Configure Phishing-Resistant MFA Policy in Entra ID

Field	Details
Document Type	How-To Guide / Runbook
Applies To	Microsoft Entra ID, Microsoft 365, Azure portals, Cloud Apps integrated with Entra ID
Audience	Entra ID / Security Administrators (2nd Line / Systems Admin / Security Ops)
Author	AK. Udofeh
Last Updated	March 2026

Overview

This document explains how to configure a **phishing-resistant MFA** Conditional Access policy using **Authentication Strengths** in Microsoft Entra ID. It covers report-only testing, scope control, and safe enforcement for privileged roles and high-value applications.

Background

- Standard Conditional Access policies using **“Require multi-factor authentication”** allow weaker methods (SMS, phone call, basic app OTP) which are still vulnerable to phishing and adversary-in-the-middle attacks.
- Organisations need to ensure that privileged accounts and sensitive applications can only be accessed with **phishing-resistant** methods such as FIDO2 security keys or other certified resistant mechanisms.
- Without Authentication Strength-based policies, admins cannot reliably enforce use of only strong, passwordless MFA for high-risk scenarios.

Basic steps that do **not** resolve this:

- Enabling MFA per user or via a simple Conditional Access “Require MFA” without specifying authentication strength.
- Relying solely on security defaults for tenants with advanced security requirements.

Other affected services/systems:

- Admin portals (Entra admin centre, Azure portal, Exchange admin centre), Microsoft 365 apps, and any SSO-integrated SaaS app federated via Entra ID will be subject to this policy when in scope.

Usecase

- Older Conditional Access policies were designed before **Authentication Strengths** and only distinguished “MFA vs no MFA”, not **which** MFA methods are acceptable.
- Attackers now commonly bypass weak MFA using real-time phishing and token replay, so stronger, phishing-resistant methods must be enforced where feasible.

Desired End State

1. Admins enable passwordless / phishing-resistant methods (Authenticator phone sign-in, FIDO2, etc.).
2. Admins create Conditional Access policies that require **Phishing-resistant MFA strength** or **Passwordless MFA strength** for sensitive scenarios.
3. Users in scope can satisfy the policy only with compliant strong methods; weaker methods are rejected.

What currently breaks:

- Existing CA policies allow weak MFA methods, so users can still sign in with SMS or legacy app codes, leaving privileged access exposed to phishing.

Known triggers:

- High-risk user roles (Global Admin, Security Admin, Exchange Admin, etc.).
- Access to high-value applications, admin portals, or from risky locations/devices.

Before You Start

Check	Where
Confirm passwordless / phishing-resistant methods are configured and piloted (Authenticator phone sign-in, FIDO2, etc.)	See the previous document and Entra > Protection > Authentication methods
Ensure you have break-glass accounts excluded from Conditional Access	Entra admin centre > Users / Groups list
Verify you have **Conditional Access Administrator** or equivalent permissions	Entra admin centre > Roles and administrators
Identify target scope (directory roles/groups/apps) and rollout phases	Security design docs/identity architecture
Confirm security defaults or conflicting policies won’t block testing	Entra > Protection > Conditional Access > Policies

Implementation Steps

1. Create a Report-Only Phishing-Resistant Policy

1. Sign in to **https://entra.microsoft.com** as a Conditional Access or Security Administrator.
2. Go to **Entra ID > Conditional Access > Policies** and select **New policy**.
3. Name the policy clearly, for example: `CA01 - Phishing-resistant MFA for Admins (Report-only)`.
4. Under **Assignments > Users**, select **Directory roles** and choose roles such as Global Administrator, Security Administrator, Exchange Administrator, and other privileged roles. ***(Ensure to exclude break glass and service accounts)***
5. Under **Cloud apps or actions**, select **All cloud apps** or a subset of high-value apps (Exchange Online, SharePoint Online, Entra admin centre, Azure portal).
6. Under **Conditions**, configure any additional conditions as needed (e.g. include all device platforms initially, or exclude trusted Workload identities).
7. Under **Access controls > Grant**, choose **Require authentication strength** and select **Phishing-resistant MFA strength**.
8. Set **Enable policy** to **Report-only**.

> This allows sign-ins to proceed but logs whether they would satisfy the policy.

9. Click **Create** to save the policy.

2. Monitor Report-Only Results

1. Allow at least some days of normal usage for covered users.
2. In the Entra admin centre, go to **Entra ID > Conditional Access > Insights and reporting** and review the policy's impact.
3. Use filters such as **Policy not satisfied** to identify sign-ins that would be blocked if enforced (e.g. users still using SMS).
4. Export reports for further analysis and plan user remediation or enrolment in passwordless methods where required.

3. Create / Adjust Production Enforcement Policy

1. Duplicate the tested report-only policy or create a new production policy with the same conditions.
2. Under **Assignments > Users**, keep scope limited initially (e.g. only core admin roles or a pilot group).

> Avoid tenant-wide enforcement until adoption is high.

3. Under **Assignments > Users > Exclude**, add:
    - Break-glass accounts.
    - Conditional Access / Global Admins used for emergency recovery (per your policy).

