SOP: Integration with IT Tools

1. Purpose

To establish a standardized procedure for integrating the Microsoft Admin Center with other IT management tools. This integration will centralize monitoring, automate tasks, and streamline overall IT operations for enhanced efficiency and security.

2. Scope

This SOP applies to all IT personnel responsible for managing the Microsoft 365 suite, security, and other third-party IT management and security platforms.

3. Strategic Benefits

Strategic Benefits:

- **Centralized Security Posture:** Unify email threat data from Proofpoint and endpoint compliance data from NinjaOne with Microsoft's security fabric, enabling faster, more correlated incident response.
- **Streamlined User Lifecycle Management:** Ensure user data is consistently synchronized between Microsoft 365, Proofpoint, and NinjaOne, reducing administrative errors.
- **Automated Threat Remediation:** Enable Proofpoint to automatically quarantine malicious emails and NinjaOne to enforce compliance policies on devices based on triggers from the Microsoft ecosystem.
- **Operational Efficiency:** Reduce manual effort by automating mail flow configuration, device monitoring, and user management across all three platforms.

4.0 Scope

- **In-Scope:** This document governs the technical procedure for establishing and maintaining the integration between Microsoft 365 (specifically Exchange Online, Azure Active Directory, and Microsoft Intune), Proofpoint (Email Protection), and NinjaOne.

- **Out-of-Scope:** This SOP is not a substitute for the full vendor documentation for each platform. It does not cover the initial procurement, licensing, or cost analysis of these tools.

5.0 Pre-requisites and Initial Assessment

- **Administrative Access:**
  - **Microsoft 365:** Global Administrator role is required for the initial setup.
  - **Proofpoint:** Full administrative access to the Proofpoint on Demand portal.
  - **NinjaOne:** Full administrative access to the NinjaOne portal.
- **Change Management Approval:**
  - Submit a formal change request detailing the business justification, risk assessment, implementation plan, and rollback strategy. This proposal **obtains formal approval directly from the Chief Technology Officer (CTO)** before implementing any new integration.

6.0 Detailed Integration Procedure: Proofpoint

The integration connects Proofpoint for email filtering and threat response directly into the Microsoft 365 mail ecosystem.

Phase 1: Planning & Design

1. **Goals:**
   a. Route all inbound and outbound company email through Proofpoint for analysis.
   b. Synchronize Microsoft 365 users and groups to Proofpoint.
   c. Allow Proofpoint's Threat Response Auto-Pull (TRAP) to remove malicious emails from user inboxes after delivery.
2. **Security Assessment:** The integration requires creating Azure AD App Registrations with specific Microsoft Graph API permissions. These permissions will be reviewed and approved by the Security Team.
3. **Mail Flow:** Document the current MX record and mail flow rules in Exchange Online. Plan the cutover time for changing the MX record to minimize disruption.

Phase 2: Implementation & Connection Steps

Step 1: Synchronize Users with Azure Active Directory This allows Proofpoint to have an up-to-date list of your users for policy enforcement.

1. In the Proofpoint Admin Portal, navigate to **User Management > Import/Sync > Directory Sync**.
2. Choose to add a new directory and select **Azure Directory**.
3. Proofpoint will provide you with the exact steps to create an App Registration in Azure AD (entra.microsoft.com). This typically involves:
    a. Creating a new App Registration named Proofpoint-Directory-Sync.
    b. Granting the following Microsoft Graph API permissions (Application type): User.Read.All, Group.Read.All, Directory.Read.All.
    c. Creating a client secret for the app, copying its value immediately, and storing it securely.
4. Return to Proofpoint and enter the **Application (client) ID**, **Directory (tenant) ID**, and the **Client Secret**.
5. Test the connection and configure the sync schedule (e.g., every 6 hours).

Step 2: Configure Inbound & Outbound Mail Flow This is the most critical step and directs your email through Proofpoint.

1. **Add Domains:** In Proofpoint, go to **System > Domains** and add the domains you want to protect (e.g., yourcompany.com).
2. **Configure Inbound Connectors in Exchange Online:**
    a. In the Exchange Admin Center (admin.exchange.microsoft.com), navigate to **Mail flow > Connectors**.
    b. Create a new connector:
        i. **Connection from:** Partner organization
        ii. **Connection to:** Office 365
    c. Name it Inbound from Proofpoint.
    d. Select "Use the sender's IP address" and add the IP ranges provided by Proofpoint in their documentation. This ensures Microsoft trusts email coming from Proofpoint.
3. **Update MX Records:**
    a. Log in to your public DNS provider.
    b. **Crucially, lower the TTL (Time to Live) of your current MX record to 5 minutes.** Wait for the old TTL to expire.
    c. Change the MX record to point to the value provided by Proofpoint.
4. **Configure Outbound Connector (Smart Host):**
    a. In the Exchange Admin Center, create another connector:
        i. **Connection from:** Office 365
        ii. **Connection to:** Partner organization
    b. Name it Outbound to Proofpoint.

c. Set it to apply only when a transport rule is met (you will create this next).
d. Route mail through the smart host provided by Proofpoint (e.g., outbound.pphosted.com).
e. Create a new **Mail flow > Rule**. Name it Route Outbound via Proofpoint. Apply it if the sender is 'Inside the organization' and set the action to 'Redirect messages to' the Outbound to Proofpoint connector.

