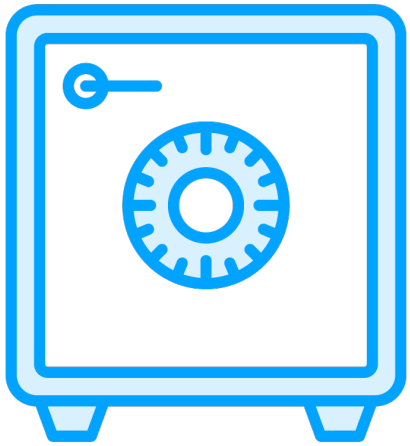
Design and Implement a Strategy for Managing Sensitive Information in Automation

John Savill
Chief Architect

@ntfaqguy | onboardtoazure.com

Azure Key Vault Review

Azure Key Vault



- Provide secure containers, called vaults
- Utilize HSMs designed to stop any tampering
- Secured using Entra-integrated RBAC
- Secrets can be securely stored and retrieved
- Keys can be securely stored or generated, then used for cryptographic operations
- Secrets are commonly used for password type data
- Each item can have its own RBAC

Using Azure Key Vault from a Pipeline

You must have authenticated to Azure

The authenticated identity must have been granted access to read the secret

Read the secret within the pipeline

Use the value in other tasks/steps

Using Secret Capabilities of ADO and GitHub

Azure DevOps Secret Features

- ADO has secret variables
- These are encrypted
- They can be used safely in pipelines
- A variable group can be shared across pipelines
- Variable groups can map to secrets in Azure Key Vault

GitHub Secrets Features

GitHub supports secrets

These can be stored at organization, repository, and repository environment levels

Secret must be added as usable in the workflow file

Secrets are redacted from workflow logs

Handling Sensitive Files during Deployment

Sensitive Files

During deployment you may have to handle sensitive files

Certificates, SSH keys, and more

Care needs to be taken in their handling

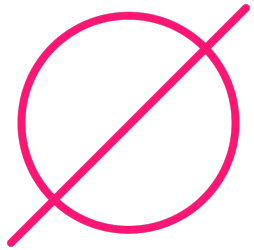
Azure DevOps has a secure files feature in the library

Up to 10MB files

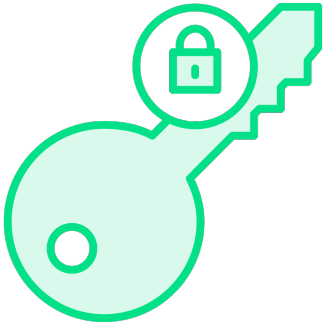
Security can be defined for the library and per file

Avoiding Leakage of Sensitive Information

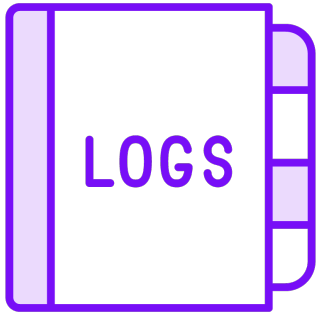
Rules of Secrets



Don’t have them!

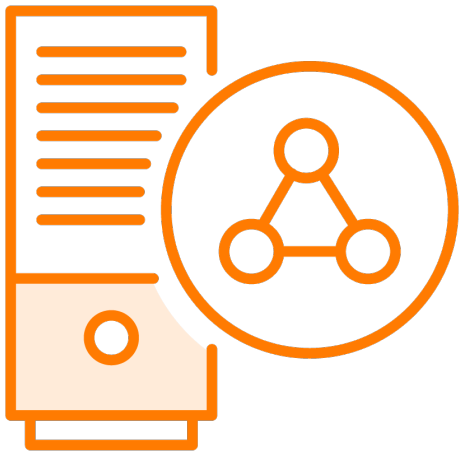


Never store unencrypted

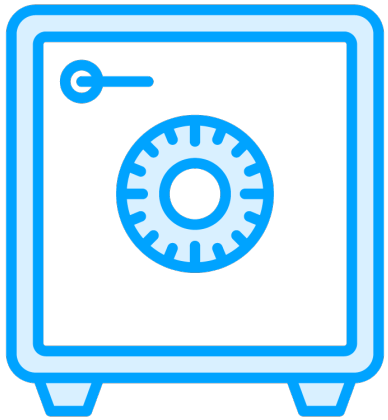


Avoid them being written to logs

Best Practices



Use federation for identity



Store secrets you must have in Key Vault

Take time to understand *where* sensitive data is and ensure it is labeled and protected accordingly.