MAESTER HARDENING GUIDE

A Practical, Prioritized Playbook for Microsoft 365 & Entra ID Security Testing

Thomas Forde

Version 2.0 — August 2026

This is an independent work and is not affiliated with, endorsed by, or sponsored by the Maester project or its maintainers. Maester, Microsoft 365, Entra ID, and related product and service names are used for identification purposes only and remain the property of their respective owners.

Every command in this guide is provided for informational and diagnostic purposes. Review each command before running it, understand what it queries or changes, and test in a non-production environment first wherever possible. The author assumes no liability for any outage, data loss, access disruption, or other consequence arising from use of the information in this guide — you are responsible for the changes you make to your own environment.

Questions, feedback, or support: thomas.forde@outlook.com

This is a free sample. It includes the full guide's Sections 1-6 (methodology, installation, operating model, and the complete prioritized roadmap) plus one full domain from the appendix as a taste of the pre-flight format. The full edition (Version 2.0) continues with 11 more M365/Entra domains (402 more tests) and an entirely new Part II covering Maester's on-premises Active Directory module (currently in preview upstream) — 269 more tests across 19 categories.

What's Inside The Full Guide

The sections below with a checkmark are included in full in this sample. Everything else is in the full edition.

1. What Is Maester

Maester is an open-source PowerShell test automation framework (built on Pester) that turns Microsoft 365 and Entra ID security best practices into runnable, repeatable tests. Instead of manually walking through admin portals during an audit, you run a command, and Maester checks your tenant's actual configuration against a curated set of security baselines, then produces an interactive HTML report with pass/fail status and remediation guidance for every check.

As of version 2.2.0, Maester ships 407 built-in tests across five suites:

- **Maester (167 tests) —** Community-curated best practices covering Conditional Access, privileged access, Exchange, Teams, Intune, Defender, Purview, and — as of recent releases — AI agent governance for Copilot Studio.
- **Entra ID Security Config Analyzer / EIDSCA (44 tests) —** Checks mapped to the Entra ID Attack and Defense Playbook — authentication methods, consent settings, password protection.
- **CISA SCuBA (79 tests) —** The US Cybersecurity and Infrastructure Security Agency's Secure Cloud Business Applications baseline for Entra ID, Exchange Online, and SharePoint.
- **CIS Microsoft 365 Foundations Benchmark (50 tests) —** Industry-standard CIS benchmark controls, including a small GitHub-specific subset.
- **ORCA (67 tests) —** Exchange Online Protection and Defender for Office 365 configuration checks, ported from the long-standing ORCA project.

Maester does not change anything in your tenant — it only reads configuration and reports on it. Every test ships with a direct link into the relevant admin portal so remediation is one click away.

2. How to Use This Guide

This guide is split into two parts:

- **Section 6, the prioritized roadmap —** a curated, five-phase sequence covering the highest-impact tests. Work through this first regardless of your tenant's size or maturity.
- **Section 7, the full appendix —** every one of the 407 tests, grouped into 12 practical domains and sorted by severity, each with a Risk / Run this / What it means block (module setup in Section 5) so you can track complete coverage over time or hand sections to different owners (e.g., the Exchange admin owns "Email & Messaging Security").

Severity levels (Critical / High / Medium / Low / Info) are Maester's own classification. "Unknown" means the test doesn't carry an explicit severity tag upstream — treat these as Medium until you've reviewed them.

