|  |  |
| --- | --- |
| **Topic:** | AD-UC Auditor |

|  |  |
| --- | --- |
| **Description:** | PowerShell application for service desk |

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Changes Made** | **By (User)** |
| 1.0 | 23/09/24 | Initial document creation | Mathew Gallehawk |

# Use case:

This script audits an on prem active directory service for users who have not been terminated properly. Such users may have access to licenses or dl groups that unnecessarily consume resources. Additionally, this script highlights users who have not logged in in an allotted time frame that may have been missed in a user termination process.

# Requirements:

* Script execution policy set to unrestricted
* Admin account with access to the Active directory server.
* This script relies on specific filters and search criteria for gjames. You may need to adjust these depending on your Active Directory structure and data.

# Process:

* Open PowerShell as admin.
* Navigate to and run script.

# Overview

## Script Explanation:

This PowerShell script automates several queries and creates a report in your OneDrive drive. It pulls data from Active Directory and writes the results to CSV files.

\*\*Overall process:\*\*

1. \*\*Checks report folder:\*\* The script checks if the designated report folder exists and creates it if not.

2. \*\*Loops through licenses:\*\* It iterates through a list of licenses and groups.

3. \*\*Identifies disabled members:\*\* For each group, it retrieves members and identifies those that are disabled.

4. \*\*Outputs disabled users:\*\* If any disabled members are found, their names are written to a CSV file in the "Licenses" folder, along with the group name.

5. \*\*Identifies enabled users in disabled OU:\*\* It fetches users from the "End-dated Users" OU and identifies those that are enabled.

6. \*\*Outputs enabled users:\*\* These users are written to a CSV file in the "Other" folder with the filename "EnabledUsersInDisabledOU".

7. \*\*Identifies inactive users:\*\* It retrieves users who haven't logged in for 365 days and writes them to a CSV file in the "Other" folder with the filename "GhostUsers".

8. \*\*Identifies disabled users in DL groups:\*\* It iterates through DL groups and identifies users that are disabled within those groups.

9. \*\*Outputs disabled DL group members:\*\* Names of disabled members in DL groups are written to a CSV file in the "DLGroups" folder, along with the group name.

10. \*\*Identifies un-migrated users:\*\* It retrieves disabled users that haven't been moved to the "End-dated Users" OU and writes them to a CSV file in the "Other" folder with the filename "UsersNotMovedToEndDatedOU".

11. \*\*Identifies un-hidden users:\*\* It retrieves disabled users that haven't been hidden from GAL and writes them to a CSV file in the "Other" folder with the filename "UsersNotHiddenFromGal".

\*\*Key functions:\*\*

\* `WriteOut`: Writes data to a CSV file.

\* `ReportFolder`: Creates the report folder structure.

\* `GetMembers`: Retrieves members of a group.

\* `GetEndDatedUsers`: Retrieves users from a specific OU.

\* `UserStatus`: Identifies users based on their enabled/disabled status.

\* `GetGhostUsers`: Retrieves users who haven't logged in for a specified number of days.

\*\*Output:\*\*

The script will create multiple CSV files in the report folder, each containing the identified users and relevant information.