SOP - Company wide Software + Procedures

Purpose

To define standardized procedures for the deployment, usage, and management of software across the organization, ensuring compliance, security, and operational efficiency.

Scope

This SOP applies to all departments, employees, contractors, and systems within the organization that interact with or rely on software applications and services.

1. Approved Software List

A list of company-approved software categorized by function:

- **Technologies**
  - **Microsoft 365:** Email, File storage, Collaboration
- **Core Operation**
  - **CCH:** Tax Planning & Returns
  - **QuickBooks:** Accounting and Bookkeeping
  - **AssureSign:** Document authorization
- **Management and Tracking**
  - **Monday.com:** Project & Time management
  - **Ignition:** Appointment scheduling
  - **Karbon HQ:** Practice Management System
  - **CPA Charge:** Client payment processing system
- **Communication:**
  - **Zoom:** Virtual meeting & webinar
  - **Loom:** Video explanations
  - **Calendly:** Appointment scheduling
  - **Teams:** Communication & Calls
- **Marketing and Design**
  - **Mailchimp:** campaigns and communication
  - **Stripo:** Email design and templates
  - **Dripify:** Outreach and lead generation
  - **Canva:** Marketing material design
  - **Adobe Suite:** Marketing material design
  - **Descript:** Video Editing
  - **Capcut:** Video Editing
  - **SEMRUSH:** SEO Tool

- **Development and Design**
  - **GitHub:** Code Repository & Versioning
  - **WebFlow:** Web Page development
  - **Figma:** UI/UX Design
  - **Jira:** Agile Software Development
- **IT & Security Tools**
  - **NinjaOne:** Endpoint & patch management
  - **ProofPoint:** Email protection and prevention
  - **LastPass:** Password vault
  - **HikVision:** Surveillance and Security
- **Support & Automation**
  - **Trainual:** Onboarding and SOP training
  - **Zapier:** Workflow automation
- **Vulnerability Scanning:** Nessus, OpenVAS

2. Software Request & Approval Process

- All new software requests must be submitted via the **IT Service Request Form**.
- Approval is required from:
  - **IT Security Team**
  - **CTO**

3. Definitions and Abbreviations

- **NinjaOne:** The Remote Monitoring and Management (RMM) platform used for endpoint management and software deployment.
- **Endpoint:** Any device managed by NinjaOne (e.g., workstation, server, laptop).
- **Software Package:** The installer files and any associated configuration files required for a software installation.
- **Silent Installation:** An installation process that runs without requiring user interaction or displaying a user interface.
- **Scripted Installation:** A software deployment method where a script (e.g., PowerShell for Windows, Bash for Linux/macOS) automates the silent installation process.
- **RMM:** Remote Monitoring and Management.
- **MSI:** Microsoft Installer package format for Windows applications.
- **EXE:** Executable file format for Windows applications.
- **DMG:** Disk Image file format for macOS.
- **PKG:** Package installer format for macOS.
- **Exit Code:** A numerical value returned by a program or script upon completion, indicating success (typically 0) or failure (non-zero).

4. Software Deployment via NinjaOne

The process for pushing software in NinjaOne primarily involves preparing the software for silent installation and then configuring a script or policy within NinjaOne to execute that installation.

4.1. Software Preparation and Packaging

Objective: To ensure the software installer is ready for silent deployment and can be executed without user interaction.

1. **Obtain Software Installer:** Download the official, stable, and approved installer file(s) for the desired software. Prioritize silent installer formats (e.g., MSI for Windows, PKG for macOS) if available.
2. **Identify Silent Installation Switches:**
    a. **For MSI files (Windows):** Typically use /quiet or /qn for silent installation. Other common switches include /norestart to prevent reboots, and ALLUSERS=1 for per-machine install.
        i. **Example:** msiexec /i "YourSoftwareInstaller.msi" /qn /norestart
    b. **For EXE files (Windows):** Often require specific vendor-provided switches (e.g., /S, /silent, /quiet, /qn). Check the vendor's documentation or use resources like silentinstall.org or appdeploy.com.
        i. **Example:** YourSoftwareSetup.exe /silent /norestart
    c. **For macOS (.pkg or .dmg):** PKG files can often be installed silently using installer -pkg <path_to_pkg> -target /. DMG files may require mounting and then running a contained PKG or app installer.
        i. **Example (PKG):** sudo installer -pkg /path/to/YourSoftware.pkg -target /
    d. **For Linux:** Often involves package managers (apt, yum, dnf) or custom scripts provided by the vendor.
3. **Test Silent Installation (Crucial Step):**
    a. Perform a test installation on a *non-production test machine* that mimics your target environment.
    b. Execute the silent installation command manually.
    c. Verify that the software is installed correctly, silently, and that all necessary components are present.
    d. Check the return/exit code of the installation command. An exit code of 0 generally indicates success.

4. **Gather Additional Files:** If the installation requires additional configuration files, license keys, or supporting scripts, package them together.
5. **Prepare for Upload:**
    a. For single large files (e.g., MSI, EXE, PKG), you can typically upload them directly to NinjaOne's software library.
    b. For multiple files or complex installations, it's often best to create a .zip archive containing the installer(s) and your custom installation script.

4.2. Creating a Software Deployment Policy/Script in NinjaOne

Objective: To configure NinjaOne to deliver and execute your prepared software package. NinjaOne typically uses "Policies" that can execute "Scripts" for software deployment.