3. Installation & First Run

Run the following in an elevated PowerShell session:

```
Install-Module Pester -MinimumVersion 5.7.1 -MaximumVersion 5.7.1 -SkipPublisherCheck
-Force -Scope CurrentUser

Install-Module Maester -Scope CurrentUser

md maester-tests; cd maester-tests

Install-MaesterTests
```

Then connect to your tenant and run the tests:

```
Connect-Maester   # add -Service All to include Exchange, Teams, Azure, SharePoint

Invoke-Maester
```

The CISA and ORCA suites require additional modules (ExchangeOnlineManagement, MicrosoftTeams, Az.Accounts, PnP.PowerShell) and are skipped automatically if those aren't connected — install them if you want full coverage:

```
Install-Module Az.Accounts, ExchangeOnlineManagement, MicrosoftTeams, PnP.PowerShell -Scope CurrentUser
```

Permissions note: read-only access is sufficient. For Exchange Online, the account needs View-Only Configuration or O365SupportViewConfig at minimum — you do not need to grant write access to run Maester.

4. Operating Model — Run This Continuously

A single scan tells you where you stand today. The value compounds when Maester runs on a schedule and someone actually looks at regressions:

- **GitHub Actions or Azure DevOps Pipelines —** schedule a daily or weekly run using Workload Identity Federation (no stored secrets, no credential rotation). Maester's docs include ready-made workflow templates for both.
- **Azure Automation / Container App Jobs —** for teams that want the scan running inside their own Azure subscription rather than in CI/CD.
- **Alerts —** configure email or Microsoft Teams alerts so a newly-failing test reaches someone within a day. Config drift — a Conditional Access exclusion added "temporarily" for a support ticket and never removed — is the most common way a hardened tenant quietly regresses.
- **Archive history —** the GitHub Actions integration keeps a run history automatically, which doubles as audit evidence for SOC 2, ISO 27001, or internal control testing.

5. How to Audit Before You Enable Anything

This is the part Maester's own documentation doesn't cover. Every test tells you the desired end state and links to a Microsoft Learn article on how to configure it — but flipping a security control on a live tenant without checking who depends on the current behavior is how hardening projects cause outages and lockouts.

Every one of the 407 entries in Section 7 carries three parts: the specific Risk, an actual command to Run first (PowerShell/Graph/KQL — not a pointer to "go check a report"), and What the result means in plain terms. You should be able to run the command as written (with your own object IDs/domain names substituted in) and get a direct answer, not a reason to go open another tool. Connect the right modules first so those commands actually work:

- **Microsoft Graph PowerShell SDK —** `Install-Module Microsoft.Graph -Scope CurrentUser` then `Connect-MgGraph -Scopes "Directory.Read.All,AuditLog.Read.All,Policy.Read.All,RoleManagement.Read.Directory,UserAuthenticationMethod.Read.All"`. Covers every `Get-Mg*` command in this guide (users, sign-in logs, Conditional Access, PIM/role assignments, authentication methods).
- **Exchange Online PowerShell —** `Install-Module ExchangeOnlineManagement -Scope CurrentUser` then `Connect-ExchangeOnline`. Covers every `Get-MessageTrace`, `Get-*Policy`, `Get-Mailbox`, `Get-TransportRule` command.

- **SharePoint Online Management Shell —** `Install-Module Microsoft.Online.SharePoint.PowerShell -Scope CurrentUser` then `Connect-SPOService -Url https://<tenant>-admin.sharepoint.com`. Covers every `Get-SPO*` command.
- **Microsoft Teams PowerShell —** `Install-Module MicrosoftTeams -Scope CurrentUser` then `Connect-MicrosoftTeams`. Covers every `Get-Cs*` command.
- **Az PowerShell —** `Install-Module Az -Scope CurrentUser` then `Connect-AzAccount`. Covers `Get-AzRoleAssignment` and other Azure-scope commands.

Two checks in Section 7 reference a workflow rather than a single command, because the tooling itself is the check — noted here once instead of repeated 70+ times:

- **Conditional Access Report-only mode + What-If —** any entry that says "deploy in Report-only first" means: set the policy's state to Report-only, let it run for 1–2 weeks, then review sign-in logs for entries where AppliedConditionalAccessPolicies shows reportOnlyFailure for that policy (the exact query is in the relevant appendix entries). Use What-If (Entra admin center → Conditional Access → What If) to test specific users, including break-glass, before switching to On.
- **Intune/Defender Audit mode —** any entry that says "deploy in Audit mode first" means: set Attack Surface Reduction rules or Application Control to Audit rather than Block/Enforce, then review what would have been blocked in Defender/Intune reporting (or the Windows CodeIntegrity event log for App Control) for 1–2 weeks before switching to enforced.

