Technical Note

Admin By Request
ZERO TRUST PLATFORM

Product Platform: **Windows**
Product Version: **All versions**
Document Date: **27 March 2026**
Document Version: **1.0**
Classification: **Public**

Pre-Approval Does Not Override System Subprocess Blocking

Introduction

This document describes the potential conflict that can arise when pre-approval actions **do not** override system-imposed subprocess blocks within the platform. This occurs in Admin By Request by design - the documentation here aims to clarify the intended behavior, provide technical details, and specify how the system will enforce subprocess blocks even in the presence of pre-approval.

Background

In certain situations, Admin By Request pre-approval does not override portal or sub-setting policies that block system-file or System32 subprocesses. In practice, this means a parent application can be pre-approved and still fail if it launches a child process that is blocked by policy.

The over-arching business priority is to:

a. enforce least privilege and application control on Windows endpoints, ensuring that only approved applications can run with elevated privileges, and
b. ensure that system integrity is maintained by blocking potentially risky system subprocesses (e.g. those in System32).

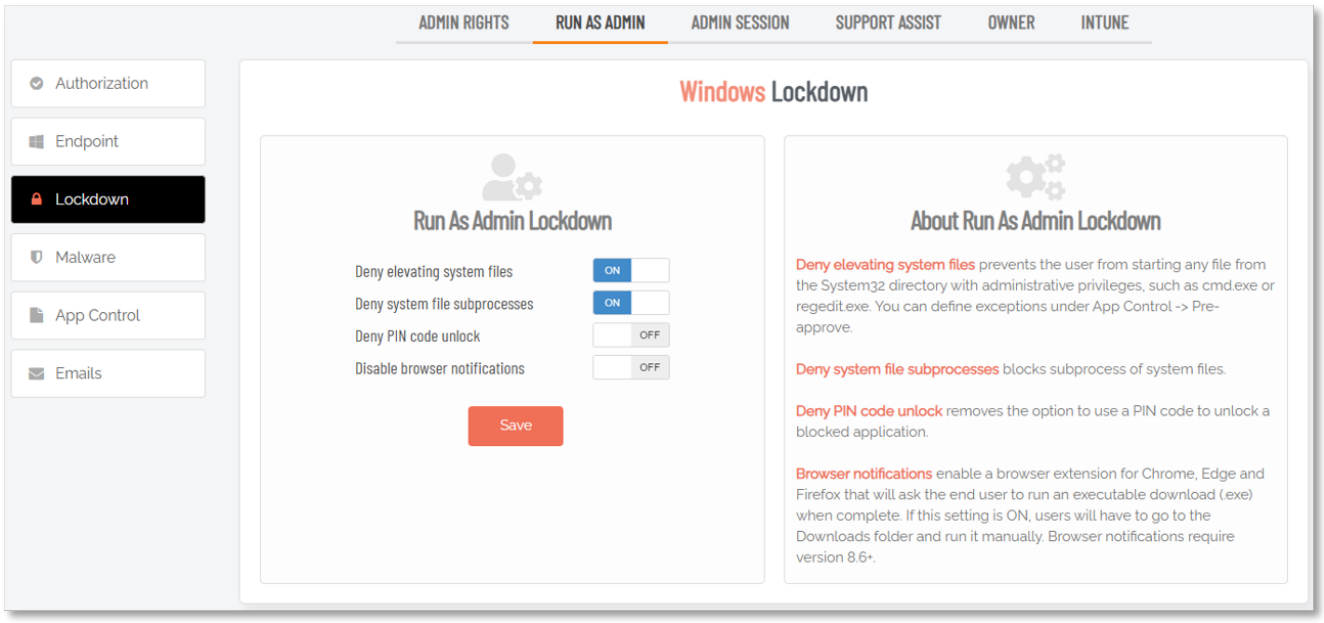
Admin By Request always attempts to achieve the correct balance between **user productivity** (allowing necessary workflows) and **security** (preventing unapproved or dangerous actions), but sometimes, those priorities do conflict.

This technical note is available online:

Pre-Approval Does Not Override System Subprocess Blocking

Observed behavior

The relevant settings in the Admin By Request portal are shown below. The two settings that are **ON** in the image are available for both RUN AS ADMIN and ADMIN SESSION (i.e. they can be configured independently):



1. **Deny elevating system files** - Prevents the user from starting any file from the System32 directory with administrative privileges, such as **cmd.exe** or **regedit.exe**.
2. **Deny system file subprocesses** - blocks subprocesses of system files from executing, *including those that are pre-approved*.

The following points apply to pre-approved applications that fire additional subprocesses during their execution:

- Pre-approval can allow the parent application to start.
- However, pre-approval does not supersede:
  - Deny system file subprocesses
  - global Sys32/System32 subprocess blocking
- This remains true even when:
  - the pre-approval is configured in a sub-setting
  - the pre-approval is intended to allow subprocesses
  - the parent executable itself is clearly pre-approved
- The effective result is that the child process is still denied, and the user sees partial execution, access denied popups, or complete failure of the workflow.

What this means for IT Admins

If an elevated workflow depends on spawning a blocked system child process, pre-approving the parent application is not enough.

To make the workflow succeed, the blocking policy itself must be changed in the relevant scope. The suggested workaround is to create a sub-setting for the affected users or devices that relaxes the subprocess block, or to temporarily disable the blocking policy where appropriate.

Examples

Example 1: Disk Cleanup

Scenario

Disk Cleanup was made available through pre-approval / tray-tool style configuration, but it still produced access denied popups and did not elevate cleanly.

Behavior

- Disk Cleanup launches **DismHost.exe**.
- **DismHost.exe** is treated as a system file subprocess.
- When **Endpoint Privilege Management > Settings > Windows Settings > Lockdown > RUN AS ADMIN | ADMIN SESSION > Deny system file subprocesses** is enabled, the child process remains blocked.
- Adding additional pre-approvals does not fully resolve the problem.

IT Admin Guidance

If Disk Cleanup must run fully elevated, the blocking policy for system file subprocesses must be relaxed in the applicable scope. Pre-approving Disk Cleanup alone is not sufficient.

Example 2: WSL Install / Update

Scenario

wsl.exe --install and **wsl.exe --update** were pre-approved, but the workflow still fails because **conhost.exe** is blocked as a child process.

Behavior

- The parent process **wsl.exe** was pre-approved.
- The child process **conhost.exe** is still blocked.
- The global Sys32/System32 subprocess block takes precedence over the pre-approvals.
- The same limitation applies when the pre-approval is created in a sub-setting.

IT Admin Guidance

If WSL or another approved command launches a blocked System32 child process, adjust the blocking policy in the relevant global or sub-setting scope. Do not assume that pre-approval of the parent executable will overrule the subprocess block.

General Rule

When documenting or configuring Admin By Request pre-approvals:

- Pre-approval is not a higher-priority override than system subprocess blocking policies.
- If the child process is blocked by policy, the approved parent process can still fail.
- When that happens, solve it with policy scope and settings design, not just with application pre-approval.