Turn off Smart App Control on Windows 11

Normal desktop, then Disable-SmartAppControl2.ps1, then one reboot, then Test-SmartAppControlOff.ps1. The scripts are the procedure. Safe Mode is not used. Do not run the retired script in __OLD__.

This is a lab procedure for a Windows 11 virtual machine you control. It turns Smart App Control off by writing the code-integrity policy state, then rebooting so that policy loads. It does not remove binaries. Take a hypervisor snapshot named pre-sac-off before the first change. That snapshot is the clean undo.

What off means

The final gate is Test-SmartAppControlOff.ps1 after the reboot. Exit code 0 and the sentence Smart App Control is off are the same result. The pre-reboot sentence Smart App Control is set to Off is not that gate.

Check	Pass
Policy state	VerifiedAndReputablePolicyState = 0, and OK Policy state is 0 (Off).
CiTool desktop policy	VerifiedAndReputableDesktop is not enforced, or is not present.
CiTool evaluation policy	VerifiedAndReputableDesktopEvaluation is not enforced, or is not present.
Script result	Smart App Control is off. Exit code 0.

MinValueSeen and SacLearningModeSwitch are printed and are not pass tests. A missing value there does not fail the check. An unreadable CiTool listing does fail it. A policy that is present with Is Currently Enforced = true fails it. Do not treat a policy name that the script could not read as absent.

The three values

0 is Off, 1 is Enforce, 2 is Evaluation. The script writes the policy state and then tries to write the same Off value into the protected minimum. It also tries to set the Defender learning switch to 0 so learning mode does not write the state back.

Value	Off	Role
VerifiedAndReputable PolicyState	0	The switch, under CI\Policy. This is the pass test.
VerifiedAndReputablePolicyState MinValueSeen	0	Stickiness, under CI\Protected. Same number as the switch.
SacLearningModeSwitch	0	Under Windows Defender. Tamper Protection can block this write.

Write the policy state and the minimum as the same number. Editing only one of them can leave code integrity blocking almost everything. The script writes a JSON backup beside itself before it changes either value. There is no Smart App Control Group Policy.

Safe Mode is not required

Run the scripts from an elevated PowerShell window on the normal desktop. Safe Mode does not take ownership of CI\Protected, and it does not load the Off policy. The reboot after a successful policy write is what loads it. CiTool -r is attempted and is not a substitute for that reboot.
A warning that CI\Protected is owned by TrustedInstaller means the first write was denied. The script enables the ownership privileges and retries. That warning is not a request to boot Safe Mode. If the retry also fails, reboot anyway and run the check. If the check fails, restore the snapshot.

What this does not turn off

Microsoft Defender Antivirus, the firewall, SmartScreen, and Tamper Protection stay as they were. The execution-policy script is a separate prep step. It does not change Smart App Control. Windows Security can still open.

Before you start

- Guest is Windows 11, local administrator, and it already boots normally.
- Snapshot name: pre-sac-off. If the guest starts blocking normal programs after the reboot, revert that snapshot.
- The title bar of PowerShell must say Administrator.
- Do not hand-edit one registry value. Run Disable-SmartAppControl2.ps1.

1. Make execution policy Bypass permanent

Do this once, from the folder that contains the scripts. It persists across reboots for LocalMachine. It does not turn Smart App Control off. A stock install is often Restricted, so the .ps1 files will not start until this launch bypasses the policy for this one process.

```
powershell -NoProfile -ExecutionPolicy Bypass -File .\Set-ExecutionPolicyBypass.ps1
```

That line is the whole of Make_Execution_Policy_Bypass_Permanent.txt. The script sets Windows PowerShell 5.1 LocalMachine, the 32-bit key, and PowerShell 7 when pwsh.exe is on PATH. Open a new elevated window and confirm Get-ExecutionPolicy -List shows LocalMachine as Bypass.

If the script warns that MachinePolicy or UserPolicy is set, Group Policy wins and this write is ignored. Fix that policy before the later steps. Process scope dies with the window. CurrentUser would cover only one account. LocalMachine is the permanent scope this step uses.