One rule that applies regardless of which specific check is relevant: never make an irreversible, tenant-wide change on a Friday afternoon. Run the command, read the verdict, stage it, then enforce it.

Field Notes: Lessons From a Real Multi-Tool Audit

The patterns below came out of running Maester alongside on-prem AD and endpoint scanners against a live production tenant over several months, tracking every finding through investigation, remediation, or documented accept-risk. They're generalized here — no tenant, company, or account detail — because the pattern is what's reusable, not the specific finding. All four map most directly onto MT.1024 (full guide, Section 7), the umbrella recommendation test that tends to produce exactly this kind of ambiguity, but the underlying discipline applies across the whole catalog.

> **FIELD NOTE — A "failed" MFA check can mean the account can't register, not that it's unprotected.**
>
> An admin-MFA recommendation flagged several role-holding accounts as "not protected by MFA." All three turned out to be service accounts — automation, a reporting tool, a break-glass identity — none of which can complete an interactive MFA registration in the first place, so this class of check will always show them as failing.
>
> Before treating a finding like this as an emergency, check whether the flagged identities are actually human interactive sign-ins. If they're service accounts, the real question is whether Conditional Access (or an equivalent control) blocks them from interactive/admin-portal sign-in entirely — if so, the exposure is already contained, and the correct remediation is service-account hygiene (remove the standing admin role, move to a managed identity or certificate auth, or formally document it as break-glass), not chasing MFA registration on an account that structurally can't complete it.
>
> **FIELD NOTE — Resolve the identifier before you escalate a privileged-access finding.**
>
> A finding reporting non-admin accounts with directory-replication rights (the kind of access that can be abused to dump credential hashes) initially looked like a live, high-severity exposure — several grants existed that didn't match any expected admin or sync account.
>
> Resolving each identifier against the directory showed all of them were orphaned — leftover access-control entries from accounts deleted long ago, not live accounts. The real, lower-urgency finding was a decommissioning-hygiene gap (permissions not cleaned up when accounts are removed), not an active compromise. The lesson: a scary-sounding finding naming an unfamiliar identifier is a resolve-first, react-second situation. Confirm the identifier is a

live, reachable account before treating the finding as urgent — and once you've found one orphaned grant, check for a broader pattern of the same cleanup gap elsewhere.

FIELD NOTE — A single check can hide inconsistent configuration across your fleet.

A hardening check covering a specific service on domain controllers came back failing tenant-wide. Checking each machine individually showed the setting was already correctly applied on the newer servers and only missing on an older cohort — the tenant wasn't uniformly unhardened, one deployment wave had simply never picked up a since-adopted baseline.

Any check that reports a single pass/fail for a group of objects (devices, mailboxes, service principals) can hide this kind of split result. Before remediating, enumerate the individual objects behind the aggregate finding — the fix is usually smaller and more targeted than the summary implies, and knowing which objects still need it tells you where your provisioning process has a gap worth closing structurally (a baseline GPO or Intune profile) so newly built systems don't reintroduce the same gap.

FIELD NOTE — Coverage-percentage checks can be skewed by stale inventory, not real gaps.

A local-admin-password-management coverage check reported well under full coverage. Auditing the actual device population showed real coverage was meaningfully higher than the raw percentage suggested — a sizeable share of the "uncovered" denominator was stale device objects (decommissioned or re-imaged machines still present in the directory), not systems genuinely missing protection.