1. **Log in to NinjaOne Dashboard:** Access your NinjaOne tenant.
2. **Navigate to the Scripting Section:**
    a. Go to **"Configuration"** (left sidebar).
    b. Select **"Scripting"**.
    c. Click on **"Add Script"**.
3. **Configure Script Details:**
    a. **Name:** Provide a descriptive name for your deployment script (e.g., "Install Adobe Reader DC 2025," "Deploy Chrome Browser").
    b. **Description:** Explain what the script does (e.g., "Downloads and silently installs Adobe Reader DC using MSI. Requires reboot.").
    c. **Operating System:** Select the target OS (Windows, macOS, or Linux).
    d. **Script Language:**
        i. **Windows:** Choose PowerShell or Batch. PowerShell is generally preferred for its robustness.
        ii. **macOS/Linux:** Choose Bash.
    e. **Maximum execution time:** Set a reasonable timeout (e.g., 30-60 minutes).
    f. **Execute as:** Select **"System"** (or "Root" for Linux/macOS) to ensure it has elevated privileges required for installation.
4. **Upload Software Files:**
    a. If your script relies on a local installer file or a .zip archive, you'll upload it here.

b. In the script editor, look for an option to "Add File" or "Upload File." Upload your prepared installer or .zip package. NinjaOne will host this file and make it available to your script on the target device.
c. **Note:** If using a .zip file, your script will need to unzip it first on the target device.

5. **Write the Installation Script:** This is the core of your deployment.

General Script Logic:

a. Download the installer file (if not uploaded directly and NinjaOne hosts it) OR ensure the uploaded file is accessible.
b. Define paths to the installer.
c. Execute the silent installation command.
d. Include error checking (check exit codes).
e. Optionally, include logging to a local file on the endpoint for troubleshooting.

6. **Save Script:** Once your script is ready, click **"Save"**.

4.3. Assigning and Deploying the Software

Objective: To schedule the execution of your software deployment script on target devices.

1. **Navigate to Policies:**
   a. Go to **"Configuration"** (left sidebar).
   b. Select **"Policies"**.
   c. Choose the specific policy applied to the devices or device groups you want to target (e.g., "Workstation Policy," "Server Policy"). Or, create a new temporary policy for this deployment.
2. **Add a Scheduled Script:**
   a. In the policy settings, go to the **"Scheduled Scripts"** section.
   b. Click **"Add Scheduled Script"**.
   c. **Script:** Select the deployment script.
   d. **Schedule:**
      i. **Run Now:** For immediate deployment.
      ii. **Once:** To run the script once at a specified time/date.
      iii. **Recurring:** For software that needs to be installed on new devices joining the policy or to ensure compliance.

        iv. **Run on Device Startup/Check-in:** If you want it to run every time the device checks in or starts up (ensure script is idempotent!).
    e. **Time (if scheduled):** Set the desired execution time. Consider off-peak hours to minimize disruption.
    f. **Conditions:** (Optional) Add conditions such as "Operating System," "Device Type," "Online Status," etc., to refine targeting.
    g. **Maximum execution time:** Can be overridden from script settings if needed.
    h. **Execution Rate (for recurring scripts):** How often to retry if it fails or check for compliance.
    i. Click **"Add"**.
3. **Save Policy:**
    a. After adding the scheduled script, ensure you click **"Save"** for the policy itself to apply the changes to the associated devices.

4.4. Monitoring Deployment Status

Objective: To track the progress and outcome of the software deployment.

1. **Check Policy Status:**
    a. In NinjaOne, navigate to **"Configuration" > "Policies"**.
    b. Select the policy you modified.
    c. Go to the **"Devices"** tab to see which devices are associated with the policy.
    d. For each device, you can see if the script execution was successful.
2. **Review Script Results:**
    a. Navigate to **"Reporting"** (left sidebar).
    b. Go to **"Activity Log"** or **"Script Results"**.
    c. Filter by the script name and date range.
    d. This will show the execution status (Success, Failed, Pending) for each device the script ran on.
    e. Click on individual script results to view the standard output (stdout) and standard error (stderr) captured by NinjaOne, which can be invaluable for troubleshooting.
3. **Check Device-Specific Logs (On the Endpoint):**
    a. For failed deployments, remotely connect to the affected device (via NinjaOne's remote tools).
    b. Check the local log file created by your script

c. Check standard system event logs (Event Viewer for Windows, journalctl for Linux) for installation-related errors.

5. Updates and Patch Management

- All company-approved software must be updated according to defined patching schedules.
- Security-critical updates are pushed automatically via centralized management tools.
- Employees should not delay or bypass updates unless approved by IT.

6. Security Requirements

- All software must be scanned and reviewed for vulnerabilities before deployment.
- Endpoint protection and monitoring must remain enabled on all devices.
- Any software with known security risks must be removed immediately.

7. Decommissioning and Uninstallation

- When software is no longer required:
  - Submit a **decommissioning request** to IT
  - IT will ensure license recycling and proper removal
- Data generated by the software must be archived.

8. User Responsibilities

- Do not install unapproved software.
- Report any software issues or security alerts to the IT Helpdesk immediately.
- Adhere to the Acceptable Use Policy (AUP).

9. Exceptions

- Exceptions must be documented, approved by IT Security, and reviewed quarterly.

10. Review and Maintenance

- This SOP is reviewed **annually** or upon major changes to IT infrastructure.
- Maintained by: **IT Security Team**

Date: 06/13/2025