Runbook: RDP Hardening Guide

Version: 1.0 **Date:** 2025-11-11 **Purpose:** Comprehensive guide to harden Remote Desktop Protocol (RDP) on Windows systems for secure remote access

---

Configuration Custody Notice
Ionic Health engineering exclusively manages all eKVM configuration, firmware, and software changes, including patching operations conducted through the jumper server. LVHN operations is solely responsible for provisioning, hardening, and maintaining the Windows jumper server environment according to the documented requirements.

Network Visibility Scope
This guidance applies strictly to the LVHN-managed jumper server; Ionic Health does not modify or audit other LVHN network segments.

Table of Contents

Overview

Remote Desktop Protocol (RDP) is a Microsoft protocol enabling remote access to Windows systems. Default configurations pose security risks including: - Weak encryption - No multi-factor authentication - Unlimited brute-force attempts - Data exfiltration via clipboard/drive sharing

This runbook implements industry best practices aligned with: - **NIST SP 800-53** (AC-17: Remote Access) - **CIS Benchmarks** for Windows Server - **HIPAA** Technical Safeguards (45 CFR § 164.312)

Pre-Hardening Assessment

Current Configuration Audit

```powershell
# Save as: RDP-Assessment.ps1
Write-Host "=== RDP Security Assessment ===" -ForegroundColor Cyan

# 1. Check if RDP is enabled
$rdpEnabled = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Serve
if ($rdpEnabled -eq 0) {
    Write-Host "[INFO] RDP is ENABLED" -ForegroundColor Yellow
} else {
    Write-Host "[INFO] RDP is DISABLED" -ForegroundColor Green
}

# 2. Check NLA requirement
$nlaEnabled = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Serve
Write-Host "[NLA] UserAuthentication = $nlaEnabled (0=Disabled, 1=Enabled)" -ForegroundColor

# 3. Check encryption level
$encLevel = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\
$encLevels = @{3="High (128-bit)"; 2="Client Compatible"; 1="Low"}
Write-Host "[Encryption] MinEncryptionLevel = $encLevel ($($encLevels[$encLevel]))" -Foregro

# 4. Check TLS version
$securityLayer = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Se
$secLayers = @{0="RDP Security"; 1="Negotiate"; 2="TLS 1.0"; 3="TLS 1.2+"}
Write-Host "[TLS] SecurityLayer = $securityLayer ($($secLayers[$securityLayer]))" -Foregroun

# 5. Check account lockout policy
$lockoutThreshold = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\RemoteA
if ($lockoutThreshold) {
    Write-Host "[Lockout] Threshold = $lockoutThreshold attempts" -ForegroundColor Green
} else {
    Write-Host "[Lockout] No account lockout configured" -ForegroundColor Red
}

# 6. Check firewall rule
$fwRule = Get-NetFirewallRule -Name "RemoteDesktop-UserMode-In-TCP" -ErrorAction SilentlyCon
if ($fwRule -and $fwRule.Enabled -eq "True") {
    Write-Host "[Firewall] RDP rule enabled" -ForegroundColor Yellow
    Get-NetFirewallAddressFilter -AssociatedNetFirewallRule $fwRule | Select-Object RemoteAd
} else {
    Write-Host "[Firewall] RDP rule disabled or not found" -ForegroundColor Green
}
```

```
Write-Host "`n=== Assessment Complete ===" -ForegroundColor Cyan
```

Save output for compliance records:

```
.\RDP-Assessment.ps1 | Out-File -FilePath "C:\Evidence\RDP-Assessment-$(Get-Date -Format 'yy
```

Network Level Authentication (NLA)

Purpose: Require authentication BEFORE establishing RDP session (prevents pre-auth exploits).

Enable NLA

```
# Registry method
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\R
                 -Name "UserAuthentication" -Value 1

# Via WMI (alternative)
(Get-WmiObject -Class "Win32_TSGeneralSetting" -Namespace root\cimv2\terminalservices -Filte
```

Group Policy Method

1. Open Group Policy Editor: `gpedit.msc`
2. Navigate to:

   ```
   Computer Configuration
      Administrative Templates
          Windows Components
              Remote Desktop Services
                  Remote Desktop Session Host
                      Security
   ```

3. Enable: **"Require user authentication for remote connections by using Network Level Authentication"**
4. Apply: `gpupdate /force`

Verify NLA

```
$nla = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinSt
if ($nla -eq 1) {
    Write-Host "[PASS] NLA is enabled" -ForegroundColor Green
} else {
    Write-Host "[FAIL] NLA is NOT enabled" -ForegroundColor Red
}
```

Compliance Note: NLA is mandatory per CIS Benchmark 18.9.62.3.3.1.