2. Run Disable-SmartAppControl2.ps1

Same elevated window, or a new one after step 1. The prompt stays on the normal desktop. Answer N at the reboot question so you can read the values, then reboot in the next step.

```
.\Disable-SmartAppControl2.ps1
```

Script output	Meaning
Backup written: SAC-backup-before-disable-....json	The three current values were saved next to the script.
Current values show VerifiedAndReputablePolicyState = 0	The switch read back as Off.
CI\Protected is owned by TrustedInstaller. Taking ownership and retrying...	The protected value needed an ownership change. The script retries.
Could not write VerifiedAndReputablePolicyStateMinValueSeen	Stickiness was not written. The policy Off write still stands. The check decides.
Could not write SacLearningModeSwitch because Tamper Protection is on	Turn Tamper Protection off, rerun this script, then turn Tamper Protection back on.
CiTool -r exited ... Reboot anyway.	The refresh did not replace the reboot.
Smart App Control is set to Off. A reboot is required.	Pre-reboot message. Not the final gate.
Reboot now? [y/N]	Answer N. Reboot yourself after you have read the current values.

If the policy write itself throws, the script stops and does not ask to reboot. Do not reboot on that failure. A warning about the protected value or the learning switch is different: the Off value was still set. Reboot and run the check. Rerunning the script writes Off again. It does not require Safe Mode.

3. Reboot once

From an elevated window, after you have read the current values:

```
shutdown /r /t 0
```

Code Integrity loads the policy at startup. Leave the guest on the normal desktop. Do not add safeboot. Answering Y at the script prompt is the same reboot, with no chance to read the values first.

4. Verify after the reboot

Elevated PowerShell, after the desktop is up:

```
.\Test-SmartAppControlOff.ps1
```

Exit code 0 and the sentence Smart App Control is off are the gate. The sentence Smart App Control is NOT off means stop. Policy state is not 0 is a failure. CiTool output could not be read is a failure, including when CiTool.exe is missing. Is Currently Enforced = true on either watched policy is a failure. A policy that is absent is a pass for that name.

The check never writes the policy. Run it again after a feature update or a Defender platform update. Enterprise management can push a Smart App Control state of its own. A failed check is not fixed by booting Safe Mode.

If the check fails

- Policy state is not 0: run Disable-SmartAppControl2.ps1 again on the normal desktop. Require the current-values line to show 0, reboot, and run the check again.

- CiTool still shows a watched policy enforced: the reboot did not load Off, or something wrote the state back. Revert pre-sac-off if a second run does not clear it.

- The guest blocks ordinary programs, including PowerShell: revert the snapshot. Do not keep editing the code-integrity keys by hand.

- Only the learning-switch warning appeared, and the check passed: Off is loaded. Turn Tamper Protection back on if you turned it off for the retry.

Files

File	When	Role
Set-ExecutionPolicy Bypass.ps1	Once, elevated, before the other files.	LocalMachine Bypass. Does not change Smart App Control.
Make_Execution_Policy_ Bypass_Permanent.txt	The one-line launch for that script.	Use it when the window is still Restricted.
Disable-SmartAppControl2.ps1	Normal desktop, elevated.	Writes Off, retries the protected value, asks for a reboot.
Test-SmartAppControlOff.ps1	After the reboot, elevated.	Final gate. Exit 0 only when Smart App Control is off.
__OLD__\ Disable-SmartAppControl.ps1	Do not run.	Retired. takeown does not take ownership of a registry key.

Order of operations

- Snapshot pre-sac-off.
- Elevated PowerShell. Run the command in Make_Execution_Policy_Bypass_Permanent.txt.
- Require LocalMachine Bypass. A Group Policy warning stops the procedure.
- .\Disable-SmartAppControl2.ps1 on the normal desktop. Answer N at the reboot prompt.
- Require VerifiedAndReputablePolicyState = 0 in the current values. Then shutdown /r /t 0.
- .\Test-SmartAppControlOff.ps1. Require Smart App Control is off. and exit code 0.

Manual undo is Restore_SmartAppControl_Windows11.pdf and Enable-SmartAppControl.ps1. The execution-policy Bypass stays in place until you change it on purpose.