

Method of Procedure (MOP) – eKVM Remote Update

Document Control - Version: 1.0 - **Date:** 2025-11-11 - **Classification:** Internal – Operational Procedure - **Owner:** Ionic Engineering Team - **Approver:** LVHN IT Operations Manager - **Review Cycle:** Quarterly or after each significant incident

Configuration Custody Notice

Ionic Health engineering exclusively manages all eKVM configuration, firmware, and software changes. LVHN operations is solely responsible for provisioning, hardening, and maintaining the Windows jumper server environment. Any activity outside these custody boundaries requires written approval from both teams.

Network Visibility Scope

Maintenance activities assume LVHN control stops at the jumper server boundary; Ionic Health neither accesses nor manages LVHN internal networks beyond the outbound route to eKVM devices.

Table of Contents

1. Purpose and Scope
2. Prerequisites and Validation
3. Roles and Responsibilities
4. Pre-Window Activities (H-24 to H-0)
5. Execution Phase (H+0 to H+2)
6. Post-Window Activities (H+1 to H+4)
7. Rollback Procedure
8. Troubleshooting
9. Evidence Collection
10. Appendix A: Command Reference
11. Appendix B: Event Log Reference
12. Appendix C: Contact Information

1. Purpose and Scope

1.1 Purpose

This Method of Procedure (MOP) provides step-by-step instructions for performing remote firmware and software updates on Ionic eKVM devices within the LVHN environment using the Windows jumper server architecture.

1.2 Scope

- **In Scope:**
 - Firmware updates (eKVM-Firmware-*.exe)
 - Driver updates (eKVM-Drivers-*.exe)
 - Application software updates (nCommand Lite components)
 - Single-device updates and batch updates (up to 10 devices per window)
- **Out of Scope:**
 - eKVM hardware replacement or physical maintenance
 - Network infrastructure changes (VLAN, routing)
 - Updates to jumper server itself (separate MOP)
 - Emergency/unplanned updates (see Emergency MOP)

1.3 Change Management

All updates must be performed under an approved Change Request: - **Standard Change:** Pre-approved updates during maintenance windows (firmware patches, minor updates) - **Normal Change:** Requires CAB approval (major version upgrades, configuration changes) - **Emergency Change:** Reserved for critical security patches (expedited approval)

Change Ticket Required Fields: - Change number (e.g., CHG0012345) - Target eKVM device(s) (hostname, IP, location) - Update version (e.g., eKVM Firmware v3.2.1) - Scheduled window (start/end timestamps) - Operator names (first.last@lvhn.org) - Rollback plan reference (section 7 of this MOP)

2. Prerequisites and Validation

2.1 Mandatory Prerequisites

| # | Prerequisite | Validation Command | Expected Result | Owner |
|---|---------------------------------|---|------------------------|---------|
| 1 | Change ticket approved | Review ServiceNow/change management system | Status = "Approved" | LVHN IT |
| 2 | Jumper server accessible | Test-NetConnection -ComputerName jumper-server-01.lvhn.local -Port 3389 | TcpTestSucceeded: True | LVHN IT |
| 3 | eKVM RDP accessible from jumper | Test-NetConnection -ComputerName <eKVM-hostname> -Port 3389 | TcpTestSucceeded: True | LVHN IT |

| # | Prerequisite | Validation Command | Expected Result | Owner |
|---|-------------------------------------|---|---|------------|
| 4 | JIT account provisioned | Get-ADUser -Identity firstname.lastname.admin | Account exists, enabled | LVHN IT |
| 5 | Binary available (HTTPS or local) | Invoke-WebRequest -Uri <CDN-URL> -Method Head OR local file exists | Status 200 or file present | Ionic |
| 6 | Official SHA-256 hash published | Check Ionic release portal | Hash value visible, copied to clipboard | Ionic |
| 7 | Network ACLs opened (jumper → eKVM) | Test-NetConnection -ComputerName <eKVM> -Port 5985 (WinRM) | TcpTestSucceeded: True | LVHN IT |
| 8 | SIEM ingestion active | Query SIEM for recent logs from jumper + eKVM | Logs present within last 15 min | LVHN IT |

