MIRO - Azure Cards

**Report Reference: CYSEC-SECREP-05**

**Version**: 1.0

**Prepared by**: David Biayna Neal

**Report Date**: 20th August 2025

**Classification**: Sensitive

**Document Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version No. | Date of Issue | Authors | Status | Comments |
| 0.1 | 20th August 2025 | David Biayna Neal | Initial Draft |  |

Contents

[1 Introduction 4](#_Toc206576700)

[2 Executive Summary 5](#_Toc206576701)

[3 Scope 6](#_Toc206576702)

[4 Background 7](#_Toc206576703)

[5 Security Recommendations 11](#_Toc206576704)

[6 Limitations and Constraints 12](#_Toc206576705)

[7 Results of Assessment 13](#_Toc206576706)

# Introduction

MHR Cyber Security conducted a security assessment on Miro’s feature of Azure Cards, which allows to Connect Azure Dev-Ops User Stories. The assessment began on the 20th of August 2025 and concluded on the 24th of August 2025. The assessment was carried out by David Biayna Neal.

# Executive Summary

The purpose of using Miro Azure Cards is so engineers can view, create, and edit Azure DevOps work items directly in Miro boards with near-real-time two-way sync. The integration is supported for Miro Business and Enterprise plans and works with Azure DevOps Services (cloud). Setup requires org/team installation by a Miro Admin who is also the Azure Org Owner/Project Collection Admin, plus per-user OAuth authorization; the picker then surfaces only items each user can access. The app uses Azure service hooks with polling as a fallback and may require allow-listing specific Miro domains/IPs. Exposed data includes fields like Title, State, Assignee, Priority, Story Points, Iteration, Area, Acceptance Criteria (Description is read-only in Miro; custom fields aren’t supported). Primary risks are third-party OAuth access to Azure DevOps, board-level oversharing inside/outside the tenant, and network egress to Miro; with correct RBAC, SSO, board-sharing controls, and network allow-listing, this integration is suitable for go-live on 15 Sep 2025.

# Scope

Azure Cards integration end-to-end: Identity & access: Admin prerequisites, per-user OAuth, least-privilege in Azure DevOps and Miro; Data flows & exposure: What fields leave Azure, how they’re rendered/stored in Miro, board sharing/export risks; Network controls required domains/IPs and proxy/firewall behaviour; Eventing & sync webhooks/service hooks, polling behaviour, failure modes.

Link: <https://help.miro.com/hc/en-us/articles/360033799934-Azure-Cards>

# Background

Miro Azure Cards integrate Azure Boards work items into Miro so teams can run PI planning, backlog prioritization, story mapping, etc., while keeping Azure DevOps as the system of record. Installation happens at company/team level (by a Miro Admin) and then each user authorizes their Microsoft account and Azure org URL; after that, users can import items via URL or the in-app picker, create new Azure work items from Miro, or convert Miro cards/sticky notes into Azure items, with changes syncing both ways. The feature set and prerequisites are documented by Miro; notably, Azure DevOps Services (cloud) is supported, specific fields are synchronized (no custom fields; Description not editable in Miro), OAuth must be enabled in Azure, and Miro IPs/domains may need to be allow-listed in restricted environments.

**Adding Azure Cards for the organization or team**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Connecting personal Azure account to use Azure Cards**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Importing Azure work items to a Miro board**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Creating and editing Azure Cards directly in Miro**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Converting Miro cards or sticky notes to Azure Cards**

**A screenshot of a computer

AI-generated content may be incorrect.**

# Security Recommendations

* Enforce SSO with Microsoft Entra ID; disable password logins. Use SCIM for automatic provisioning/deprovisioning and group-based access. Require MFA for all users (including guests)
* Limit app availability to only the Miro teams that need it; keep Azure DevOps permissions least-privilege. The initial installer must be both a Miro Admin and an Azure Organization Owner/Project Collection Admin, document this requirement in change records.
* In Azure DevOps, explicitly enable “Third-party application access via OAuth,” and maintain a quarterly review to revoke stale authorizations.
* Enable Audit Logs and Content Logs in Miro.
* Educate teams on field behaviour: supported fields sync; Description is view-only in Miro; custom fields aren’t supported. Avoid placing sensitive data into non-governed fields.

# Limitations and Constraints

# Results of Assessment

No findings were raised as a result of this assessment.