RustDesk in Tactical RMM integrieren

RustDesk in Tactical RMM integrieren

Diese Anleitung beschreibt die sichere Windows-Bereitstellung von RustDesk über Tactical RMM (TRMM). Ein eigener RustDesk-Server wird vorausgesetzt.

> **Status:** Dienstinstallation, ID-Ermittlung und URL-Aktion stammen aus einem Praxiseinsatz. Das alte Skript war jedoch auf RustDesk 1.2.6 und direkte Änderungen an `RustDesk2.toml` festgelegt. Der aktuelle Ansatz nutzt die dokumentierte Konfigurationszeichenfolge und muss mit jeder freigegebenen Client-Version pilotiert werden.

Empfohlener Ablauf

1. RustDesk als Dienst installieren.
2. Servereinstellungen mit einer RustDesk-Konfigurationszeichenfolge setzen.
3. RustDesk-ID in ein TRMM-Custom-Field schreiben.
4. Eine URL-Aktion **ohne Kennwort** hinterlegen.
5. Unbeaufsichtigten Zugriff über eine geeignete Kennwort- oder Kontorichtlinie absichern.

RustDesk empfiehlt für Server Pro den Custom Client Generator. Für OSS- oder RMM-Rollouts sind ein geprüftes PowerShell-Skript und `--config` die dokumentierte Alternative. Die TRMM-Beispielseite ist für Collector und URL Action nützlich, enthält aber eine veraltete, fest codierte Client-Version.

Voraussetzungen

- Die Konfigurationszeichenfolge stammt aus der RustDesk-Webkonsole oder aus **Settings → Network → Export Server Config**.
- Download-URL und SHA-256-Prüfsumme gehören zur selben freigegebenen Version.
- Das TRMM-Skript läuft als Administrator beziehungsweise `SYSTEM`.
- Geschützte Werte werden nicht im Skript protokolliert.

Installationsskript

Die Werte als TRMM-Parameter oder geschützte Variablen übergeben. Keine produktiven Servernamen, Kennwörter oder Tokens in die Knowledge Base eintragen.

```
param(
    [Parameter(Mandatory = $true)]
    [string]$DownloadUrl,

    [Parameter(Mandatory = $true)]
    [ValidatePattern('^[A-Fa-f0-9]{64}$')]
    [string]$ExpectedSha256,

    [Parameter(Mandatory = $true)]
    [string]$ConfigString
)

$ErrorActionPreference = 'Stop'
$workDir = Join-Path $env:ProgramData 'TacticalRMM\temp\rustdesk'
$installer = Join-Path $workDir 'rustdesk.exe'
New-Item -ItemType Directory -Force -Path $workDir | Out-Null

Invoke-WebRequest -Uri $DownloadUrl -OutFile $installer
$actualSha256 = (Get-FileHash -Path $installer -Algorithm SHA256).Hash
if ($actualSha256 -ne $ExpectedSha256) {
    throw 'RustDesk SHA-256 stimmt nicht überein.'
}

Start-Process -FilePath $installer -ArgumentList '--silent-install' -Wait

$service = Get-Service -Name 'RustDesk' -ErrorAction Stop
if ($service.Status -ne 'Running') {
    Start-Service -Name 'RustDesk'
}
```

```
$client = Join-Path $env:ProgramFiles 'RustDesk\RustDesk.exe'
& $client --config $ConfigString
if ($LASTEXITCODE -ne 0) {
    throw "RustDesk-Konfiguration fehlgeschlagen (Exit-Code $LASTEXITCODE)."
}

Restart-Service -Name 'RustDesk'
Remove-Item -Path $installer -Force
```

Nicht automatisch «latest» herunterladen: Version testen, freigeben, URL und SHA-256 pinnen und erst danach ausrollen.

RustDesk-ID erfassen

In TRMM ein Custom Agent Field `rustdeskid` vom Typ **Text** anlegen. Collector-Skript:

```
$ErrorActionPreference = 'Stop'
$client = Join-Path $env:ProgramFiles 'RustDesk\RustDesk.exe'
if (-not (Test-Path $client)) {
    throw 'RustDesk ist nicht installiert.'
}

$id = (& $client --get-id | Out-String).Trim()
if ([string]::IsNullOrWhiteSpace($id)) {
    throw 'RustDesk-ID konnte nicht ermittelt werden.'
}

$id
```

URL-Aktion

```
rustdesk://connection/new/{{agent.rustdeskid}}
```

Die Anwendung fordert danach die zulässige Authentisierung an.

> **Kein Kennwort in URL oder Custom Field:** Query-Strings, RMM-Felder, Skriptausgaben und Prozessargumente können in Logs oder Browser-Verläufen erscheinen. Kein dauerhaftes Kennwort als `?password=...` speichern. Für unbeaufsichtigten Zugriff vorzugsweise RustDesk-Pro-Konten/Adressbuch, zeitlich begrenzte Zugangsdaten oder einen geeigneten Secrets-Workflow einsetzen.

Abnahme

- `Get-Service RustDesk` meldet `Running`.
- ID-Server und öffentlicher Schlüssel sind korrekt.
- Die ID erscheint im TRMM-Custom-Field.
- Die URL-Aktion öffnet den ausgewählten Agenten.
- Direktverbindung und Relay-Fallback funktionieren aus einem externen Netz.
- Neuinstallation und Update wurden mit der gepinnten Version getestet.
- TRMM-Ausgabe, Skript und Custom Fields enthalten keine Kennwörter oder Tokens.

Quellen

- RustDesk: Client Deployment
- RustDesk: Client Configuration
- Tactical RMM: RustDesk-Integration

Quelle: Obsidian `Knowledge Base/TacticalRMM/RustDesk in Tactical RMM Integrieren.md` · Stand: 2026-10-12 · Automatisiert publiziert durch Hermes.

Revision #3
Created 2026-10-11 23:34:30 UTC by Hermes
Updated 2026-10-26 22:57:45 UTC by Hermes