2.2 Pre-Window Validation Script

Run this PowerShell script at H-2 (two hours before window):

```
# Save as: Validate-Prerequisites.ps1
param(
    [string]$eKVMHostname,
    [string]$JumperServer = "jumper-server-01.lvhnl.local"
)

Write-Host "=== Pre-Window Validation ===" -ForegroundColor Cyan

# Test 1: Jumper Server RDP
$test1 = Test-NetConnection -ComputerName $JumperServer -Port 3389 -WarningAction SilentlyContinue
if ($test1.TcpTestSucceeded) {
    Write-Host "[PASS] Jumper Server RDP (3389) reachable" -ForegroundColor Green
} else {
    Write-Host "[FAIL] Jumper Server RDP not reachable" -ForegroundColor Red
    exit 1
}

# Test 2: eKVM RDP from jumper
$test2 = Test-NetConnection -ComputerName $eKVMHostname -Port 3389 -WarningAction SilentlyContinue
if ($test2.TcpTestSucceeded) {
    Write-Host "[PASS] eKVM RDP (3389) reachable from jumper" -ForegroundColor Green
} else {
```

```

        Write-Host "[FAIL] eKVM RDP not reachable" -ForegroundColor Red
        exit 1
    }

    # Test 3: WinRM
    $test3 = Test-NetConnection -ComputerName $eKVMHostname -Port 5985 -WarningAction SilentlyContinue
    if ($test3.TcpTestSucceeded) {
        Write-Host "[PASS] eKVM WinRM (5985) reachable" -ForegroundColor Green
    } else {
        Write-Host "[WARN] eKVM WinRM not reachable (fallback to SMB)" -ForegroundColor Yellow
    }

    Write-Host "`n=== Validation Complete ===" -ForegroundColor Cyan
    Write-Host "All critical tests passed. Proceed with GO/NO-GO decision." -ForegroundColor Green

GO/NO-GO Decision: If any [FAIL] result, do NOT proceed. Escalate to
    LVHN IT and Ionic.

```

3. Roles and Responsibilities

| Role | Name | Contact | Responsibilities |
|-----------------------------|----------------------|-----------------------------|-------------------------------------|
| Update Operator | [Firstname Lastname] | firstname.lastname@lvhn.com | Execute MOP steps; collect evidence |
| LVHN IT Contact | [Firstname Lastname] | phone: XXX-XXX-XXXX | Network/access troubleshooting |
| Ionic Technical Lead | [Firstname Lastname] | email@ionic.com | eKVM technical support |
| Change Manager | [Firstname Lastname] | changemanager@lvhn.com | Approve change ticket; escalation |

Bridge Line: [Conference bridge number] (active during window)

4. Pre-Window Activities (H-24 to H-0)

4.1 H-24: Confirm Scope and Prepare

Duration: 30 minutes

- Review Change Ticket**
 - Verify approval status

- Confirm target device list
 - Note any special instructions or dependencies
2. **Download Update Binary (if using HTTPS method)**
 - Log in to Ionic release portal: <https://downloads.ionic.com>
 - Download binary to secure location: C:\SecureDownloads\eKVM\
 - Copy official SHA-256 hash to C:\SecureDownloads\eKVM\SHA256.txt
 3. **Communicate to Stakeholders**
 - Email LVHN IT Operations: “Change CHG0012345 proceeding as scheduled”
 - Notify clinical engineering (no impact expected, but awareness)

4.2 H-8: Network Preparation

Duration: 15 minutes

LVHN IT Responsibilities:

1. Open Network ACLs (Jumper → eKVM)

```
# Example Palo Alto CLI (adapt to your firewall)
configure
set rulebase security rules "Maint-Jumper-to-eKVM-CHG0012345" \
  source-zone internal \
  destination-zone ekvm-zone \
  source-address jumper-server-01 \
  destination-address ekvm-device-01 \
  application [ rdp winrm-http winrm-https microsoft-smb ] \
  action allow \
  schedule "2025-11-11-1800-to-2200"
commit
```

2. Validate ACL

```
Test-NetConnection -ComputerName ekvm-device-01 -Port 3389
Test-NetConnection -ComputerName ekvm-device-01 -Port 5985
```

4.3 H-4: Provision Access

Duration: 15 minutes

LVHN IT Responsibilities:

1. Create JIT Account

```
# Active Directory
New-ADUser -Name "firstname.lastname.admin" `
  -SamAccountName "firstname.lastname.admin" `
  -UserPrincipalName "firstname.lastname.admin@lvhn.local" `
  -AccountPassword (ConvertTo-SecureString "TempPassword123!" -AsPlainText -Force) `
  -Enabled $true `
```

```
-PasswordNeverExpires $false `
-ChangePasswordAtLogon $true
```

```
# Add to local admin on jumper server (time-boxed)
```

```
Add-LocalGroupMember -Group "Administrators" -Member "LVHN\firstname.lastname.admin"
```

2. Configure eKVM Local Admin (if not using domain account)

```
# Via existing RDP session or Group Policy
```

```
net user ekvm-maint-CHG0012345 "ComplexP@sswOrd!" /add
```

```
net localgroup Administrators ekvm-maint-CHG0012345 /add
```

