|  |  |
| --- | --- |
| **Topic:** | Last Logged on lookup script |

|  |  |
| --- | --- |
| **Description:** | Script that outputs the last users to log on to a intune enrolled dvice |

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Changes Made** | **By (User)** |
| 1.0 | 11/09/2024 | Initial document creation | Mathew Gallehawk |

# Requirements

* Requires Powershell graph. (will set up when script is run)
* Powershell in Admin mode.
* Admin account with authorisation.
* Service desk PIM permissions are active.

# Process

1. Open Powershell / Termianl as administrator.
2. Navigate to this directory using cd command:  
   cd "C:\Users\xxxxxx\G.James Australia Pty Ltd\Team ICT Service Desk - General\tools\Last Loged on User lookup"
3. Run the script using .\DeviceLookup.ps1
4. You may get some security prompts, select open if there is a popup or R if prompted in terminal.
5. The first time you run this you may have to install graph api which involves selecting Y repeatedly.
6. A windows login popup will appear, and authentication will be required for execution.

|  |  |
| --- | --- |
| 1. A prompt for the host name of a device will appear, once the modules are checked and installed if required. |  |
| Note if you receive red text like this check that service desk permissions are active. |  |
| 1. Enter the host name of a device and select enter.   The results of the device shall be displayed if the device is found.  If a device does not exist the fields will be blank. |  |
| If you have a second device to look up select y,  IMPORTANT: if you want to quit select n, this will close you cloud session and not leave it open. Best practice. | |

# Script Review

**Overview**

This PowerShell script is designed to look up details for an Intune-managed device. It checks for required modules, installs them if necessary, and retrieves device information and the last logged-on user details.

**Author**

Mathew Gallehawk

**Functions**

1. **intro**
   * **Description**: Displays an introductory message about the script.
   * **Usage**: Automatically called at the start of the script.
2. **module-check**
   * **Description**: Checks if the required modules (Microsoft.Graph.Beta.Devicemanagement and Microsoft.Graph.Beta.Users) are installed.
   * **Returns**: Two variables indicating the presence of the modules.
3. **module-install**
   * **Description**: Installs the specified module based on user choice.
   * **Parameters**:
     + [$choice](vscode-file://vscode-app/c:/Users/e103719/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): A string indicating which module to install ([1](vscode-file://vscode-app/c:/Users/e103719/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) for Microsoft.Graph.Beta.Devicemanagement, [2](vscode-file://vscode-app/c:/Users/e103719/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) for Microsoft.Graph.Beta.Users).
4. **moduleManage**
   * **Description**: Manages the installation of required modules by checking their presence and prompting the user to install them if they are missing.
   * **Usage**: Automatically called during the script execution.
5. **captureDevice**
   * **Description**: Prompts the user to enter the hostname of the device they want to look up.
   * **Returns**: The device hostname entered by the user.
6. **lastLoggedOn**
   * **Description**: Retrieves the last logged-on user details for the specified device.
   * **Parameters**:
     + [$device](vscode-file://vscode-app/c:/Users/e103719/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): The hostname of the device.
   * **Returns**: A list of users who have logged on to the device, including their user ID, display name, and last logon date/time.
7. **pullDeviceDetails**
   * **Description**: Retrieves detailed information about the specified device.
   * **Parameters**:
     + [$device](vscode-file://vscode-app/c:/Users/e103719/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): The hostname of the device.
   * **Returns**: An object containing various properties of the device, such as device name, ID, type, manufacturer, model, serial number, OS version, etc.
8. **main**
   * **Description**: The main function that orchestrates the script's workflow. It calls other functions to display the intro, manage modules, connect to Microsoft Graph, capture device details, and display the information.
   * **Usage**: Automatically called when the script is run.

**Workflow**

1. **Introduction**: The script starts by displaying an introductory message.
2. **Module Management**: It checks if the required modules are installed and prompts the user to install them if they are not.
3. **Module Import**: The script imports the necessary modules.
4. **Microsoft Graph Connection**: It connects to Microsoft Graph.
5. **Device Lookup Loop**: The script enters a loop where it:
   * Prompts the user to enter a device hostname.
   * Retrieves and displays the last logged-on user details.
   * Retrieves and displays detailed device information.
   * Asks the user if they want to check another device.
6. **Disconnection**: The script disconnects from Microsoft Graph when the user chooses to exit the loop.

**Usage**

1. **Run the Script**: Execute the script in a PowerShell environment with administrative privileges.
2. **Follow Prompts**: Respond to prompts to install required modules if necessary.
3. **Enter Device Hostname**: Input the hostname of the device you want to look up.
4. **Review Information**: The script will display the last logged-on user details and detailed device information.
5. **Repeat or Exit**: Choose to look up another device or exit the script.