manuell gelöscht werden.

? Status- und Fehlerdiagnose

Befehl	Beschreibung
<code>sudo systemctl status linux-agent</code>	Zeigt den aktuellen Status des Agent-Dienstes
<code>journalctl -u linux-agent -n 50</code>	Zeigt die letzten 50 Log-Zeilen
<code>cat /var/log/linuxrmm/install.log</code>	Zeigt detaillierte Installations- und Laufzeit-Logs
<code>sudo bash linuxrmm.sh update</code>	Erzwingt eine manuelle Aktualisierung

? Ordnerstruktur

Pfad	Inhalt
<code>/opt/linuxrmm/</code>	Hauptverzeichnis des Agents
<code>/opt/linuxrmm/conf/</code>	Konfiguration, Client- und Site-Informationen
<code>/opt/linuxrmm/bin/</code>	Agent-Binary und Startskripte
<code>/var/log/linuxrmm/</code>	Installations- und Laufzeit-Logs

?? Häufige Probleme

Fehler	Ursache / Lösung
<code>unsupported arch</code>	Architektur nicht unterstützt → <code>uname -m</code> prüfen
Dienst startet nicht automatisch	<code>sudo systemctl enable --now linux-agent</code> ausführen
Keine Verbindung zum Dashboard	Firewall / Proxy prüfen
API-Auth-Fehler	Auth-Key abgelaufen → neuen Key generieren

? Best Practice

- Skript immer aus **interner Quelle** oder dem **internen Git-Repository** verwenden
- Keine Auth-Keys oder API-URLs in Klartext speichern
- Nach Installation prüfen, ob der Agent im Dashboard erscheint
- Bei Server-Deployments kann das Skript automatisiert

Revision #6

Created 2025-10-06 22:22:25 UTC by CoreStream Group

Updated 2025-10-06 23:37:59 UTC by CoreStream Group