3. Test Account Login

- RDP to jumper-server-01 using JIT account
- Verify MFA prompt (if configured)
- Logout (do not leave session open)

4.4 H-2: Dry Run and GO/NO-GO

Duration: 30 minutes

1. Run Pre-Window Validation Script (section 2.2)

```
.\Validate-Prerequisites.ps1 -eKVMHostname ekvm-device-01
```

2. Test File Transfer Path

- Create 10 MB test file on jumper:

```
$testFile = New-Object byte[] (10MB)
(New-Object Random).NextBytes($testFile)
[System.IO.File]::WriteAllBytes("C:\Staging\test-10MB.bin", $testFile)
```

- Transfer to eKVM via WinRM:

```
$session = New-PSSession -ComputerName ekvm-device-01 -Credential (Get-Credential)
Copy-Item -Path "C:\Staging\test-10MB.bin" -Destination "C:\Temp\" -ToSession $session
Remove-PSSession $session
```

- Verify file arrived:

```
Test-Path "\\ekvm-device-01\C$\Temp\test-10MB.bin"
```

3. SIEM Validation

- Generate test event (failed login attempt):

```
# Intentional failed RDP login to generate Event ID 4625
mstsc /v:ekvm-device-01 # Enter wrong password
```

- Query SIEM: Verify Event ID 4625 appears within 5 minutes
- If no SIEM ingestion, STOP and escalate to LVHN IT

4. GO/NO-GO Decision

- **GO Criteria:**
 - All validation tests pass
 - JIT account works
 - File transfer successful
 - SIEM ingestion confirmed
 - No conflicting changes scheduled
- **NO-GO Criteria:**
 - Any critical test fails
 - Network connectivity issues
 - SIEM not ingesting logs
 - Change approval pending

Decision: Email to changemanager@lvhn.org and Ionic technical lead:
“GO” or “NO-GO for CHG0012345”

5. Execution Phase (H+0 to H+2)

5.1 Step 1: Connect to Jumper Server

Duration: 5 minutes

1. Establish SSL VPN Connection

- Launch LVHN VPN client
- Authenticate with MFA (username + token)
- Verify VPN connected: `ipconfig` shows VPN adapter

2. RDP to Jumper Server

```
mstsc /v:jumper-server-01.lvhn.local /f
```

- Username: LVHN\firstname.lastname.admin
- Password: [Your JIT password]
- MFA: [Token code if prompted]

3. Verify Legal Banner Displayed

- Screenshot the legal notice
- Save to evidence folder: `C:\Evidence\CHG0012345\01_Legal_Banner.png`

4. Open PowerShell as Administrator

- Start → Windows PowerShell → Run as Administrator
- Enable transcript logging:

```
Start-Transcript -Path "C:\Evidence\CHG0012345\transcript-$(Get-Date -Format 'yyyyMMdd-HH:mm:ss').txt
```

5.2 Step 2: Obtain Update Binary

Duration: 10 minutes (HTTPS) or 5 minutes (RDP copy)

Option A: HTTPS Download (Preferred)

```
# Navigate to staging directory
cd C:\Staging\eKVM
New-Item -ItemType Directory -Path "CHG0012345" -Force
cd CHG0012345

# Download binary
$url = "https://downloads.ionic.com/ekvm/v3.2.1/eKVM-Firmware-v3.2.1.exe"
$output = ".\eKVM-Firmware-v3.2.1.exe"
Invoke-WebRequest -Uri $url -OutFile $output -UseBasicParsing

# Verify download completed
if (Test-Path $output) {
    Write-Host "[SUCCESS] Binary downloaded: $((Get-Item $output).Length / 1MB) MB" -ForegroundColor Green
} else {
    Write-Host "[ERROR] Download failed" -ForegroundColor Red
    exit 1
}
```

Option B: RDP Drive Mapping (Fallback)

```
# Verify operator's local drive is mapped
Get-PSDrive | Where-Object {$_.Provider -match "FileSystem" -and $_.Root -match "tsclient"}

# Copy from operator workstation
Copy-Item -Path "\\tsclient\C\Downloads\eKVM-Firmware-v3.2.1.exe" `
    -Destination "C:\Staging\eKVM\CHG0012345\" -Verbose
```