TLS Configuration

Purpose: Enforce TLS 1.2+ encryption; disable weak protocols (SSL 3.0, TLS 1.0, TLS 1.1).

Set Security Layer to TLS

```
# 0 = RDP Security Layer (weak, deprecated)
# 1 = Negotiate (fallback to RDP if TLS unavailable)
# 2 = TLS (enforce TLS)
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\R
                 -Name "SecurityLayer" -Value 2

# Restart Terminal Services
Restart-Service TermService -Force
```

Disable Weak TLS Versions

```
# Disable SSL 3.0
New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr

# Disable TLS 1.0
New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr

# Disable TLS 1.1
New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr

# Enable TLS 1.2
New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr

# Enable TLS 1.3 (Windows Server 2022+)
New-Item -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Pr
```

Set Minimum Encryption Level

```
# 1 = Low (56-bit)
# 2 = Client Compatible (varies)
# 3 = High (128-bit minimum)
```

```
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\R
                 -Name "MinEncryptionLevel" -Value 3
```

Verify TLS Configuration

```
$secLayer = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\
$encLevel = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\

if ($secLayer -eq 2 -and $encLevel -eq 3) {
    Write-Host "[PASS] TLS enforced with 128-bit encryption" -ForegroundColor Green
} else {
    Write-Host "[FAIL] Weak encryption configuration" -ForegroundColor Red
}
```

Account Lockout Policies

Purpose: Prevent brute-force password attacks by locking accounts after failed attempts.

Configure Lockout Policy (Local)

```
# Set account lockout threshold (5 failed attempts)
net accounts /lockoutthreshold:5

# Set lockout duration (30 minutes)
net accounts /lockoutduration:30

# Reset lockout counter after (30 minutes)
net accounts /lockoutwindow:30

# Verify
net accounts
```

Configure via Group Policy (Domain)

1. Open Group Policy Editor: `gpmc.msc`
2. Navigate to:

   ```
   Computer Configuration
     Policies
         Windows Settings
             Security Settings
                 Account Policies
                     Account Lockout Policy
   ```

3. Set:

- **Account lockout threshold:** 5 invalid logon attempts
- **Account lockout duration:** 30 minutes
- **Reset account lockout counter after:** 30 minutes

Monitor Lockout Events

```
# Query recent lockout events (Event ID 4740)
Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4740; StartTime=(Get-Date).AddDays(-1
    Select-Object TimeCreated, @{Name='Account';Expression={$_.Properties[0].Value}}, @{Name
    Format-Table -AutoSize
```

Legal Banner

Purpose: Display legal notice upon RDP connection (satisfies "authorized use only" requirement).

Configure Legal Notice

```
# Set caption (title)
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" `
                 -Name "LegalNoticeCaption" `
                 -Value "Authorized Access Only" -Type String

# Set message text
$legalText = @"
This computer system is the property of Lehigh Valley Health Network (LVHN).
It is for authorized use only.

Users (authorized or unauthorized) have no explicit or implicit expectation of privacy.

Any or all uses of this system and all files on this system may be intercepted, monitored, r

By using this system, the user consents to such interception, monitoring, recording, copying

Unauthorized or improper use of this system may result in administrative disciplinary actior

By continuing to use this system you indicate your awareness of and consent to these terms a

LOG OFF IMMEDIATELY if you do not agree to the conditions stated in this warning.
"@

Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" `
                 -Name "LegalNoticeText" `
                 -Value $legalText -Type String
```

Verify Legal Banner

```
# Check registry
$caption = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies
$text = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Sy

if ($caption -and $text) {
    Write-Host "[PASS] Legal banner configured" -ForegroundColor Green
    Write-Host "Caption: $caption"
    Write-Host "Text (first 100 chars): $($text.Substring(0,100))..."
} else {
    Write-Host "[FAIL] Legal banner NOT configured" -ForegroundColor Red
}

# Test: RDP to system and verify banner appears before login
```

Device Redirection Controls

Purpose: Prevent data exfiltration via clipboard, drive mapping, printer redirection.