Any test that reports a ratio (X of Y covered, X of Y compliant) is only as accurate as Y. Before treating the gap as the full remediation scope, separate genuinely active objects from directory cruft — the real work is usually a smaller, more specific list than the headline number implies, and cleaning up the stale objects is worth doing anyway.

6. Prioritized Hardening Roadmap

407 tests is not a to-do list you clear in a week. These five phases sequence the work by blast-radius: identity first (it's the perimeter now), then privileged access, then the two channels attackers actually use to get in (email, file sharing), then endpoints, then the newer AI-governance surface. Each phase lists representative test IDs — cross-reference the full ID in Section 7 for the exact remediation link and pre-flight check.

Phase 1 — Identity & Access Foundations

Highest impact, usually lowest effort. Do this before anything else.

- **CISA.MS.AAD.1.1 / MT.1009 / MT.1010 —** Block legacy authentication entirely, including a dedicated Conditional Access rule for Exchange ActiveSync (the protocol most often missed).
- **CISA.MS.AAD.3.1 / 3.2 —** Enforce phishing-resistant MFA (FIDO2, Windows Hello for Business, Authenticator with number matching) for all users; require at least one strong MFA method everywhere else.
- **CISA.MS.AAD.3.5 / EIDSCA.AS04 / EIDSCA.AV01 —** Disable SMS, voice call, and email one-time-passcode as sign-in methods — all three are vulnerable to SIM-swap and real-time phishing.
- **MT.1006 / MT.1007 / MT.1008 —** Conditional Access requires MFA for admins, for all users, and specifically for Azure management (portal, CLI, PowerShell).
- **MT.1005 / MT.1034 —** Every Conditional Access policy excludes a documented emergency/break-glass account, and that account is verified to never be blocked.
- **CIS.M365.1.1.1 / CISA.MS.AAD.7.3 —** Administrative accounts are cloud-only — never synced from an on-premises directory.
- **CIS.M365.1.1.3 / MT.1032 / CISA.MS.AAD.7.1 —** Between two and four Global Administrators are assigned — no more, no fewer.
- **CISA.MS.AAD.6.1 / CIS.M365.1.3.1 —** Passwords are set to never expire (current guidance: MFA and Smart Lockout replace forced rotation, which pushes users toward weaker, predictable passwords).

- **MT.1021 —** Security Defaults are enabled as a floor if you don't yet have Entra ID P1/P2 for full custom Conditional Access.

Phase 2 — Privileged Access & Governance

Close the paths an attacker uses once they land a foothold.

- **MT.1031 / CISA.MS.AAD.7.5 —** All privileged roles are managed exclusively through Privileged Identity Management (PIM) — no standing access.
- **CISA.MS.AAD.7.4 / MT.1025–1028 —** No permanent role assignments on control-plane roles for external users, hybrid users, service principals with client secrets, or mailbox-holding accounts.
- **CISA.MS.AAD.7.6 —** Global Administrator activation requires approval, not just MFA.
- **CISA.MS.AAD.7.7 / 7.8 / 7.9 —** Alerts fire automatically whenever a privileged role is activated.
- **MT.1029 —** Stale/inactive accounts are removed from privileged roles on a schedule.
- **CISA.MS.AAD.5.1–5.4 —** Application registration and consent are restricted to administrators; group owners cannot grant consent.
- **MT.1050 / MT.1051 / MT.1063 —** No application — directly or indirectly — has a permission path to Global Admin, and every app registration owner has MFA registered.

Phase 3 — Email & Collaboration Security

Email and file sharing remain the top two initial-access vectors.