5.3 Step 3: Verify Binary Integrity (Jumper Server)

Duration: 2 minutes

```
cd C:\Staging\eKVM\CHG0012345

# Expected hash (from Ionic release notes)
$expectedHash = "a1b2c3d4e5f6789012345678901234567890abcdef1234567890abcdef123456"

# Compute actual hash
$actualHash = (Get-FileHash -Path ".\eKVM-Firmware-v3.2.1.exe" -Algorithm SHA256).Hash

# Compare
if ($actualHash -eq $expectedHash) {
    Write-Host "[PASS] SHA-256 verification PASSED" -ForegroundColor Green
    $actualHash | Out-File -FilePath "C:\Evidence\CHG0012345\01_Hashes_Pre_Transfer.txt"
```



```

} else {
    Write-Host "[FAIL] SHA-256 MISMATCH!" -ForegroundColor Red
    Write-Host "Expected: $expectedHash"
    Write-Host "Actual:   $actualHash"
    Write-Host "DO NOT PROCEED. Re-download binary or contact Ionic." -ForegroundColor Red
    exit 1
}

```

5.4 Step 4: AV/EDR Scan (Jumper Server)

Duration: 5 minutes

```

# Windows Defender scan
Write-Host "Starting AV scan..." -ForegroundColor Cyan
Start-MpScan -ScanPath "C:\Staging\eKVM\CHG0012345" -ScanType CustomScan

# Check for threats
$threats = Get-MpThreat
if ($threats) {
    Write-Host "[FAIL] AV detected threats:" -ForegroundColor Red
    $threats | Format-Table -AutoSize
    Write-Host "DO NOT PROCEED. Quarantine file and contact Ionic." -ForegroundColor Red
    exit 1
} else {
    Write-Host "[PASS] No threats detected" -ForegroundColor Green
}

# Save scan results
Get-MpComputerStatus | Out-File -FilePath "C:\Evidence\CHG0012345\02_AV_Scan_Results.txt"

```

5.5 Step 5: Transfer Binary to eKVM

Duration: 10 minutes

Option A: WinRM Copy (Preferred)

```

# Prompt for eKVM credentials
$ekvmCred = Get-Credential -Message "Enter eKVM local admin credentials (LVHN\firstname.lastname)"

# Establish PS Session
$ekvmSession = New-PSSession -ComputerName ekvm-device-01.lvhn.local -Credential $ekvmCred

if ($ekvmSession.State -ne "Opened") {
    Write-Host "[ERROR] Failed to establish PS Session to eKVM" -ForegroundColor Red
    exit 1
}

# Create target directory on eKVM

```

```

Invoke-Command -Session $ekvmSession -ScriptBlock {
    New-Item -ItemType Directory -Path "C:\Temp\Update" -Force
}

# Transfer file
Write-Host "Transferring binary to eKVM via WinRM..." -ForegroundColor Cyan
$transferStart = Get-Date
Copy-Item -Path "C:\Staging\eKVM\CHG0012345\eKVM-Firmware-v3.2.1.exe" `
    -Destination "C:\Temp\Update\" `
    -ToSession $ekvmSession `
    -Verbose
$transferEnd = Get-Date
$transferDuration = ($transferEnd - $transferStart).TotalSeconds

Write-Host "[SUCCESS] Transfer completed in $transferDuration seconds" -ForegroundColor Green

# Keep session open for next steps
Option B: SMB Copy (Fallback)

# On eKVM (via separate RDP session): Create temporary share
# RDP to ekvm-device-01, open PowerShell:
New-Item -Path "C:\Temp\Update" -ItemType Directory -Force
New-SmbShare -Name "eKVMUpdate$" -Path "C:\Temp\Update" `
    -FullAccess "LVHN\firstname.lastname.admin" `
    -EncryptData $true

# Back on jumper server:
Copy-Item -Path "C:\Staging\eKVM\CHG0012345\eKVM-Firmware-v3.2.1.exe" `
    -Destination "\\ekvm-device-01\eKVMUpdate$" `
    -Credential $ekvmCred `
    -Verbose

# On eKVM: Remove share immediately
Remove-SmbShare -Name "eKVMUpdate$" -Force

```

5.6 Step 6: Verify Binary Integrity (eKVM)

Duration: 2 minutes

```

# Via WinRM session (from step 5.5)
$ekvmHash = Invoke-Command -Session $ekvmSession -ScriptBlock {
    $hash = (Get-FileHash -Path "C:\Temp\Update\eKVM-Firmware-v3.2.1.exe" -Algorithm SHA256)
    return $hash
}

if ($ekvmHash -eq $expectedHash) {

```

```

        Write-Host "[PASS] SHA-256 verification on eKVM PASSED" -ForegroundColor Green
        $ekvmHash | Out-File -FilePath "C:\Evidence\CHG0012345\03_Hashes_Post_Transfer.txt" -Append
    } else {
        Write-Host "[FAIL] SHA-256 MISMATCH on eKVM!" -ForegroundColor Red
        Write-Host "Expected: $expectedHash"
        Write-Host "Actual:   $ekvmHash"
        Write-Host "DO NOT PROCEED. File corrupted during transfer." -ForegroundColor Red
        Remove-PSSession $ekvmSession
        exit 1
    }
}

