

Forum:[Hack Forums](#)

Thread: [UAC Bypass Silent Elevation on Fully Updated Win 7/11 \(Functional up to Build 26200\)](#)

Wake - September 20, 2026, 1:04 pm

? loading encrypted image...

UAC Bypass Silent Elevation on Fully Updated Win 7/11 (Functional up to Build 26200)

? loading encrypted image...

Written by [Wake](#)

? loading encrypted image...

About this thread

In this thread I will be going over a Silent UAC Bypass from Windows 7-11 that elevates our privileges from medium integrity to high. I will explain what UAC is, different integrity levels what they mean and some prerequisites that must be met.

For more in-depth guidance feel free to convo me.

Disclaimer

This thread is for informational and educational purposes only. I will not be responsible for any action performed by any reader.

Introduction

User Account Control (UAC) is a security feature in Windows designed to prevent unauthorized administrative changes without the user knowing. Even if you are logged in as an administrator everyday tasks run in a restricted "medium integrity" state. The goal of a silent UAC bypass is to transition into "high integrity" (full admin) by taking advantage of trusted Windows binaries that skip the standard prompt.

Understanding Integrity Levels

To understand how this bypass works we need to look at Windows integrity levels. Windows assigns every running process an integrity level to restrict what it can do even if the user is an admin.

Low Integrity: Sandboxed applications like web browser tabs. These have very restricted access and cannot modify normal user files.

Medium Integrity: Everyday user tasks and standard administrator sessions. This is where your desktop runs. You have access to your personal files but you cannot make changes to system files or core settings.

High Integrity: Elevated administrative tasks ("Run as Administrator"). This gives full control over the system allowing you to edit system files and install system drivers.

System Integrity: Reserved for the Windows operating system core and background services.

When an administrator logs in Windows splits your access into two tokens. You're day to day apps get a medium integrity token. The goal of this silent UAC bypass is to jump from our medium integrity session directly into high integrity without triggering the Windows UAC prompt.

? loading encrypted image...

What will I need for this?

Target Account Rights: Your account must be in the local Administrators group (running in medium integrity). This is the standard integrity level for local administrators. This UAC bypass takes a local administrator at medium integrity and runs a program of choice at high integrity with zero prompt.

UAC Level: UAC must be left at Notify me only when apps try to make changes ("the Windows default setting").

Supported operating systems: Windows 7 through 11 (up to Build 26200).

PowerShell version: PowerShell 5.1 or newer.

UAC-Bypass files: For this guide we will be using the 3 files from this [GitHub repo](#).

NtObjectManager: We need NtObjectManager installed on the target machine we can do this by downloading it [here](#) and staging it with the other 3 files we downloaded before or we can run one PowerShell command on medium integrity.

Code Install-Module "NtObjectManager" -Scope CurrentUser

Normally installing modules requires elevation to write to Program Files. The -Scope CurrentUser flag installs it directly to your personal profile folder instead completing entirely in medium integrity without triggering a prompt.

PowerShell Gallery is marked as "untrusted" by default in Windows, it will ask if you are sure you want to install software from PSGallery accept all install prompts.

? loading encrypted image...

How It Works (Step-by-Step Guide)

Now that we have our files ready and understand integrity levels lets look at how we actually execute this bypass and what happens behind the scenes.

Step 1: Running the script

Open a standard PowerShell prompt (running at medium integrity). Navigate to the folder where you saved your bypass files and run the main script.

Code .\run.ps1

Step 2: Abusing the AppInfo service via RPC

Windows handles UAC elevation requests through a background service called Application Information (AppInfo). Under the default Windows UAC settings certain built in system programs (like Task Manager) have auto elevation enabled. This script makes an RPC call directly to the AppInfo service asking it to launch one of these trusted Windows binaries. Because the binary is digitally signed by Microsoft and flagged to auto elevate Windows launches it into high integrity immediately without popping a UAC prompt on the screen.

Step 3: Parent Process Spoofing

Now that our trusted Windows binary is running silently at high integrity the script captures its process handle. From here we tell Windows to spawn our chosen program (like cmd.exe) and set the trusted high integrity binary as its parent. Because child processes in Windows inherit the security token and integrity level of their parent our new command prompt launches directly into high integrity with full admin rights.

How to verify you are High Integrity

Because this happens silently with no prompt we need to verify we actually reached high integrity. In your new command prompt run this command.

Code net sessionIf it says System error 5 has occurred. Access is denied. The session is still restricted to medium integrity.

If it says There are no entries in the list. The bypass was successful and you are running with full administrative privileges.

Why this won't work on newer Windows updates

In Build 26200 Microsoft patched the AppInfo service to block this exact behavior. Now when a medium integrity process tries to grab or duplicate handles from the elevated process AppInfo immediately denies the request.

? loading encrypted image...

Finished Result

If everything was done correctly, you have now elevated from medium integrity to high and bypassed the UAC prompt, cmd.exe is a PoC how you chose to use this is up to you.

If you run into any issues, feel free to reply and I'll help out.

[Thank you for taking the time to read this thread. If you enjoyed this please check out the rest of my tutorials and write-ups](#)