Tactical RMM

Tactical RMM ist ein Open-Scource Tool zur Fernüberwachung und -verwaltung, das mit Django, Vue und Golang entwickelt wurde. Es verwendet einen in Golang geschriebenen Agenten und ist mit MeshCentral integriert.

Aktualisieren des RMM

Server auf dem neuesten Stand halten

Sie sollten in regelmäßigen Abständen `sudo apt update` und `sudo apt -y` upgrade ausführen, um Ihren Server auf dem neuesten Stand zu halten. Ansonsten sollten Sie keine Änderungen an Ihrem Server vornehmen und alles andere dem Skript update.sh überlassen.

> Versuchen Sie NICHT, Betriebssystem-Upgrades an Ort und Stelle durchzuführen, wie z.B. ein Upgrade von Debian 11 auf 12 an Ort und Stelle. Sie werden Ihren Server zerstören!

Aktualisierung auf die neueste RMM-Version¶

> Machen Sie vor dem Update immer ein Backup rsp. ein Snapshot der aktuellen Installation

Melden Sie sich per SSH an Ihren Server an mit dem Benutzer, den Sie während der Installation angelegt haben.

> Führen Sie niemals Update-Skripte oder -Befehle als Root-Benutzer aus. Das bringt die Berechtigungen durcheinander und macht Ihre Installation kaputt.

Laden Sie das Update-Skript herunter und führen Sie es aus:

```
wget -N https://raw.githubusercontent.com/amidaware/tacticalrmm/master/update.sh
chmod +x update.sh
./update.sh
```

Wenn Sie bereits auf der neuesten Version sind, wird das Update-Skript Sie darüber informieren und sofort zurückkehren.

Sie können die optionale Option --force an das Update-Skript übergeben, um ein Update zu erzwingen, wodurch die Prüfung auf die neueste Version umgangen wird.

```
wget -N https://raw.githubusercontent.com/amidaware/tacticalrmm/master/update.sh
chmod +x update.sh
./update.sh --force
```

Dies ist nützlich für eine verpfuschte Aktualisierung, die möglicherweise nicht vollständig abgeschlossen wurde.

Das Update-Skript korrigiert auch alle Berechtigungen, die während eines verpfuschten Updates durcheinander geraten sind, oder wenn Sie das Update-Skript versehentlich als Root-Benutzer ausgeführt haben.

> Versuchen Sie nicht, MeshCentral manuell auf eine neuere Version zu aktualisieren. Sie sollten dies dem update.sh-Skript überlassen. Die Entwickler testen MeshCentral und stellen sicher, dass die Integration nicht unterbrochen wird, bevor sie die Mesh-Version aktualisieren.

Aktualisieren des Let's Encrypt SSL-Zertifikat

> Derzeit erneuert das Update-Skript Ihr Let's Encrypt-Wildcard-Zertifikat, das alle 3 Monate abläuft, nicht automatisch, da es nicht ohne weiteres möglich ist, dies mit der DNS-TXT-Eintragsmethode zu automatisieren.

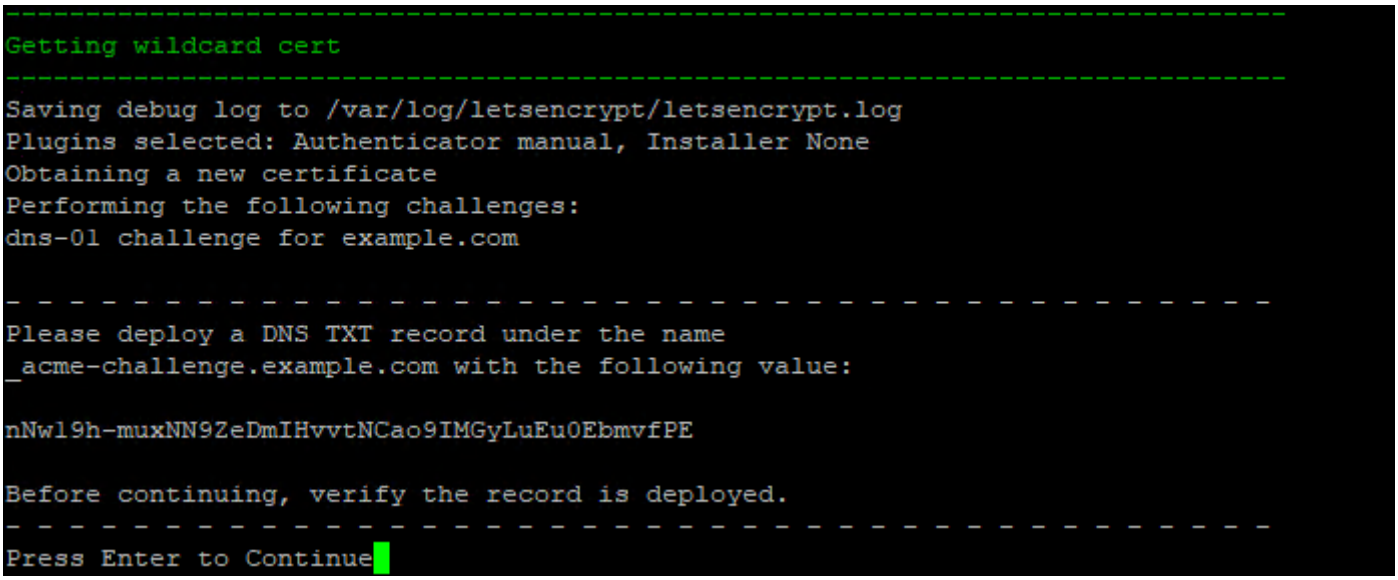
Let's Encrypt Wildcard Zertifikat erneuern