Step 3: Enable Threat Response Auto-Pull (TRAP)

1. In Proofpoint, navigate to the **Threat Response** module.
2. Follow the setup wizard to connect to your Microsoft 365 environment. This will require creating a **new Azure AD App Registration** (Proofpoint-TRAP).
3. This app requires more powerful permissions. Grant the following Microsoft Graph API permissions (Application type):
    a. Mail.ReadWrite: To read and modify emails in any user's mailbox.
    b. User.Read.All: To find the correct user.
4. Once configured, you can create policies in Proofpoint to automatically "pull" emails containing newly identified threats from user inboxes.

Phase 3: Testing & Validation

- Send test emails from external accounts to your company email and verify they are logged in Proofpoint.
- Send a test email from a company account to an external account and verify it routes through Proofpoint.
- Use Proofpoint's tools to send a benign test threat and confirm TRAP successfully removes it from the test user's inbox.

Detailed Integration Procedure: NinjaOne

The integration connects NinjaOne with Microsoft 365 to centralize device management, monitor endpoints, and synchronize user/device information for comprehensive IT operations.

Step 1: Deploy NinjaOne Agent via Microsoft Intune (Win32 App)

- This method leverages Intune's robust Win32 app deployment capabilities to push the NinjaOne agent to your managed Windows workstations.

- **Prepare the NinjaOne Agent Installer:**
    - **Download Agent:** Log in to your NinjaOne portal, navigate to the **Administration** section, and download the **Windows Agent Installer** (usually an .exe or .msi file).
    - **Identify Silent Install Command:** Consult NinjaOne's official documentation for the exact silent installation command-line arguments. This typically includes:
        - A silent switch (e.g., /qn, /quiet, -s).
        - Your unique **Organization ID** or **Installation Token** (provided in your NinjaOne portal, often in the agent download area).
        - Example command for an MSI: msiexec /i "NinjaRMM_Agent.msi" /qn /norestart INSTALLTOKEN="YOUR_INSTALL_TOKEN" ORGID="YOUR_ORG_ID"
        - Example command for an EXE: NinjaRMMAgent.exe /s /q /install_token YOUR_INSTALL_TOKEN /org_id YOUR_ORG_ID
- **Add the Win32 App to Microsoft Intune:**
    - Sign in to the **Microsoft Intune admin center** at endpoint.microsoft.com.
    - Navigate to **Apps > All apps**.
    - Click **+ Add**.
    - Select **Windows app (Win32)** from the "App type" dropdown.
    - Click **Select**.
    - **App package file:**
        - Click "Select app package file".
        - Browse to and upload the .intunewin file you created.
        - Click **OK**.
    - **App information:**
        - Fill in required fields: **Name**, **Description**, **Publisher** (NinjaOne), **Category**.
        - Upload a logo if desired.
        - Click **Next**.
    - **Program:**
        - **Install command:** Enter the full silent installation command you identified (e.g., msiexec /i "NinjaRMM_Agent.msi" /qn /norestart INSTALLTOKEN="YOUR_INSTALL_TOKEN" ORGID="YOUR_ORG_ID").
        - **Uninstall command:** Provide the silent uninstall command (e.g., msiexec /x "{MSI_PRODUCT_CODE}" /qn - you'll find the MSI

Product Code in the detection rules if using MSI, or use NinjaOne's uninstall command).
        - **Install behavior:** Select System.
        - **Device restart behavior:** Select No specific action (as agent installs usually don't require reboots).
        - Click **Next**.
    - **Detection rules:** This tells Intune how to know if the agent is successfully installed.
        - **Rule format:** Select Manually configure detection rules.
        - **Add rule:**
            - **Rule type:** If you're deploying an MSI, choose MSI. Enter the **MSI product code** (you can find this by running msiexec /i "NinjaRMMAgent.msi" or inspecting the MSI with a tool like Orca, or from NinjaOne's documentation).
            - If using an EXE or a custom install, choose File or Registry and specify a unique file path/name or registry key that indicates successful installation.
        - Click **OK**, then **Next**.
    - **Dependencies:** Leave blank unless your NinjaOne agent requires other software to be installed first.
    - **Supersedence:** Leave blank for initial deployment.
    - **Assignments:**
        - Under **Required**, click "+ Add group".
        - Select the **Azure AD device group(s)** containing the workstations where you want to deploy the NinjaOne agent. Avoid "All Devices" unless absolutely necessary.
        - Click **Select**, then **Next**.
    - **Review + Create:** Review all your settings and click **Create**.

Phase 2: Testing & Validation

- **Intune Deployment Monitoring:** In the Intune admin center, navigate to **Apps > Windows apps > NinjaOne Agent** and monitor the **Device install status** and **User install status** tabs for successful deployments.
- **NinjaOne Console Verification:**
    - After devices receive the agent, log in to your NinjaOne portal.
    - Verify that the newly deployed devices appear in your NinjaOne console under the correct organizations/sites.
    - Confirm that detailed device information (OS version, hardware, running processes, etc.) is populating correctly from the agent.

    - Test a simple remote action from NinjaOne to a newly deployed device to confirm communication.

7.0 Appendices

- **Appendix A:** Link to Proofpoint IP ranges for firewall and connector configuration Connection Details - Proofpoint, Inc.
- **Appendix B:** Link to Microsoft Graph API permissions documentation Microsoft Graph permissions reference - Microsoft Graph | Microsoft Learn