**Solution Design for Managing Windows Features and Local Users**

**Status:**

**Reviewed By:**

**Link of Jira Tickets for this Doc:**

**Overview**

The solution will consist of a configuration file and a main script that automates the installation of Windows Features and the creation of local users on a server. It will also incorporate secure handling of user secrets.

**Detailed Design**

**1. Configuration File**

**Purpose**: Centralize configuration settings for Windows Features, local user accounts, and logging.

**Format**: PowerShell script.

**File Name**: config.ps1

**Contents**:

* **Log File Path**: Path to the log file for recording script activities.
* **Windows Features**: List of Windows Features to be installed.
* **Local Users**: Array of local users with usernames and passwords.
* **Auto Restart**: Boolean indicating whether the server should automatically restart after operations.
* **Error Messages**: Custom messages for error handling.

**Example Configuration (config.ps1)**:

powershell

Copy code

# Define the log file path

$Global:LogFilePath = "C:\Path\To\Your\LogFile.log"

# Define Windows Features to install

$Global:WindowsFeatures = @(

"NET-Framework-Features",

"Web-Server"

)

# Define local users to create, with their passwords

$Global:LocalUsers = @(

@{

Username = "User1"

Password = "P@ssw0rd1"

},

@{

Username = "User2"

Password = "P@ssw0rd2"

}

)

# Optional settings

$Global:AutoRestart = $true

# Define error handling messages

$Global:ErrorMessages = @{

FeatureInstallFailed = "Feature installation failed. Please check the system and address any issues before proceeding."

UserCreationFailed = "User creation failed. Please check the system and address any issues."

}

**2. Main Script**

**Purpose**: Automate the installation of Windows Features and the creation of local user accounts using the configuration file.

**File Name**: main.ps1

**Responsibilities**:

* Import configuration from config.ps1.
* Install Windows Features based on the configuration.
* Create local users using provided credentials.
* Log the outcome of each operation.

**Execution Flow**:

1. **Import Configuration**: Load settings from config.ps1.
2. **Install Windows Features**: Iterate over the list of features and perform installation.
3. **Create Local Users**: Iterate over the list of users and create accounts.
4. **Log Results**: Record success and failure messages to the log file.

**Key Functions**:

* Log-Message: Logs messages to the specified log file and writes to console.
* Install-WindowsFeatures: Installs features listed in the configuration.
* Create-LocalUsers: Creates local users with specified usernames and passwords.

**3. Secret Management**

**Purpose**: Securely handle user passwords and other sensitive data.

**Approaches**:

* **Encrypted Configuration Files**: Encrypt sensitive parts of the configuration file.
* **Windows Credential Manager**: Store and retrieve credentials securely.
* **Secret Management Services**: Utilize services like Azure Key Vault or AWS Secrets Manager for managing secrets.

**Considerations**:

* **Encryption**: Ensure configuration files containing sensitive data are encrypted.
* **Access Control**: Restrict access to configuration files and secret storage.
* **Backup**: Implement a strategy for securely backing up configuration files and logs.

**Goals**

1. **Automation**: Automate the installation and user creation processes to reduce manual intervention.
2. **Centralized Configuration**: Use a single configuration file for managing features and users to simplify updates.
3. **Security**: Implement secure methods for handling sensitive data to prevent unauthorized access.
4. **Logging**: Maintain detailed logs for tracking the success and failure of operations.
5. **Scalability**: Ensure the solution is adaptable to accommodate additional features, users, or settings.