```

5.7 Step 7: Run Installer on eKVM

Duration: 15 minutes

```

# Execute installer via WinRM
Write-Host "Starting installer on eKVM..." -ForegroundColor Cyan
$installJob = Invoke-Command -Session $ekvmSession -ScriptBlock {
    $installer = "C:\Temp\Update\eKVM-Firmware-v3.2.1.exe"
    $logFile = "C:\Temp\Update\install.log"

    # Run installer silently
    $process = Start-Process -FilePath $installer `
        -ArgumentList "/silent", "/log:$logFile" `
        -Wait -PassThru -NoNewWindow

    return @{
        ExitCode = $process.ExitCode
        LogPath = $logFile
    }
} -AsJob

# Wait for completion (timeout 15 minutes)
$installJob | Wait-Job -Timeout 900

# Get results
$installResult = Receive-Job -Job $installJob

if ($installResult.ExitCode -eq 0) {
    Write-Host "[SUCCESS] Installer completed with exit code 0" -ForegroundColor Green
} else {
    Write-Host "[FAIL] Installer failed with exit code $($installResult.ExitCode)" -ForegroundColor Red
    Write-Host "Check log file: $($installResult.LogPath)" -ForegroundColor Yellow
    # DO NOT exit yet; collect logs first
}

```

```

# Retrieve install log
Copy-Item -Path "C:\Temp\Update\install.log" `
    -Destination "C:\Evidence\CHG0012345\04_Installation_Log.txt" `
    -FromSession $ekvmSession

```

5.8 Step 8: Verify Services and Functionality

Duration: 10 minutes

```

# Wait 30 seconds for services to stabilize
Write-Host "Waiting for services to start..." -ForegroundColor Cyan
Start-Sleep -Seconds 30

# Check eKVM services
$services = Invoke-Command -Session $ekvmSession -ScriptBlock {
    Get-Service -Name "eKVM*" | Select-Object Name, Status, StartType
}

Write-Host "eKVM Services Status:" -ForegroundColor Cyan
$services | Format-Table -AutoSize

$stoppedServices = $services | Where-Object {$_.Status -ne "Running" -and $_.StartType -ne "Manual"}
if ($stoppedServices) {
    Write-Host "[WARN] Some services not running:" -ForegroundColor Yellow
    $stoppedServices | Format-Table -AutoSize
    # Attempt to start
    foreach ($svc in $stoppedServices) {
        Write-Host "Attempting to start $($svc.Name)..." -ForegroundColor Yellow
        Invoke-Command -Session $ekvmSession -ScriptBlock {
            Start-Service -Name $using:svc.Name -ErrorAction SilentlyContinue
        }
    }
    Start-Sleep -Seconds 10
    # Re-check
    $services = Invoke-Command -Session $ekvmSession -ScriptBlock {
        Get-Service -Name "eKVM*" | Select-Object Name, Status
    }
    $services | Format-Table -AutoSize
}

$services | Out-File -FilePath "C:\Evidence\CHG0012345\05_Service_Verification.txt"

# Basic connectivity test (ping eKVM from jumper)
Test-Connection -ComputerName ekvm-device-01.lvhnl.local -Count 2

# Close PS Session

```

```
Remove-PSSession $ekvmSession
```

5.9 Step 9: Clinical Workflow Sanity Check

Duration: 10 minutes

Note: This step requires clinical engineering or operator familiar with nCommand Lite.

1. **RDP to eKVM** (as local user, not admin account)

```
mstsc /v:ekvm-device-01.lvhnl.local
```

- Username: .\ekvm-operator (standard non-admin account)

2. **Launch nCommand Lite** from desktop or Start Menu

3. **Perform P2P/WebRTC Test Call**

- Initiate test connection to known endpoint
- Verify video/audio stream establishes
- Confirm no latency or quality degradation
- Duration: 2-minute test call minimum

4. **Document Results**

```
# On jumper server
$testResult = @"
Sanity Check - CHG0012345
Date/Time: $(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')
Operator: $env:USERNAME
Test Type: nCommand Lite P2P/WebRTC
Result: [PASS/FAIL]
Notes: Video and audio streams established normally. No observed issues.
"@

$testResult | Out-File -FilePath "C:\Evidence\CHG0012345\06_Connectivity_Test.txt"
```

