Deployment Prerequisite

System Requirements

- Windows 11 (workstations) or Windows Server (server fleet).
- Local Administrator rights for installation. The agent runs as a system service afterward.
- Outbound internet access (HTTPS).

Antivirus / EDR Exclusions

The agent installs, updates, and removes software as part of normal operation, which can trigger real-time, heuristic, and behavior-based detection alike. Please exclude the following from ALL scanning modes — standard file scanning, heuristic/behavior-based detection, and any ransomware or "protected folder" module — not just the default file scanner:

Item	Detail
Folder (recursive)	C:\ProgramData\Actonix\
Trusted publisher	ACTONIX SOFTWARE PRIVATE LIMITED — all Actonix executables are Authenticode-signed under this publisher. Where your product supports publisher/certificate-based trust, prefer that over a path-only exclusion.

Note: a folder exclusion alone often only covers the file scanner. Please confirm your product's behavior-detection / EDR module is excluded separately — this is the most common reason agents still get blocked after a folder exclusion is added.

PowerShell Process & Behavioural Detection

If the folder and publisher exclusions above are already in place but a script still gets blocked, the exclusion is most likely only covering the file scanner — the block is coming from PowerShell behavioral/AMSI detection or a process-chain rule instead. This is common and expected on EDR products; it needs its own, separate allow rule.

1. Process exclusion

For most operations, Actonix executables run PowerShell logic in-process — no separate powershell.exe is started, so this doesn't apply. The one case where it does:

- powershell.exe — started as a child process only for a local scheduled task re-running a signed Actonix script.
- pwsh.exe (PowerShell 7 / Core) — not applicable. Actonix uses Windows PowerShell 5.1 only.
- cmd.exe — not applicable. The script is launched directly by the scheduled task action, with no intermediate command shell.

2. Process chain

The confirmed chain for this scenario is:

```
Windows Task Scheduler → powershell.exe → <signed Actonix script>.ps1
```

There is no persistent Actonix agent process that launches these — each script is scheduled and run independently, as SYSTEM, directly by Windows Task Scheduler. So the "trusted parent" in your EDR's rule (if it requires one) is Task Scheduler itself, not a separate Actonix binary.

3. Script / content detection

If the block message mentions AMSI detection, script blocked, PowerShell behavioral detection, malicious script, memory execution, or suspicious PowerShell — a filesystem exclusion will not resolve it. This needs a script-content or process-behavior allow rule specifically.

4. Confirm the exact launch pattern

On the endpoint, the script is always launched as:

```
powershell.exe -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden -File "<path under
C:\ProgramData\Actonix\>"
```

To find the actual block event on a Windows Defender-protected endpoint:

```
Get-WinEvent -LogName "Microsoft-Windows-Windows Defender/Operational" -MaxEvents 100 |
    Where-Object { $_.Message -match "Actonix|PowerShell|blocked|threat" } |
    Select-Object TimeCreated, Id, LevelDisplayName, Message |
    Format-List
```

Recommended rule (instead of a global exclusion)

Please don't disable AMSI/script scanning or exclude PowerShell globally. Instead, request a process-aware allow rule scoped to:

```
Trusted process:        powershell.exe
Launched by:            Windows Task Scheduler (no separate Actonix parent process)
Allowed script location: C:\ProgramData\Actonix\*.ps1
Command-line pattern:   -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden -File
"C:\ProgramData\Actonix\..."
Publisher:              ACTONIX SOFTWARE PRIVATE LIMITED
```

If you can share the exact block message or event from your AV/EDR, we can point you to the specific rule that's missing — path, process, process-chain, publisher, or AMSI/content-based.

Firewall / Network

Destination	Port / Protocol	Purpose
app.actonix.in, agent.actonix.in	TCP 443 (HTTPS), outbound	Console and agent communication
Actonix cloud service endpoints	TCP 443 (HTTPS), outbound	Software catalog and update downloads
Other Actonix-managed endpoints on the same local network	TCP/UDP 7680, LAN-only	Optional — Windows' own built-in peer caching feature (Delivery Optimization), used for OS/Office updates
Other Actonix-managed endpoints on the same local network/subnet	UDP 4580 + TCP 4580, LAN-only (default; configurable)	Optional — Actonix's own local peer-to-peer caching, used for software package installs. UDP is multicast-based discovery (administratively-scoped multicast, 239.192.0.0/16 range); TCP is the peer-to-peer file transfer. If blocked, the agent still installs software — it just downloads directly instead of from a LAN peer.

The two LAN peering rows above are separate mechanisms (Windows' native feature vs. Actonix's own) and both are optional performance optimizations, not required for the agent to function.