==

CATEGORY: LocalAccounts

==

Local account operations directly affect system access, privilege boundaries, and security posture. These SOPs ensure every local account action performed through RDAM Script Wizard is controlled, auditable, and aligned with enterprise security and compliance standards.

SOP 1 – Get Local User Details

Script Name: Get Local User Details **Category:** LocalAccounts **Version:** 1.0 **Approved By:** IT Operations / Security

1. Purpose

This script retrieves detailed information about a local user account on a Windows system. It supports troubleshooting, access validation, compliance checks, and forensic analysis.

2. Scope

- **Systems:** Windows servers and workstations
- **Accounts:** Local (non-domain) user accounts
- **Authorized Personnel:**
    - System administrators
    - Security analysts
    - Helpdesk Tier-2/3

3. Definitions

- **Local User:** An account stored in the local SAM database.
- **SID:** Security Identifier unique to the account.

4. Preconditions

- Operator must have administrative rights on the target system.

- System must be reachable.
- Account must exist.

5. Required Inputs

- Local username

6. Procedure Steps

1. **Input Collection**
    - Wizard prompts for username.
    - Validate non-empty.
2. **Account Resolution**
    - Query local SAM for the account.
    - If not found, abort and log.
3. **Attribute Retrieval**
    - Retrieve:
        - SID
        - Account enabled/disabled state
        - Password last set
        - Last logon
        - Group memberships
        - Account description
4. **Output Formatting**
    - Present structured details.
5. **Logging**
    - Log username, operator, timestamp.

7. Expected Output

- Detailed metadata for the local user.

8. Post-Execution Validation

- Operator may verify via `lusrmgr.msc`.

9. Error Handling

- Access denied
- User not found
- SAM corruption (rare)

10. Security Considerations

- Local accounts may have elevated privileges; handle data carefully.
- Avoid exposing sensitive account metadata.

11. Audit Logging Requirements

- Operator ID
- Username
- Timestamp

12. Organizational Benefit Statement

This script provides a consistent, auditable method for retrieving local account details, supporting troubleshooting, compliance, and security reviews.

SOP 2 – Create Local User

Script Name: Create Local User **Category:** LocalAccounts

1. Purpose

This script creates a new local user account on a Windows system, supporting onboarding, break-glass access, and system-specific service accounts.

2. Scope

- Windows servers and workstations
- Local SAM database
- Used by system administrators and security teams

3. Definitions

- **Local User Creation:** Adding a new identity to the local SAM.

4. Preconditions

- Operator must have administrative rights.

- Username must follow naming conventions.
- Password must meet complexity requirements.
- Account creation must be authorized.

5. Required Inputs

- Username
- Password
- Optional: Description
- Optional: Require password change at next logon

6. Procedure Steps

1. **Input Collection**
    - Wizard prompts for username, password, and options.
2. **Validation**
    - Check username uniqueness.
    - Validate password complexity.
3. **Account Creation**
    - Create user in local SAM.
    - Apply password and flags.
4. **Post-Creation Verification**
    - Confirm account exists.
    - Validate attributes.
5. **Logging**
    - Log username, operator, timestamp.

7. Expected Output

- Confirmation of successful account creation.

8. Post-Execution Validation

- Operator may verify via `lusrmgr.msc`.

9. Error Handling

- Username already exists

- Password complexity failure
- Access denied

10. Security Considerations

- Local accounts should be minimized; prefer domain accounts.
- Passwords must not be logged.
- Break-glass accounts must follow strict policy.

11. Audit Logging Requirements

- Operator ID
- Username
- Timestamp

12. Organizational Benefit Statement

This script ensures local account creation is performed safely and consistently, reducing misconfiguration and supporting secure system access.

SOP 3 – Add Local User to Group

Script Name: Add Local User to Group **Category:** LocalAccounts

1. Purpose

This script adds a local user to a local group, granting additional privileges or access rights.

2. Scope

- Local groups on Windows systems
- Used by system administrators and security teams

3. Definitions

- **Local Group:** A group stored in the local SAM.
- **Membership Assignment:** Adding a user to a group.

4. Preconditions

- Operator must have administrative rights.
- User and group must exist.
- Action must align with approved access request.

5. Required Inputs

- Username
- Group name

6. Procedure Steps

1. **Input Collection**
    - Wizard prompts for username and group.
2. **Resolution**
    - Confirm user exists.
    - Confirm group exists.
3. **Membership Check**
    - If user already in group, return informational message.
4. **Add Operation**
    - Add user to group.
5. **Post-Change Verification**
    - Confirm membership.
6. **Logging**
    - Log user, group, operator, timestamp.

7. Expected Output

- Confirmation of successful membership addition.

8. Post-Execution Validation

- Operator may verify via `lusrmgr.msc`.

9. Error Handling

- User not found
- Group not found
- Access denied

10. Security Considerations

- Adding users to privileged groups (e.g., Administrators) requires strict approval.

11. Audit Logging Requirements

- Operator ID
- Username
- Group
- Timestamp

12. Organizational Benefit Statement

This script ensures privilege assignments are performed safely and with full accountability, supporting least-privilege principles.

SOP 4 – Remove Local User from Group

Script Name: Remove Local User from Group **Category:** LocalAccounts

1. Purpose

This script removes a local user from a local group, supporting de-provisioning and privilege reduction.

2. Scope

- Local groups on Windows systems
- Used by system administrators and security teams

3. Definitions

- **De-provisioning:** Removing access when no longer required.

4. Preconditions

- Operator must have administrative rights.
- User and group must exist.
- Action must align with approved request.

5. Required Inputs

- Username
- Group name

6. Procedure Steps

1. **Input Collection**

    - Wizard prompts for username and group.
2. **Resolution**
    - Confirm user exists.
    - Confirm group exists.
3. **Membership Check**
    - If user not in group, return informational message.
4. **Remove Operation**
    - Remove user from group.
5. **Post-Change Verification**
    - Confirm removal.
6. **Logging**
    - Log user, group, operator, timestamp.

7. Expected Output

- Confirmation of successful removal.

8. Post-Execution Validation

- Operator may verify via `lusrmgr.msc`.

9. Error Handling

- User not found
- Group not found
- Access denied

10. Security Considerations

- Removing users from critical groups may impact system functionality; ensure approvals.

11. Audit Logging Requirements

- Operator ID
- Username
- Group
- Timestamp

12. Organizational Benefit Statement

This script ensures privilege removal is performed safely and consistently, reducing risk and supporting least-privilege enforcement.