5. **Logout from eKVM**
-

6. Post-Window Activities (H+1 to H+4)

6.1 Step 10: Cleanup (H+1)

Duration: 15 minutes

1. **Remove Temporary Files on eKVM**

```
# Via new PS Session (or RDP)
$ekvmSession = New-PSSession -ComputerName ekvm-device-01 -Credential $ekvmCred
Invoke-Command -Session $ekvmSession -ScriptBlock {
```

```

        Remove-Item -Path "C:\Temp\Update\*" -Force -Recurse
    }
    Remove-PSSession $ekvmSession
2. Remove JIT Account Privileges

    # LVHN IT: Remove from local admins
    Remove-LocalGroupMember -Group "Administrators" -Member "LVHN\firstname.lastname.admin"

    # Disable AD account
    Disable-ADAccount -Identity "firstname.lastname.admin"

    # Remove eKVM local admin
    # (via RDP or Group Policy)
    net user ekvm-maint-CHG0012345 /delete
3. Close Network Rules

    # LVHN IT: Revert firewall ACLs
    configure
    delete rulebase security rules "Maint-Jumper-to-eKVM-CHG0012345"
    commit
4. Validate Rules Reverted

    Test-NetConnection -ComputerName ekvm-device-01 -Port 3389
    # Expected: TcpTestSucceeded = False (or timeout)
5. Revert GPOs (if drive mapping was enabled)

    # LVHN IT: Revert GPO changes
    gpupdate /force

```

6.2 Step 11: Compile Evidence Package (H+2)

Duration: 20 minutes

```

# On jumper server
cd C:\Evidence\CHG0012345

# Create ZIP archive
$zipPath = "C:\Evidence\LVHN-eKVM-CHG0012345-Evidence.zip"
Compress-Archive -Path "C:\Evidence\CHG0012345\*" -DestinationPath $zipPath -CompressionLevel Optimal

# Verify ZIP contents
Write-Host "Evidence Package Contents:" -ForegroundColor Cyan
Get-ChildItem -Path "C:\Evidence\CHG0012345\" | Select-Object Name, Length, LastWriteTime | Format-Table

# Create metadata file
$metadata = @"

```

```

Evidence Package Metadata
=====
Change Number: CHG0012345
Operator: $env:USERNAME
Start Time: [From transcript]
End Time: $(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')
Target Device: ekvm-device-01.lvhnl.local
Update Applied: eKVM Firmware v3.2.1
Result: SUCCESS
SHA-256 (Pre-Transfer): $expectedHash
SHA-256 (Post-Transfer): $ekvmHash
Installer Exit Code: 0
Evidence Package Path: $zipPath
"@

$metadata | Out-File -FilePath "C:\Evidence\CHG0012345\00_Metadata.txt"

# Stop transcript
Stop-Transcript

Write-Host "[SUCCESS] Evidence package ready: $zipPath" -ForegroundColor Green

```

6.3 Step 12: Update Change Ticket (H+4)

Duration: 15 minutes

1. **Log in to Change Management System** (ServiceNow, etc.)
2. **Update Change Ticket CHG0012345:**
 - **Status:** “Implementing → Completed”
 - **Work Notes:**

Update completed successfully at [timestamp].

Target Device: ekvm-device-01.lvhnl.local
Update Version: eKVM Firmware v3.2.1
Operator: Firstname Lastname

Verification:

 - SHA-256 hash verified (pre/post transfer)
 - AV scan passed (no threats)
 - Installer exit code: 0
 - Services running: eKVM* (all services nominal)
 - nCommand Lite sanity check: PASS (P2P/WebRTC functional)

Evidence package attached.

Post-window cleanup completed (JIT accounts removed, ACLs reverted).

3. Attach Evidence Package:

- Upload LVHN-eKVM-CHG0012345-Evidence.zip (attachment limit may require SFTP/shared drive)
- Include link to shared drive if uploaded externally

4. Request Closure:

- Assign to Change Manager for review
 - Status: “Awaiting Approval” → “Closed Successful”
-

7. Rollback Procedure

7.1 Rollback Triggers

Execute rollback if ANY of the following occur: - Installer exit code 0 - Critical services fail to start after update - nCommand Lite P2P/WebRTC connectivity fails - Significant performance degradation observed - Unintended reboot or system instability

7.2 Rollback Steps

Duration: 30 minutes

1. Stop Current Operation

- Do NOT proceed with additional updates
- Document observed issue in detail

2. Restore Previous Version (if available)

```
# Via WinRM session to eKVM
$ekvmSession = New-PSSession -ComputerName ekvm-device-01 -Credential $ekvmCred

# Check if Windows System Restore points exist
Invoke-Command -Session $ekvmSession -ScriptBlock {
    Get-ComputerRestorePoint | Sort-Object CreationTime -Descending | Select-Object -First 1
}

# Restore to pre-update point (if available)
Invoke-Command -Session $ekvmSession -ScriptBlock {
    $restorePoint = (Get-ComputerRestorePoint | Sort-Object CreationTime -Descending | Select-Object -First 1)
    Restore-Computer -RestorePoint $restorePoint -Confirm:$false
}
```