4. Under **Access controls > Grant**, ensure **Require authentication strength** is set to **Phishing-resistant MFA strength** (or a custom

strength including FIDO2, etc.).

5. Set **Enable policy** to **On**.

6. Save the policy and notify in-scope admins to use their phishing-resistant methods going forward.

4. (Optional) Use Authentication Strength with Passwordless MFA

- For broader user populations that may not yet be able to use fully phishing-resistant methods, create another Conditional Access policy:
  - **Grant > Require authentication strength > Passwordless MFA strength** for high-value apps, allowing strong but not necessarily certified phishing-resistant methods.
- Apply similar report-only and enforcement phases to minimise disruption.

Automated / Script Option (Graph PowerShell Example)

> Example policy that creates a **report-only** Conditional Access policy requiring phishing-resistant MFA strength for key admin roles.

```
# Connect to Microsoft Graph with appropriate scopes
Connect-MgGraph -Scopes "Policy.Read.All","Policy.ReadWrite.ConditionalAccess"

# Built-in Phishing-resistant MFA authentication strength ID
$phishResistantStrengthId = "00000000-0000-0000-0000-000000000004"

# Define key admin roles (template IDs)
$adminRoles = @(
    "62e90394-69f5-4237-9190-012177145e10", # Global Administrator
    "194ae4cb-b126-40b2-bd5b-6091b380977d", # Security Administrator
    "29232cdf-9323-42fd-ade2-1d097af3e4de"  # Exchange Administrator
)

# Build the Conditional Access policy body
$policyParams = @{
    DisplayName = "Require phishing-resistant MFA for admins (Report-only)"
    State       = "enabledForReportingButNotEnforced" # Report-only mode
    Conditions  = @{
        Users = @{
```

```
            IncludeRoles = $adminRoles             # Target listed directory roles
        }
        Applications = @{
            IncludeApplications = @("All")         # All cloud apps
        }
    }
    GrantControls = @{
        Operator = "OR"
        AuthenticationStrength = @{
            Id = $phishResistantStrengthId        # Built-in Phishing-resistant MFA strength
        }
    }
}

# Create the Conditional Access policy
New-MgIdentityConditionalAccessPolicy -BodyParameter $policyParams
```

Deployment notes:

- Run as a privileged admin with permission to manage Conditional Access.
- Adjust `$adminRoles` and `IncludeApplications` to suit your environment.

Expected output / success indicators:

- `New-MgIdentityConditionalAccessPolicy` returns the created policy object with `state` set to `enabledForReportingButNotEnforced` and the correct authentication strength ID.

Script Breakdown

- `Connect-MgGraph` establishes a session with Microsoft Graph using scopes for Conditional Access policy management.
- `$phishResistantStrengthId` holds the known ID of the built-in **Phishing-resistant MFA strength**, which defines allowed methods such as FIDO2 keys.
- `$adminRoles` lists the directory role template IDs for privileged admin roles that will be included in the policy.
- `$policyParams` defines policy properties:
  - `DisplayName` gives a descriptive name.
  - `State` as `enabledForReportingButNotEnforced` places the policy in report-only mode.
  - `Conditions.Users.IncludeRoles` targets specific directory roles.
  - `Conditions.Applications.IncludeApplications = "All"` targets all cloud apps.
  - `GrantControls.AuthenticationStrength.Id` references the phishing-resistant MFA strength.
- `New-MgIdentityConditionalAccessPolicy` sends the policy definition to Graph and creates the policy in Entra ID.

Troubleshooting

Users are blocked after policy enforcement

- Use **Sign-in logs** to verify which policy blocked access and what authentication method was used.
- Check whether affected users have registered any phishing-resistant methods (FIDO2, etc.). If not, move them temporarily out of scope or provide registration guidance.

Policy does not appear to have any effect

- Confirm the policy is **Enabled** (not just in report-only) for enforcement scenarios.
- Verify the user and app in question are actually within **Assignments** and not excluded by another condition or exclusion.

Graph script fails with insufficient privileges

- Ensure the account running the script has Conditional Access Administrator or equivalent role and that Graph scopes include `Policy.ReadWrite.ConditionalAccess`.
- If using app-only auth, ensure the app registration has `Policy.ReadWrite.ConditionalAccess` application permissions and admin consent granted.

Conflicts with existing policies

- Review other Conditional Access policies in **What If** or simulation tools to check aggregate effect.
- Consolidate overlapping policies or adjust priority/scope to avoid unexpected blocks.

Expected Outcome

Item	Detail
Resolution time	30–60 minutes to create report-only and enforcement policies plus a 7–10 day observation window
User impact	Privileged users and scoped users must use phishing-resistant or passwordless MFA methods to access targeted apps; legacy MFA may no longer be accepted
Recurrence risk	Low once adopted; policies should be reviewed periodically as roles, apps, and methods evolve

---

Revision #3
Created 2026-03-27 12:08:50 UTC by AK. Udofeh
Updated 2026-03-31 14:45:33 UTC by AK. Udofeh