- **CISA.MS.EXO.5.1 —** Disable SMTP AUTH tenant-wide; re-enable per-mailbox only for a verified legacy device.
- **CISA.MS.EXO.1.1 —** Block automatic forwarding to external domains.
- **CISA.MS.EXO.4.2 / MT.1182 —** Publish DMARC with p=reject (not just monitor) once SPF and DKIM are validated for every domain.
- **CISA.MS.EXO.13.1 / MT.1172 —** Mailbox auditing and unified audit log ingestion are enabled — you cannot investigate an incident you never logged.
- **ORCA.104 / ORCA.140 / ORCA.142 —** High-confidence phishing, high-confidence spam, and phishing actions are all set to Quarantine, not just tagged.
- **ORCA.114 / ORCA.228 / ORCA.239 —** No IP allow-lists, no blanket trusted senders, and no exclusions on the built-in protection policies — the three most common ways admins unintentionally punch holes in Defender for Office 365.
- **CISA.MS.SHAREPOINT.1.1–1.3 —** External sharing in SharePoint and OneDrive is limited to existing guests or explicitly approved domains, never "Anyone."
- **CIS.M365.8.4.1 —** Third-party and custom Teams apps are blocked by default; allow only what has been reviewed.

Phase 4 — Devices & Endpoint Security

The Defender and Intune baseline most tenants have partially drifted from.

- **MT.1123 / MT.1178 / MT.1179 —** BitLocker enforced via Intune, Attack Surface Reduction rules configured, and App Control for Business enabled — the three biggest wins against ransomware on managed Windows endpoints.

- **MT.1148–1171 —** Defender's core engine settings — real-time monitoring, cloud protection, behavior monitoring, PUA protection, network protection — are all enabled. Maester checks 24 individual Defender settings; most tenants have at least a few off by default or drifted over time.
- **MT.1014 —** Conditional Access requires a compliant or hybrid-joined device specifically for admin sign-in.
- **MT.1092–1097 —** Device-management plumbing itself is healthy: APNS, Apple ADE/VPP certificates are not close to expiry, and the Android Enterprise connection is active. If these expire silently, device management quietly stops working tenant-wide.

Phase 5 — Data Protection, AI Governance & Continuous Monitoring

Close the newest gap (Copilot/AI agents) and stop doing this as a one-off.

- **MT.1172 / MT.1175 / MT.1176 —** Unified audit log, DLP, and retention policies all explicitly cover the Microsoft 365 Copilot location — a gap almost every tenant currently has.
- **MT.1113–1122 —** If Copilot Studio or other AI agents are in use: require user authentication, avoid overly broad sharing, remove hard-coded credentials from topics, and review any MCP server tools before they're connected.
- **Operationalize Maester —** Schedule it to run continuously via GitHub Actions, Azure DevOps Pipelines, or Azure Automation rather than running it as a one-off scan. Configuration drifts — a policy someone "temporarily" loosens for a support ticket is the single most common way a hardened tenant regresses.
- **Alerting —** Wire up email or Teams alerts so a failed test reaches someone within a day, not at the next quarterly review.

7. Full Test Catalog by Domain — With Pre-Flight Checks (All 407 Tests)

Every test currently shipped in Maester v2.2.0, grouped into 12 practical domains (my grouping — not an official Maester taxonomy) and sorted by severity. Each entry has three parts: the specific Risk of changing that setting, a command to Run first (connect the modules listed in Section 5 first), and what the result means — a genuine yes/no answer, not a pointer to go check somewhere else. Where a setting carries no real risk, the entry says so directly ("safe to enable, no check needed") instead of manufacturing busywork. Use this section as a coverage tracker: work down each domain, run the check, confirm the finding in the Maester HTML report, remediate or formally accept the risk, and re-run.

This sample includes one full domain below, unedited, exactly as it appears in the full guide. The full edition continues with 11 more domains covering all 407 tests.

Guest & External Collaboration (5 tests)

CISA.MS.AAD.8.2 **High · CISA** Only users with the Guest Inviter role SHOULD be able to invite guest users.