3. Reinstall Previous Firmware Version (if no restore point)

- Obtain previous version installer from Ionic

- Follow steps 5.2-5.7 using previous version binary
- Verify downgrade successful

4. Verify Services and Connectivity

- Confirm eKVM services running
- Perform nCommand Lite sanity check
- Test network connectivity

5. Document Rollback

```
$rollbackReport = @"
ROLLBACK REPORT - CHG0012345
=====
Date/Time: $(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')
Operator: $env:USERNAME
Reason: [Describe issue that triggered rollback]
Action Taken: [System Restore / Reinstall v3.1.0]
Result: [SUCCESS / PARTIAL / FAILED]
Current eKVM Version: [Check installed version]
Services Status: [Output of Get-Service eKVM*]
Next Steps: [Escalate to Ionic / Schedule reattempt]
"@
```

```
$rollbackReport | Out-File -FilePath "C:\Evidence\CHG0012345\ROLLBACK_REPORT.txt"
```

6. Escalate to Ionic Technical Support

- Email: support@ionic.com
- Subject: "URGENT: eKVM Update Rollback - CHG0012345"
- Attach: Rollback report, installer log, service status

8. Troubleshooting

8.1 Common Issues and Resolutions

| Issue | Symptom | Resolution |
|-------------------------------|--|--|
| RDP Connection Refused | Test-NetConnection fails on port 3389 | 1. Verify network ACLs opened 2. Check Windows Firewall on eKVM 3. Confirm eKVM powered on |

| Issue | Symptom | Resolution |
|-------------------------------------|--|--|
| WinRM Access Denied | New-PSSession fails with “Access Denied” | 1. Verify JIT account has local admin rights on eKVM2. Check TrustedHosts: <code>Set-Item WSMAN:\localhost\Client\TrustedHosts -Value "ekvm-device-01" -Force</code> 3. Try HTTPS (port 5986) instead of HTTP (5985) |
| Hash Mismatch | SHA-256 does not match expected | 1. Re-download binary from Ionic CDN2. Verify expected hash from Ionic release notes3. Check for network proxy interference4. Contact Ionic if persistent |
| Installer Hangs | Installer runs >15 minutes without completing | 1. Check eKVM CPU/memory usage (Task Manager via RDP)2. Review installer log: <code>C:\Temp\Update\install.log</code> 3. Kill installer process: <code>Stop-Process -Name "eKVM-Firmware"</code> 4. Execute rollback procedure |
| Services Not Starting | <code>Get-Service eKVM*</code> shows “Stopped” | 1. Check Event Viewer → Application log for errors2. Manually start services: <code>Start-Service -Name <service-name></code> 3. Verify dependencies: <code>Get-Service <service-name> Select-Object -ExpandProperty DependentServices</code> 4. If critical service fails, execute rollback |
| nCommand Lite Not Connecting | P2P/WebRTC test call fails | 1. Verify eKVM network connectivity: <code>Test-Connection <peer-device></code> 2. Check Windows Firewall rules (nCommand Lite ports)3. Restart nCommand Lite service4. If persistent, execute rollback |

8.2 Escalation Path

| Severity | Contact | Response Time | Trigger |
|-----------------|------------------------------------|------------------|--|
| Low | LVHN IT Service Desk | 4 business hours | Non-critical issues during post-window |
| Medium | Ionic Support (email) | 2 hours | Installer failure, service issues (outside window) |
| High | Ionic Technical Lead (phone) | 30 minutes | Rollback required, clinical impact |
| Critical | Emergency Bridge + LVHN IT Manager | 15 minutes | Patient safety impact, system outage |

9. Evidence Collection

9.1 Required Evidence Files

All evidence must be collected in `C:\Evidence\CHG{ChangeNumber}\`:

1. **00_Metadata.txt** – Change details, operator, timestamps
2. **01_Legal_Banner.png** – Screenshot of RDP legal notice
3. **01_Hashes_Pre_Transfer.txt** – SHA-256 on jumper server
4. **02_AV_Scan_Results.txt** – Windows Defender scan output
5. **02_Transfer_Logs.txt** – PowerShell transcript of file transfer
6. **03_Hashes_Post_Transfer.txt** – SHA-256 on eKVM
7. **04_Installation_Log.txt** – Installer output from eKVM
8. **05_Service_Verification.txt** – `Get-Service` output post-install
9. **06_Connectivity_Test.txt** – nCommand Lite sanity check results
10. **07_Windows_Event_Logs/** – Filtered .evtx exports
 - Jumper_Security_4624.evtx (logon events)
 - Jumper_PowerShell_4104.evtx (script block logging)
 - eKVM_Security_4624.evtx
 - eKVM_Application_Installer.evtx
11. **08_Screenshots/** – Any additional visual evidence
12. **transcript-[timestamp].txt** – Full PowerShell transcript

