CHECKLIST

Windows 10 Hardening Checklist: Local Security Policy and GPO

A practical checklist for deciding which Windows 10 security controls belong in Local Security Policy, Group Policy, or both, with validation and rollback steps for operationally safe hardening.

Windows 10 Hardening Checklist: Local Security Policy and GPO

Use this checklist to harden Windows 10 endpoints without creating policy collisions, breaking support workflows, or losing track of the effective security state.

---

1) Define the control scope

- [ ] Identify the endpoint class: workstation, admin workstation, developer laptop, kiosk, lab system, or special-purpose device.
- [ ] Define the business purpose of the device and the support requirements it must preserve.
- [ ] Confirm whether the device is standalone, domain-joined, or managed through another centralized tool.
- [ ] Classify each control as:
- [ ] local-only
- [ ] domain-enforced
- [ ] environment-dependent
- [ ] exception-based

- [ ] Record the control owner and the approval path for exceptions.

2) Decide the source of truth

- [ ] Use Group Policy for settings that must remain consistent across managed endpoints.
- [ ] Use Local Security Policy for standalone systems, temporary baselines, test systems, or break-glass scenarios.
- [ ] Avoid depending on local settings as the long-term enforcement mechanism on domain-managed devices.
- [ ] Verify whether the same setting is already defined by another policy layer.
- [ ] Document which layer wins if both local and domain settings exist.

3) Review policy precedence and collision risk

- [ ] Check whether the setting is computer-based or user-based.
- [ ] Determine whether the control is additive, last-writer-wins, or behaves specially.
- [ ] Identify linked GPOs, inherited settings, and any enforced GPOs that may override the change.
- [ ] Look for overlapping controls in security baselines, administrative templates, or local templates.
- [ ] Confirm that no existing policy will undo the intended result during refresh.

4) Build the baseline carefully

- [ ] Define mandatory controls in a domain baseline.
- [ ] Keep the baseline as strict as possible without breaking sign-in, management, recovery, or application access.
- [ ] Use scoped policies or security groups for legitimate exceptions.
- [ ] Keep exception handling explicit, documented, and reviewable.

- [ ] Avoid one-size-fits-all settings for mixed-role fleets.

5) Apply changes in a test scope first

- [ ] Test the setting on a non-production device or pilot group.
- [ ] Use a device that matches production as closely as possible.
- [ ] Confirm the test system has the same VPN, endpoint protection, encryption, and management dependencies as production.
- [ ] Apply only one meaningful change at a time when possible.
- [ ] Record the exact settings changed and the expected outcome.

6) Validate the effective policy on the endpoint

- [ ] Confirm the policy was applied after refresh.
- [ ] Verify the resulting setting on the device, not only in the console.
- [ ] Check whether the setting exists locally, in domain policy, or both.
- [ ] Confirm audit logging, privilege assignment, and authentication behavior still match expectations.
- [ ] Validate that administrative access still works.
- [ ] Confirm remote support tools still function if they are required.
- [ ] If the setting affects logon or user rights, test with a standard user and an admin account.

7) Check for operational side effects

- [ ] Verify that sign-in is not broken.
- [ ] Verify that device management agents still communicate.
- [ ] Verify that remote access paths still work.
- [ ] Verify that software installation workflows still work where required.

- [ ] Verify that application access for the affected role is intact.
- [ ] Check for unexpected changes in event logs or audit results.
- [ ] Confirm that the change does not create a hidden support burden.

8) Coordinate adjacent security controls

- [ ] Review encryption and boot-protection dependencies before applying hardening changes.
- [ ] Align the baseline with BitLocker, TPM, and recovery-related policy where relevant.
- [ ] Confirm recovery procedures are documented and tested.
- [ ] Ensure boot or recovery prompts will not be mistaken for unrelated policy failures.
- [ ] Validate that endpoint security controls work together rather than against each other.

9) Promote to production safely

- [ ] Expand from pilot to broader scope only after validation succeeds.
- [ ] Use phased rollout by department, role, or device tier.
- [ ] Monitor for authentication failures, support incidents, and policy refresh issues.
- [ ] Keep a rollback plan available before broad deployment.
- [ ] Track when the policy was deployed and who approved it.

10) Document the result

- [ ] Record the final effective setting.
- [ ] Note whether the setting is managed locally, by GPO, or by both.
- [ ] Save the validation evidence.
- [ ] Document the reason for any exception.
- [ ] Include rollback steps and the conditions that would trigger them.

- [ ] Review the baseline periodically for drift, exceptions, and changing business requirements.

Quick decision guide

Use this simple rule set when choosing where a control belongs:

- Local Security Policy: standalone devices, lab systems, staging systems, or emergency local recovery states.
- Group Policy: managed fleet controls, repeatable baselines, and settings that must not drift.
- Both: only when you intentionally use local configuration as a starting state and GPO as the enforcement layer.

Pre-deployment questions

- [ ] What exact problem is this control solving?
- [ ] Which users, devices, or roles are affected?
- [ ] What else depends on this behavior?
- [ ] How will we confirm the effective state on the endpoint?
- [ ] What is the rollback path if the control causes a sign-in or support issue?
- [ ] Is there a documented exception process?

Rollback checklist

- [ ] Identify the setting to revert.
- [ ] Restore the previous known-good value.

- [ ] Remove or disable the pilot GPO link if needed.
- [ ] Force a policy refresh only after the rollback is staged.
- [ ] Revalidate logon, management, and support access.
- [ ] Capture the root cause and update the hardening baseline if appropriate.

Validation notes

Use these fields during rollout:

- Device name:
- Device role:
- Managed by:
- Policy source of truth:
- Setting changed:
- Expected effect:
- Actual effect:
- Validation date:
- Validator:
- Rollback owner:
- Exception approved by:

Success criteria

A Windows 10 hardening change is successful when:

- [ ] the intended control is enforced on the endpoint
- [ ] the change does not break sign-in, management, or support workflows
- [ ] the effective state matches the intended baseline

- [ ] exceptions are documented and reviewable
- [ ] rollback steps exist and are tested

Final reminder

Do not harden from the console alone. Harden the endpoint, validate the result, and only then expand the scope.