Disable Clipboard Redirection

```
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableClip" -Value 1 -Type DWord -Force
```

Disable Drive Mapping

```
# Disable all drive redirection (most secure)
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCdm" -Value 1 -Type DWord -Force
```

Note: For eKVM update process, drive mapping may be required during maintenance window. Enable temporarily via GPO:

```
# TEMPORARILY enable drive mapping (change window only)
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCdm" -Value 0 -Type DWord -Force

# Revert post-window
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCdm" -Value 1 -Type DWord -Force
```

Disable Printer Redirection

```
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCpm" -Value 1 -Type DWord -Force
```

Disable COM Port Redirection

```
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableLPT" -Value 1 -Type DWord -Force
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCcm" -Value 1 -Type DWord -Force
```

Disable Audio Redirection

```
# 0 = Allow redirection
# 1 = Do not allow
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCam" -Value 1 -Type DWord -Force
```

Verify Redirection Settings

```
$rdpPath = "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services"

$settings = @{
    "Clipboard" = (Get-ItemProperty -Path $rdpPath -ErrorAction SilentlyContinue).fDisableCl
    "Drive Mapping" = (Get-ItemProperty -Path $rdpPath -ErrorAction SilentlyContinue).fDisab
    "Printer" = (Get-ItemProperty -Path $rdpPath -ErrorAction SilentlyContinue).fDisableCpm
    "COM Port" = (Get-ItemProperty -Path $rdpPath -ErrorAction SilentlyContinue).fDisableLPT
    "Audio" = (Get-ItemProperty -Path $rdpPath -ErrorAction SilentlyContinue).fDisableCam
}

Write-Host "=== RDP Redirection Status ===" -ForegroundColor Cyan
foreach ($setting in $settings.GetEnumerator()) {
    $status = if ($setting.Value -eq 1) {"Disabled "} else {"Enabled "}
    Write-Host "$($setting.Key): $status"
}
```

Session Configuration

Set Idle Session Timeout

```
# Disconnect idle sessions after 15 minutes
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "MaxIdleTime" -Value 900000 -Type DWord -Force  # milliseconds

# End disconnected sessions after 1 hour
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "MaxDisconnectionTime" -Value 3600000 -Type DWord -Force
```

Limit Concurrent Sessions

```
# Limit to 2 concurrent RDP sessions per user
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server" `
                 -Name "MaxInstanceCount" -Value 2 -Type DWord -Force
```

Require Secure RPC

```
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fEncryptRPCTraffic" -Value 1 -Type DWord -Force
```

Firewall Rules

Restrict RDP Access by IP

```
# Remove default RDP rule (allows any source)
Remove-NetFirewallRule -Name "RemoteDesktop-UserMode-In-TCP" -ErrorAction SilentlyContinue

# Create scoped rule (jumper server only)
$jumperIP = "10.50.100.10"
New-NetFirewallRule -Name "RDP-Jumper-Only-TCP" `
                    -DisplayName "Remote Desktop (Jumper Server Only)" `
                    -Enabled True `
                    -Direction Inbound `
                    -Protocol TCP `
                    -LocalPort 3389 `
                    -RemoteAddress $jumperIP `
                    -Action Allow `
                    -Profile Domain `
                    -Program "%SystemRoot%\system32\svchost.exe"

# Verify rule
Get-NetFirewallRule -Name "RDP-Jumper-Only-TCP" | Get-NetFirewallAddressFilter
```

Block RDP from Internet

```
# Explicit deny from Internet (add to edge firewall)
New-NetFirewallRule -Name "RDP-Block-Internet" `
                    -DisplayName "Block RDP from Internet" `
                    -Enabled True `
                    -Direction Inbound `
                    -Protocol TCP `
                    -LocalPort 3389 `
                    -RemoteAddress Internet `
                    -Action Block `
                    -Profile Public
```

Test Firewall Rules

```
# From authorized IP (should succeed)
Test-NetConnection -ComputerName ekvm-device-01 -Port 3389

# From unauthorized IP (should fail)
# Run this from a different workstation outside allowed range
```