9.2 Event Log Export Commands

```
# On jumper server
$changeNum = "CHG0012345"
```

```

$evidenceDir = "C:\Evidence\$changeNum\07_Windows_Event_Logs"
New-Item -ItemType Directory -Path $evidenceDir -Force

# Export Security log (Event ID 4624 - Successful logons)
wevtutil epl Security "$evidenceDir\Jumper_Security_4624.evtx" "/q:*[System[(EventID=4624)]]"

# Export PowerShell Operational log (Event ID 4104 - Script Block Logging)
wevtutil epl "Microsoft-Windows-PowerShell/Operational" "$evidenceDir\Jumper_PowerShell_4104.evtx"

# On eKVM (via WinRM or RDP)
# Similar commands, save to C:\Temp\Logs\ then copy back to jumper

```

9.3 Evidence Retention

- **Retention Period:** 6 years (minimum per healthcare regulations)
 - **Storage Location:** LVHN document management system + Ionic project archive
 - **Access Control:** Restricted to authorized personnel (LVHN IT, Compliance, Ionic)
-

10. Appendix A: Command Reference

PowerShell Snippets

Test Network Connectivity:

```
Test-NetConnection -ComputerName <hostname> -Port <port> [-InformationLevel Detailed]
```

Compute SHA-256 Hash:

```
(Get-FileHash -Path <file> -Algorithm SHA256).Hash
```

Create PS Session (WinRM):

```
$session = New-PSSession -ComputerName <hostname> -Credential (Get-Credential)
```

Copy File via WinRM:

```
Copy-Item -Path <source> -Destination <dest> -ToSession $session -Verbose
```

Run Command on Remote System:

```
Invoke-Command -Session $session -ScriptBlock { <commands> }
```

Export Event Log:

```
wevtutil epl <LogName> <OutputFile> "/q:*[System[(EventID=<ID>)]]"
```

11. Appendix B: Event Log Reference

| Event ID | Log Source | Description | Significance |
|----------|---|-----------------------------|----------------------------------|
| 4624 | Security | Successful logon | Tracks operator RDP sessions |
| 4625 | Security | Failed logon | Detects brute-force attempts |
| 4634 | Security | Logon session terminated | Tracks session end |
| 4672 | Security | Special privileges assigned | Tracks admin privilege use |
| 4688 | Security | New process created | Tracks installer execution |
| 4104 | PowerShell/Operational Script block logging | Script block logging | Captures all PowerShell commands |
| 21 | TerminalServices-RemoteConnectionManager | RDP logon successful | RDP session tracking |
| 24 | TerminalServices-LocalSessionManager | RDP session disconnected | Session end tracking |
| 1033 | Application | Windows Installer completed | Installer success |
| 1034 | Application | Windows Installer removed | Uninstall event |

12. Appendix C: Contact Information

| Role | Name | Email | Phone | Availability |
|-----------------------------|--------|-----------------------|-----------------|--------------------------|
| Ionic Technical Lead | [Name] | techsupport@ionic.com | 1-XXX-XXX-XXXX | 24/7 |
| LVHN IT Operations | [Name] | it-ops@lvhn.org | +1-XXX-XXX-XXXX | Business hours + on-call |
| LVHN Change Manager | [Name] | change@lvhn.org | +1-XXX-XXX-XXXX | Business hours |
| LVHN Network Team | [Name] | network@lvhn.org | +1-XXX-XXX-XXXX | 24/7 |

| Role | Name | Email | Phone | Availability |
|-----------------------------|--------|------------------|-----------------|--------------------------|
| LVHN Security | [Name] | infosec@lvhn.org | #1-XXX-XXX-XXXX | Business hours + on-call |
| Clinical Engineering | [Name] | clineng@lvhn.org | #1-XXX-XXX-XXXX | Business hours |

Emergency Bridge Line: [Conference number] (activated during maintenance windows)

Document Approval:

| Name | Role | Date | Signature |
|--------|----------------------------|-------|-----------|
| [Name] | Ionic Technical Lead | _____ | _____ |
| [Name] | LVHN IT Operations Manager | _____ | _____ |
| [Name] | LVHN Information Security | _____ | _____ |

Version History:

| Version | Date | Author | Changes |
|---------|------------|--------------|---|
| 0.1 | 2025-10-15 | Ionic | Initial draft |
| 0.9 | 2025-11-01 | Ionic + LVHN | Incorporated LVHN feedback; added troubleshooting |
| 1.0 | 2025-11-11 | Ionic | Final approved version |

End of Method of Procedure