> **Risk:** Breaks self-service guest invites for anyone not granted Guest Inviter.
>
> **Run this first:**
>
> ```
> Get-MgAuditLogDirectoryAudit -Filter "activityDisplayName eq 'Invite external user'" -All | Select InitiatedBy | Sort -Unique
> ```
>
> **What the result means:** This is everyone who's actually invited a guest recently. Grant them the Guest Inviter role explicitly before restricting the tenant-wide default.
>
> **How to change it:** Entra admin center > Identity > External Identities > External collaboration settings > Guest invite settings > set "Members can invite" to No, leaving only "Admins and users in the guest inviter role can invite" active. Before flipping this, explicitly assign the Guest Inviter role (Identity > Roles & administrators > Guest Inviter) to everyone the check command shows has been legitimately inviting guests, so their workflow doesn't break the moment the tenant-wide default changes.

CISA.MS.AAD.8.1 **Medium · CISA** Guest users SHOULD have limited or restricted access to Azure AD directory objects.

> **Risk:** Can break a guest-facing app querying the directory.
>
> **Verdict:** Test against your most guest-dependent internal app/portal with a pilot guest account before rolling out broadly.
>
> **How to change it:** Entra admin center → Identity → External Identities → External collaboration settings → "Guest user access" → set to "Guest users have limited access to properties and memberships of directory objects."

CISA.MS.AAD.8.3 **Medium · CISA** Guest invites SHOULD only be allowed to specific external domains that have been authorized by the agency for legitimate business purposes.

> **Risk:** Breaks collaboration with any partner domain not yet on the approved list.
>
> **Run this first:**
>
> ```
> Get-SPOExternalUser -PageSize 200 | Select DisplayName, Email | ForEach-Object { ($_.Email -split "@")[1] } | Sort -Unique
> ```
>
> **What the result means:** This is every external domain currently collaborating with you. Get business sign-off on this exact list before enabling the allow-list.
>
> **How to change it:** Entra admin center > Identity > External Identities > External collaboration settings > Collaboration restrictions > select "Allow invitations only to the specified domains (most restrictive)" and add each approved domain. Build the list from the check command's output and get business sign-off per domain before enabling -- any domain not listed gets cut off immediately, including active collaborations.

CIS.M365.5.1.3.1 **Unknown · CIS** Ensure a dynamic group for guest users is created

> **Risk:** Additive; no restriction by itself.
>
> **Verdict:** Safe to create directly; verify the dynamic membership rule (userType eq 'Guest') actually captures guest accounts after creating it.

> **How to change it:** Entra admin center → Identity → Groups → New group → Membership type: Dynamic User, with rule `(user.userType -eq "Guest")`.

CIS.M365.5.1.6.2 Unknown · CIS Ensure that guest user access is restricted

> **Risk:** Can break guest-facing workflows relying on broader default access.
>
> **Run this first:**

```
Get-MgUser -Filter "userType eq 'Guest'" -All | Measure-Object
```

> **What the result means:** Test against your most guest-dependent app before restricting broadly.
>
> **How to change it:** Same control as EIDSCA.AP07 elsewhere in this catalog: Entra admin center > Identity > External Identities > External collaboration settings > Guest user access restrictions > move from "same as members" to a more limited level.

Sample Ends Here

That's 1 of 12 domains — 5 of the 407 tests in the full guide. The full edition continues with:

- Identity & Authentication Foundations — 111 tests
- Email & Messaging Security — 116 tests
- Devices & Endpoint Security — 49 tests
- Privileged Access & Identity Governance — 39 tests
- Conditional Access & Access Control — 34 tests
- SharePoint & OneDrive — 14 tests
- Teams & Collaboration — 10 tests
- AI Agents & Copilot Governance — 10 tests
- Applications, Consent & Service Principals — 7 tests
- Azure Platform & Governance — 7 tests
- Data Protection & Compliance — 5 tests

Plus Section 8 (AI-Assisted Custom Audits From Your Own Report — using Claude Code to turn your actual Maester findings into tenant-scoped scripts) and Section 9 (Next Steps).

Get the full guide: thomas.forde@outlook.com