Audit and Logging

Enable RDP Audit Policies

```
# Enable logon/logoff auditing
auditpol /set /subcategory:"Logon" /success:enable /failure:enable
auditpol /set /subcategory:"Logoff" /success:enable
auditpol /set /subcategory:"Account Lockout" /success:enable /failure:enable
auditpol /set /subcategory:"Other Logon/Logoff Events" /success:enable /failure:enable

# Verify
auditpol /get /category:"Logon/Logoff"
```

Enable RDP-Specific Logs

```
# Enable TerminalServices-LocalSessionManager log
wevtutil set-log "Microsoft-Windows-TerminalServices-LocalSessionManager/Operational" /enabl

# Enable TerminalServices-RemoteConnectionManager log
wevtutil set-log "Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational" /e

# Increase log sizes (1 GB each)
wevtutil set-log "Security" /maxsize:1073741824
wevtutil set-log "Microsoft-Windows-TerminalServices-LocalSessionManager/Operational" /maxsi
```

Key Event IDs to Monitor

Event ID	Log Source	Description	Action
4624	Security	Successful logon (Type 10 = RDP)	Track authorized access
4625	Security	Failed logon	Alert on >5 failures (brute force)
4634	Security	Logon session terminated	Track session duration

Event ID	Log Source	Description	Action
4672	Security	Special privileges assigned	Track admin access
4778	Security	Session reconnected	Track session resumption
4779	Security	Session disconnected	Track disconnections
21	TerminalServices-RemoteConnectionManager	Remote Desktop Services: Session logon succeeded	Correlation with 4624
22	TerminalServices-LocalSessionManager	Remote Desktop Services: Shell start notification received	Session initialization
24	TerminalServices-LocalSessionManager	Remote Desktop Services: Session has been disconnected	Track idle disconnects
25	TerminalServices-LocalSessionManager	Remote Desktop Services: Session reconnection succeeded	Track reconnect patterns
39	TerminalServices-RemoteConnectionManager	Session has been disconnected, reason code	Reason code analysis
40	TerminalServices-RemoteConnectionManager	Session has been disconnected by session	Admin termination tracking

Query Recent RDP Logins

```
# Successful RDP logons (last 24 hours)
Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4624
    StartTime=(Get-Date).AddDays(-1)
} | Where-Object {$_.Properties[8].Value -eq 10} |  # LogonType 10 = RDP
    Select-Object TimeCreated,
                  @{Name='Username';Expression={$_.Properties[5].Value}},
                  @{Name='SourceIP';Expression={$_.Properties[18].Value}} |
    Format-Table -AutoSize
```

Failed RDP Attempts

```
# Failed RDP logons (brute force detection)
Get-WinEvent -FilterHashtable @{
    LogName='Security'
```

```
    ID=4625
    StartTime=(Get-Date).AddHours(-1)
} | Where-Object {$_.Properties[10].Value -eq 10} |
    Group-Object {$_.Properties[5].Value} |
    Where-Object {$_.Count -gt 5} |
    Select-Object @{Name='Account';Expression={$_.Name}}, Count |
    Format-Table -AutoSize
```

