PROCEDURE

Completely disable Microsoft Defender on Windows 11 Enterprise

The script that does the disable is Disable_Defender2.PS1, verified on a Windows 11 Enterprise trial inside VMware Workstation Pro. disable_Defender.PS1 is archived under __OLD__ and is not used. Smart App Control is left alone.

This is a lab procedure for a virtual machine you control. It disables the antivirus engine and its boot drivers. It does not remove the binaries, and it is not a supported Microsoft configuration. A later Defender platform update can restore the service start values. Take a VMware snapshot before the first change; that snapshot is the clean undo.

What "off" means after this procedure

Group Policy and the old DisableAntiSpyware registry value do not do this job. DisableAntiSpyware is a legacy setting. Microsoft documents it as ignored on devices onboarded to Microsoft Defender for Endpoint from platform 4.18.2108.4 (September 2021), and points antivirus vendors to Windows Security Center registration. Tamper Protection blocks changes while it is on, and WdFilter.sys registers a registry callback over Defender's service keys while that driver is loaded. Run Disable_Defender2.PS1 in Safe Mode, where WdFilter is not loaded. Start value 4 means Disabled.

On the machine this guide was verified on, the post-reboot checks were:

Check	Result on the verified guest
WinDefend, WdNisSvc, Sense	Stopped, StartType Disabled
MsMpEng.exe	Not running (Get-Process returned nothing)
Get-MpComputerStatus	AMServiceEnabled, AntivirusEnabled, RealTimeProtectionEnabled, IsTamperProtected all False
WdFilter, WdBoot config	START_TYPE 4 DISABLED
WdNisDrv state	STATE 1 STOPPED, Win32 exit 1077 (never started)
fltmc	WdFilter absent from the loaded minifilter list
Defender scheduled tasks	Each task under \Microsoft\Windows\Windows Defender\ reports State Disabled

Get-MpComputerStatus is often stale after a hard disable and can still report True, or throw. On this guest it agreed with the service list. Trust the service list, Task Manager, sc query, fltmc, and the task State lines if they ever disagree.

What this does not turn off

Windows Firewall and the Windows Security app shell stay. SecurityHealthService is not modified. The Security app will open and report that the threat service has stopped. That is the expected result, not a failed disable.

LEAVE SMART APP CONTROL ALONE

Smart App Control is not Defender Antivirus. This procedure does not turn it off, and Disable_Defender2.PS1 does not change it. Leave Smart App Control where it is.

Before you start

- Guest is Windows 11 Enterprise (trial is fine). Local administrator. VMware Workstation Pro, guest already booting normally.
- VM menu → Snapshot → Take Snapshot. Name it pre-defender-disable. If Safe Mode comes up black, or the guest will not leave Safe Mode, revert this snapshot. Do not rely on System Restore.
- Mouse integration is often dead in Safe Mode. Ctrl+Alt releases the cursor to the host. Click the console to type again.
- Safe Mode uses the basic display driver. A low-resolution desktop is normal. A black screen for more than a few minutes is not: revert the snapshot rather than hard-resetting, because a reset with safeboot still set just boots Safe Mode again.

1. Turn Tamper Protection off, still in the normal session

Do this before Safe Mode. Tamper Protection cannot be flipped from the command line while WdFilter is loaded.

- Sign in as a local admin.
- Windows Security → Virus & threat protection → Manage settings.
- Turn Tamper Protection off and accept the UAC prompt.
- On the same page, turn Real-time protection off. It may turn itself back on later. That does not matter; the service disable is what sticks.
- Leave Smart App Control alone. Leave the firewall alone. Neither one is Defender Antivirus.

2. Optional policy layer (Enterprise has gpedit)

This does not stop MsMpEng.exe by itself. It stops some of the "turn me back on" logic and makes Windows Security say the setting is managed. Skip it if you only want the service kill.

- Win+R → gpedit.msc.
- Computer Configuration → Administrative Templates → Windows Components → Microsoft Defender Antivirus.
- Turn off Microsoft Defender Antivirus → Enabled → OK.
- Under Real-time Protection, set Turn off real-time protection to Enabled.
- Do not reboot yet. A reboot now just lets Defender start again and argue with the policy.