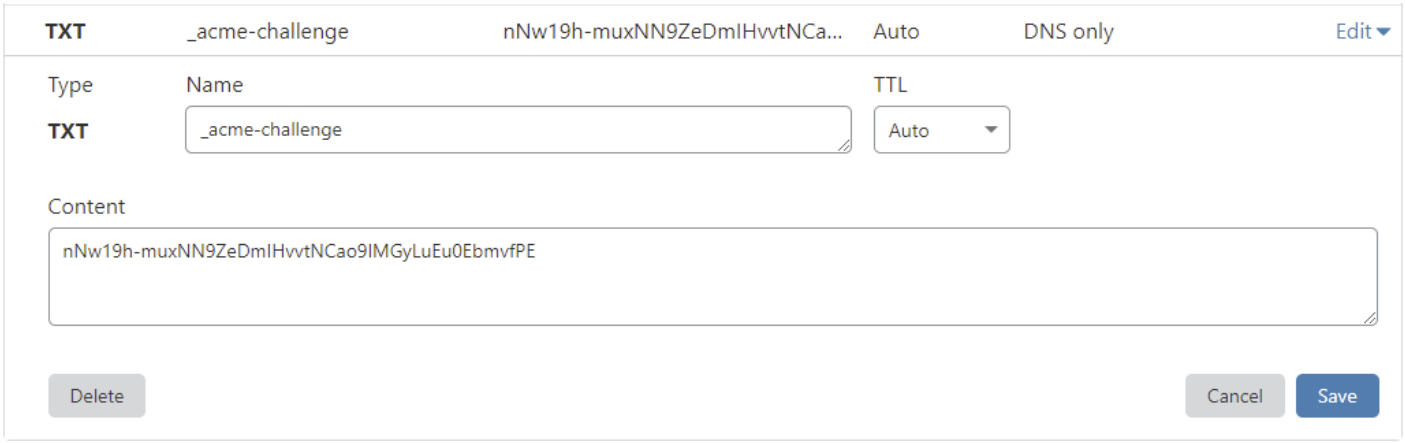
Um Ihre Let's Encrypt Wildcard Cert SSL-Zertifikate zu erneuern und zu aktualisieren, führen Sie den folgenden Befehl aus, wobei Sie `example.com` durch Ihre Domain und `admin@example.com` durch Ihre E-Mail-Adresse ersetzen:

```
sudo certbot certonly --manual -d *.example.com --agree-tos --no-bootstrap --preferred-challenges dns -m admin@example.com --no-eff-email
```

DNS TXT Eintrag aktualisieren

> Die Propagierung von TXT-Einträgen kann je nach DNS-Anbieter zwischen 1 Minute und einigen Stunden dauern.
> Bevor Sie die Eingabetaste drücken, sollten Sie überprüfen, ob der TXT-Datensatz bereitgestellt wurde.
> Mit dem folgenden Befehl können Sie dies schnell überprüfen:
> `dig -t txt _acme-challenge.example.com` (nicht vom TRMM-Server)
> oder testen Sie mit: https://viewdns.info/dnsrecord/ Enter: `_acme-herausforderung.beispiel.com`





Neues Wildcard Zertifikat einbinden

Das erneuerte Zertifikat wird immer noch vollständig in Tactical verwendet. Führen Sie nun das Update Skript.

```
./update.sh --force
```

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
`<MESH_URL>`	Download-Link des Mesh Agents (Linux Installer)	TacticalRMM → *Settings* → *Global Settings* → *MeshCentral* → Abschnitt **Mesh Settings** → Eintrag **"Linux Agent Download URL"**
`<API_URL>`	URL zur TacticalRMM API	TacticalRMM → *Settings* → *Global Settings* → Feld **"Base API URL"** (z. B. `https://rmm.example.com`)
`<CLIENT_ID>`	Eindeutige Kunden-ID	TacticalRMM → *Clients* → Kunde auswählen → Browser-Adresszeile (`/clients/3`) oder Tooltip „Client ID“

Parameter	Beschreibung	Wo im TacticalRMM zu finden
`<SITE_ID>`	Standort-ID des Kunden	TacticalRMM → *Clients* → *Sites* → Site auswählen → ID in der Adresszeile (`/sites/1`)
`<AUTH_KEY>`	Installationsschlüssel (Authentifizierung)	TacticalRMM → *Settings* → *API Keys* → 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
`sudo systemctl status linux-agent`	Zeigt den aktuellen Status des Agent-Dienstes
`journalctl -u linux-agent -n 50`	Zeigt die letzten 50 Log-Zeilen
`cat /var/log/linuxrmm/install.log`	Zeigt detaillierte Installations- und Laufzeit-Logs
`sudo bash linuxrmm.sh update`	Erzwingt eine manuelle Aktualisierung

? Ordnerstruktur

Pfad	Inhalt

`/opt/linuxrmm/`	Hauptverzeichnis des Agents
`/opt/linuxrmm/conf/`	Konfiguration, Client- und Site-Informationen
`/opt/linuxrmm/bin/`	Agent-Binary und Startskripte
`/var/log/linuxrmm/`	Installations- und Laufzeit-Logs

?? Häufige Probleme

Fehler	Ursache / Lösung
`unsupported arch`	Architektur nicht unterstützt → `uname -m` prüfen
Dienst startet nicht automatisch	`sudo systemctl enable --now linux-agent` 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