Secure Multi-Server Active Directory Synchronization

Version 1.2

This project provides a robust framework for synchronizing Active Directory (AD) users, security groups, and identity attributes between disconnected or air-gapped environments. It leverages **OpenBao** as a centralized security vault to provide "Zero-Knowledge" transport of sensitive credentials.

1. System Architecture

The system follows a **Source-to-Target** synchronization pattern designed for high-security environments where servers cannot communicate directly over a network.

Primary Operational Modes

- **Export (Source Server):** Scans a designated Target OU, extracts user metadata, generates secure passwords (if missing), and encrypts all sensitive data into a portable payload.
- **Import (Target Server):** Restores the cryptographic environment, verifies the integrity of the payload, decrypts user credentials, and reconciles the local Active Directory to match the source state.

2. Component Directory

File	Role	Description
Initialize-SyncServer.ps1	**Infrastructure**	Prepares the Windows environment: creates directories, sets NTFS permissions, installs the OpenBao Windows Service, and configures local firewalls.
Invoke-BaoAutomation.ps1	**Security**	Automates the Vault lifecycle: initializes the master keys, unseals the vault, enables engines, and securely ingests/deletes

		temporary AD admin credentials.
Sync-AD-Transport.ps1	**Engine (v4.3)**	The primary logic driver. Handles AD attribute mapping, group membership reconciliation, and the asymmetric encryption/decryption workflow.
Receive-ADSyncPayload.ps1	**Transport (New)**	**Automated Retrieval:** Securely "pulls" the export package from a remote source via SFTP and automatically triggers the Sync Engine upon successful receipt.
Update-VaultPassword.ps1	**Utility (New)**	**Admin Override:** Interactive tool for administrators to manually rotate user passwords in the Vault and generate local log receipts.

3. Cryptographic Security Deep-Dive

The synchronization engine implements a high-entropy security model to ensure that sensitive data remains encrypted even if the physical transport media (USB drive, etc.) is intercepted.

Asymmetric Implementation (RSA-4096)

The system uses the OpenBao **Transit Engine** configured for rsa-4096.

- **The Public Key:** Is used by the **Source Server** to encrypt user passwords. Public keys can only encrypt; they cannot be used to reverse the process.
- **The Private Key:** Is stored strictly within the **Target Server's** OpenBao instance. This key is required for decryption.

Scenario: What if the Export files are stolen?

If an attacker intercepts the C:\ADSync\Export folder, they face three massive cryptographic barriers:

1. **The RSA Barrier:** The AD_State_Export.json contains "Ciphertext". Without the RSA Private Key, this data is mathematically impossible to decrypt.
2. **The Transit Key Encryption:** The transport-key.backup file is an encrypted "blob". To restore this key, an attacker would need the **exact same Master Keys** used by the original vault.
3. **The Signature Barrier:** The AD_State_Export.hmac file ensures integrity. Any attempt to modify the user data in the JSON file will result in an HMAC mismatch, causing the Import script to immediately reject the payload.

4. Administrative Utilities

Manual Password Updates (Update-VaultPassword.ps1)

In scenarios where a specific user's password must be changed outside of the automated sync cycle (e.g., emergency rotation), administrators can use this utility.

- **Functionality:** Prompts for a UserID and a masked password, updates the KV-V2 store in OpenBao, and writes a reference log to C:\ADSync\Users\.
- **Audit Trail:** Every manual update is logged to the ADSync Windows Event Log (Event ID 2000), capturing the administrator's identity.

5. Automated Transport (SFTP Pull)

The Receive-ADSyncPayload.ps1 utility automates the transport phase using a pull-based methodology:

- **Remote Retrieval:** Connects to the Source server via SFTP (Port 22) using the WinSCP .NET Assembly to fetch new payloads.
- **Auto-Execution:** Once the three required components (JSON, HMAC, and Key Backup) are pulled into the local Import folder, the script automatically launches the Sync-AD-Transport.ps1 engine.
- **Re-entrancy Protection:** Uses lock files to ensure multiple transfer tasks do not overlap.
- **Logging:** Pull results and engine trigger status are reported to SFTP_Pull.log and the Windows Event Log (Event ID 3001).

6. Credential Provisioning (ad_creds_temp.json)

To perform AD operations, the system requires a Domain Admin or delegated Service Account.

Security Lifecycle

1. **Creation:** Admin creates the file on Source and Target.
2. **Ingestion:** Invoke-BaoAutomation.ps1 moves credentials into the Vault's internal KV-V2 storage.

3. **Cleanup:** The script **permanently deletes** the JSON file once ingested.

7. Account Access Requirements

Local Execution Context (Task Scheduler)

The account running the Scheduled Task requires:

- **Log on as a batch job** rights.
- **Full Control** over C:\ADSync.
- **Local Administrator Privileges**.

AD Synchronization Context (Vaulted Credentials)

The account stored inside OpenBao requires the following delegated permissions on the **Target OU**:

- **Create/Delete User Objects**
- **Write All Properties** (for attribute sync)
- **Reset Password**
- **Read/Write Group Membership**

8. Automation via Scheduled Task

1. **Task Name:** AD-Sync-Pull-and-Import.
2. **Security:** "Run whether user is logged on or not" + "Run with highest privileges".
3. **Action:**
    - **Program:** powershell.exe
    - **Arguments:** -ExecutionPolicy Bypass -File "C:\ADSync\Receive-ADSyncPayload.ps1"

9. Setup & Operation Workflow

Phase 1: Preparation (Both Servers)

1. Copy scripts and bao.exe to C:\ADSync.
2. Run Initialize-SyncServer.ps1 as Administrator.

Phase 2: Credential Provisioning

1. Create ad_creds_temp.json.
2. Run Invoke-BaoAutomation.ps1 to unseal and ingest.

Phase 3: Daily Operation

1. **Source:** Run Sync-AD-Transport.ps1 (generates Export files in the Export folder).
2. **Target:** The Scheduled Task runs Receive-ADSyncPayload.ps1, which pulls the files from Source and automatically kicks off the local Import process.
3. **Manual Backup:** In the event of network failure, manually move files from Source Export

to Target Import.

10. AD Hardening & Safety

- **CannotChangePassword:** Set to True to maintain sync parity.
- **PasswordNeverExpires:** Set to True to prevent lockout.
- **Group Reconciliation (v4.2):** Implements a HashSet keep-list to ensure groups created in the same session are not flagged as "stale" and deleted.