3. Force the next boot into Safe Mode

Open an elevated Command Prompt (Win+X → Terminal (Admin), or cmd → Run as administrator). Do this in Command Prompt, not PowerShell. In PowerShell, {current} is a script block.

```
bcdedit /set {current} safeboot minimal
bcdedit /enum {current}
shutdown /r /t 0
```

bcdedit /enum {current} must show safeboot Minimal before you reboot. If it does not, do not reboot. Alternate path: Win+R → msconfig → Boot tab → Safe boot → Minimal → OK → Restart. Same result.

> **VMWARE**
> If the console stays black for several minutes, revert the snapshot from the VMware UI (VM → Snapshot → Revert to Snapshot). Do not keep resetting the guest.

4. In Safe Mode, run Disable_Defender2.PS1

This is the script that disables Defender. Do not run __OLD__\disable_Defender.PS1. Confirm you are actually in Safe Mode before running it. Elevated PowerShell:

```
(Get-ItemProperty HKLM:\SYSTEM\CurrentControlSet\Control\SafeBoot\Option `
  -ErrorAction SilentlyContinue).OptionValue
```

1 means Minimal Safe Mode. No key means a normal boot. The title bar must also say Administrator.

Execution policy

A stock Windows 11 install is Restricted, so every .ps1 is refused until you bypass execution policy for this window only. The prompt defaults to N; answer Y. Do not change the machine policy. Process scope dies with the window.

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
.\Disable_Defender2.PS1
```

From an elevated Command Prompt the equivalent is powershell -NoProfile -ExecutionPolicy Bypass -File followed by the full path of Disable_Defender2.PS1. Pasting the script straight into elevated PowerShell also works, and interactive commands are not subject to execution policy.

The script (Disable_Defender2.PS1)

Run this file as it is. Do not edit it.

```
Get-ScheduledTask -TaskPath '\Microsoft\Windows\Windows Defender\' -ErrorAction SilentlyContinue |
  ForEach-Object { Disable-ScheduledTask -InputObject $_; $_.TaskName }

'WinDefend','WdFilter','WdBoot','WdNisSvc','WdNisDrv' | ForEach-Object {
  '{0} = {1}' -f $_, (Get-ItemProperty "HKLM:\SYSTEM\CurrentControlSet\Services\$_").Start
}
```

The first command disables every task under \Microsoft\Windows\Windows Defender\ and prints each task name. The second command prints Start for WinDefend, WdFilter, WdBoot, WdNisSvc, and WdNisDrv. Every one of those five lines must print = 4. Do not leave Safe Mode until it does. Those five are the engine and its drivers.

IF NO TASK NAMES PRINT

Minimal Safe Mode often does not start the Task Scheduler service. Get-ScheduledTask then returns nothing, and the Start lines can still print = 4. The Start lines are the gate for leaving Safe Mode. After the normal reboot, Check_Defender_Disabled.PS1 prints each task's State. State must be Disabled. If the scheduler was down and any task is still Ready, run Disable_Defender2.PS1 again from an elevated PowerShell window on the normal desktop. Disabling those tasks does not require WdFilter to be unloaded.

5. Leave Safe Mode

Elevated Command Prompt, still inside Safe Mode:

```
bcdedit /deletevalue {current} safeboot
bcdedit /enum {current}
shutdown /r /t 0
```

safeboot must be absent from the enum output before you reboot. If you used msconfig instead of bcdedit, open msconfig → Boot → clear Safe boot → OK → Restart. If the next boot lands in Safe Mode again, the delete did not stick: run bcdedit /deletevalue {current} safeboot again from that Safe Mode session.

On this guest the reboot from Safe Mode back to a normal desktop succeeded.

6. Verify after the normal boot

Elevated PowerShell. This is Check_Defender_Disabled.PS1. The last three lines are the task-state check: each Defender task's State, or an explicit line if none were returned. Same process-scope execution-policy bypass if you run the file.