Validation

RDP Hardening Validation Script

```
# Save as: Validate-RDP-Hardening.ps1
Write-Host "=== RDP Hardening Validation ===" -ForegroundColor Cyan

$results = @()

# Test 1: NLA Enabled
$nla = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinSt
$results += [PSCustomObject]@{
    Test = "Network Level Authentication"
    Expected = 1
    Actual = $nla
    Result = if ($nla -eq 1) {"PASS"} else {"FAIL"}
}

# Test 2: TLS Enforced
$secLayer = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\
$results += [PSCustomObject]@{
    Test = "TLS Security Layer"
    Expected = 2
    Actual = $secLayer
    Result = if ($secLayer -eq 2) {"PASS"} else {"FAIL"}
}

# Test 3: High Encryption
$encLevel = (Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\
$results += [PSCustomObject]@{
    Test = "128-bit Encryption"
    Expected = 3
    Actual = $encLevel
    Result = if ($encLevel -eq 3) {"PASS"} else {"FAIL"}
}

# Test 4: Legal Banner
```

```powershell
$banner = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\
$results += [PSCustomObject]@{
    Test = "Legal Notice Banner"
    Expected = "Configured"
    Actual = if ($banner) {"Configured"} else {"Not Configured"}
    Result = if ($banner) {"PASS"} else {"FAIL"}
}

# Test 5: Clipboard Disabled
$clipboard = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal
$results += [PSCustomObject]@{
    Test = "Clipboard Redirection Disabled"
    Expected = 1
    Actual = $clipboard
    Result = if ($clipboard -eq 1) {"PASS"} else {"WARN"}
}

# Test 6: Drive Mapping Disabled
$drives = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Ser
$results += [PSCustomObject]@{
    Test = "Drive Redirection Disabled"
    Expected = 1
    Actual = $drives
    Result = if ($drives -eq 1) {"PASS"} else {"WARN (may be needed for updates)"}
}

# Test 7: Account Lockout
$lockout = (Get-ItemProperty -Path "HKLM:\SAM\SAM\Domains\Account" -ErrorAction SilentlyCont
$results += [PSCustomObject]@{
    Test = "Account Lockout Policy"
    Expected = "Configured"
    Actual = if ($lockout) {"Configured"} else {"Check via 'net accounts'"}
    Result = "INFO"
}

# Test 8: Firewall Rule
$fwRule = Get-NetFirewallRule -Name "RDP-Jumper-Only-TCP" -ErrorAction SilentlyContinue
$results += [PSCustomObject]@{
    Test = "Scoped Firewall Rule"
    Expected = "Exists"
    Actual = if ($fwRule) {"Exists"} else {"Default rule"}
    Result = if ($fwRule) {"PASS"} else {"WARN"}
}

# Display results
$results | Format-Table -AutoSize
```

```powershell
$failCount = ($results | Where-Object {$_.Result -eq "FAIL"}).Count
if ($failCount -eq 0) {
    Write-Host "`n[SUCCESS] All critical tests PASSED" -ForegroundColor Green
} else {
    Write-Host "`n[WARNING] $failCount test(s) FAILED" -ForegroundColor Red
}

# Export results
$results | Export-Csv -Path "C:\Evidence\RDP-Hardening-Validation-$(Get-Date -Format 'yyyyMM
Write-Host "Results saved to: C:\Evidence\RDP-Hardening-Validation-$(Get-Date -Format 'yyyyM
```

Run Validation

```
.\Validate-RDP-Hardening.ps1
```

Troubleshooting

Issue: Cannot Connect After Enabling NLA

Symptom: "The remote computer requires Network Level Authentication, which your computer does not support."

Resolution:

```powershell
# On client workstation:
# 1. Verify RDP client version (must be 6.0+)
mstsc /?  # Check version

# 2. Enable CredSSP on client
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Lsa" -Name "DisableRestricted

# 3. Temporarily disable NLA (if client cannot be upgraded)
# On eKVM:
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\R
#   Only for testing; re-enable NLA for production
```

Issue: Certificate Errors with TLS

Symptom: "The certificate or associated chain is not valid."

Resolution:

```powershell
# Generate new self-signed certificate
$cert = New-SelfSignedCertificate -DnsName "ekvm-device-01.lvhn.local" `
                                  -CertStoreLocation "Cert:\LocalMachine\My" `
                                  -KeySpec KeyExchange `
```

```
                                    -NotAfter (Get-Date).AddYears(5)

# Bind certificate to RDP
wmic /namespace:\\root\cimv2\TerminalServices PATH Win32_TSGeneralSetting Set SSLCertificate

# Restart Terminal Services
Restart-Service TermService -Force
```

Issue: Connection Slow After Hardening

Symptom: RDP sessions take >30 seconds to establish.

Resolution:

```
# Disable bandwidth optimization (if on high-speed LAN)
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fClientDisableUDP" -Value 0  # Allow UDP for faster connections

# Disable printer mapping (speeds up logon)
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services" `
                 -Name "fDisableCpm" -Value 1
```

Related Documents

- MOP: eKVM Remote Update
- Runbook: WinRM Setup
- ADR-004: RDP as Primary Protocol
- CIS Microsoft Windows Server 2019 Benchmark
- NIST SP 800-53 AC-17

Document Owner: Ionic Engineering Team **Last Updated:** 2025-11-11 **Review Cycle:** Quarterly or after security incidents