```
Get-Service WinDefend, WdNisSvc, Sense | Format-Table Name, Status, StartType
Get-Process MsMpEng -ErrorAction SilentlyContinue
Get-MpComputerStatus | Select AMServiceEnabled, AntivirusEnabled, RealTimeProtectionEnabled, IsTamperProtected
Get-Service WinDefend
sc.exe qc WdFilter
sc.exe qc WdBoot
sc.exe query WdNisDrv
driverquery | findstr /i "WdFilter WdBoot WdNis"
fltmc
$defenderTasks = @(Get-ScheduledTask -TaskPath '\Microsoft\Windows\Windows Defender\' -ErrorAction SilentlyContinue)
if ($defenderTasks.Count -eq 0) { 'No tasks returned under \Microsoft\Windows\Windows Defender\' }
$defenderTasks | ForEach-Object { '{0} = {1}' -f $_.TaskName, $_.State }
```

How to read the output

Pass condition: WinDefend, WdNisSvc, and Sense are Stopped and Disabled; Get-Process MsMpEng prints nothing; Get-Service WinDefend shows Stopped. On this guest Get-MpComputerStatus also printed False for all four fields. That is sufficient to call the engine off. It is not sufficient to call the boot drivers unloaded, which is why the sc and fltmc checks exist. The task lines are a separate pass: every printed State is Disabled. “No tasks returned” is not a pass.

driverquery matches are not a failure

driverquery with no switch lists every driver registered on the system. It has no State column. A match only means the .sys file is still in the service database. That is what Start = 4 does. It does not delete them. sc.exe qc is configuration. Current state is a different query.

Command	What it answers	Pass on this guest
sc qc WdFilter / WdBoot	Configured start type	START_TYPE 4 DISABLED
sc query WdNisDrv	Running or not this boot	STATE 1 STOPPED, exit 1077
fltmc	Minifilters actually loaded	No WdFilter in the list
driverquery	Installed, not loaded	Matches expected; ignore as a fail
Task State lines	Scheduled tasks disabled	Each task = Disabled

Win32 exit 1077 is ERROR_SERVICE_NEVER_STARTED. The driver was not started this boot because Start is 4. It is not a crash. fltmc on this guest listed bindflt, UCPD, applockerfltr, storqosflt, wcifs, CldFlt, bfs, FileCrypt, luafv, UnionFS, npsvctrig, Wof, and FileInfo. Those are ordinary Windows filters. WdFilter was not among them.

DONE

Stopped and Disabled services, no MsMpEng.exe, WdFilter and WdBoot at start type 4, WdNisDrv stopped with 1077, no WdFilter in fltmc, and every Defender scheduled task State Disabled. That is a finished disable. Leave the .sys files where they are. Deleting them does not make the disable stronger; a later update or Windows File Protection will restore them. Smart App Control was not part of this and stays as it was.

What will try to undo this

A cumulative update or a Defender platform update can write the Start values back, especially on WdBoot, WdFilter, and WinDefend. Feature upgrades are the usual offender, not a simple reboot. On a lab VM, pause Windows Update (Settings → Windows Update → Pause), or re-run Disable_Defender2.PS1 from Safe Mode after any update that brings MsMpEng.exe back. Re-check with Check_Defender_Disabled.PS1, including the task State lines. Do not assume the disable survived an update.

Undo

Snapshot revert is the clean undo. Manual undo is the restore procedure in Restore_Defender_Windows11_Enterprise.pdf: write Start back to the values recorded on this guest, re-enable the tasks under \Microsoft\Windows\Windows Defender\, clear Group Policy if you set it, reboot once, then turn Tamper Protection back on. Do not use __OLD__\disable_Defender.PS1 for the undo.

Files used

File	When	Role
Disable_Defender2.PS1	Safe Mode, elevated. Again on the normal desktop only if task State is not Disabled.	Disables the Defender scheduled tasks and prints Start. This is the script that does the disable. Do not edit it.
Check_Defender_Disabled.PS1	Normal boot, elevated	Service, process, status, driver, fltmc, and each task's State.
__OLD__\disable_Defender.PS1	Not used	Archived. Not part of this procedure.

Run each live file from an elevated PowerShell window after Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass.

Order of operations

- Snapshot.
- Tamper Protection off, real-time protection off, in the normal session. Leave Smart App Control alone.
- Optional: gpedit "Turn off Microsoft Defender Antivirus" = Enabled. Do not reboot for the policy alone.
- bcdedit /set {current} safeboot minimal, from an elevated Command Prompt. Confirm with bcdedit /enum, then reboot.
- Elevated PowerShell, process-scope Bypass, Disable_Defender2.PS1. Require = 4 on WinDefend, WdFilter, WdBoot, WdNisDrv, and WdNisSvc.
- bcdedit /deletevalue {current} safeboot, confirm safeboot is gone, reboot.
- Check_Defender_Disabled.PS1. Ignore driverquery matches. Require fltmc without WdFilter, sc qc start type Disabled, and every task State Disabled.
- If the check says no tasks were returned, or any State is not Disabled, run Disable_Defender2.PS1 again on the normal desktop and re-check. The task step does not need WdFilter unloaded.