Module 6 - Secrets Management

AUBURN UNIVERSITY

CPSC 4970 Applied Cyber Security

What is a Secret?

AUBURN

- Definition of “Secret
  - Non-human privileged credentials
  - Private piece of information used to unlock protected resources or sensitive information
  - Location can be in applications, databases, servers, cloud-native environments, build pipelines.
- Examples
  - User or auto-generated passwords
  - API and other application keys/credentials (including within containers)
  - SSH Keys
  - Database and other system-to-system passwords.
  - Private certificates
    - Secure communication, transmitting and receiving of data (TLS, SSL etc.)
    - Code or file signing
  - Private encryption keys for systems like PGP
  - RSA and other one-time password devices
  - API Keys

Industry Challenges Drivers Secrets Mgmt

AUBURN

- Visibility
  - Proliferation of secrets among admins, developers, and other team members who all manage their secrets separately, if they're managed at all.
  - Security risk with all different people and methods who handle secrets
- Hardcoded/embedded credentials
  - Privileged passwords and other secrets are needed to facilitate authentication for communications and access to resources (database, applications)
  - Applications are shipped and installed with hardcoded, default credentials, which are easy to crack by hackers using scanning tools and applying simple guessing or dictionary-style attacks.
  - DevOps tools frequently have secrets hardcoded in scripts or files, which jeopardizes security
- Privileged credentials and the cloud
  - Cloud and SaaS administrator access (as with AWS, Office 365, etc.) provide broad superuser privileges.
- DevOps tools
- While secrets need to be managed across the entire IT ecosystem, DevOps

Industry Challenges Drivers Secrets Mgmt

AUBURN

- DevOps tools
  - DevOps teams require access to orchestration, configuration management, and other tools a(Chef, Puppet, Docker containers, etc.) relying on automation and other scripts that require secrets to work.
  - Secrets should all be managed according to best security practices
    - Credential rotation
    - Limited time/activityaccess
    - Auditing trail
- Manual secrets management processes
  - Risk increases when people manually manage secrets
  - Weak secrets
    - Lack of password rotation
    - Default passwords
    - Embedded secrets
    - Password sharing
    - Easy-to-remember passwords
  - Manual secrets management processes equate to a higher likelihood of security gaps and bad practice
-

Secrets Management Best Practices

AUBURN

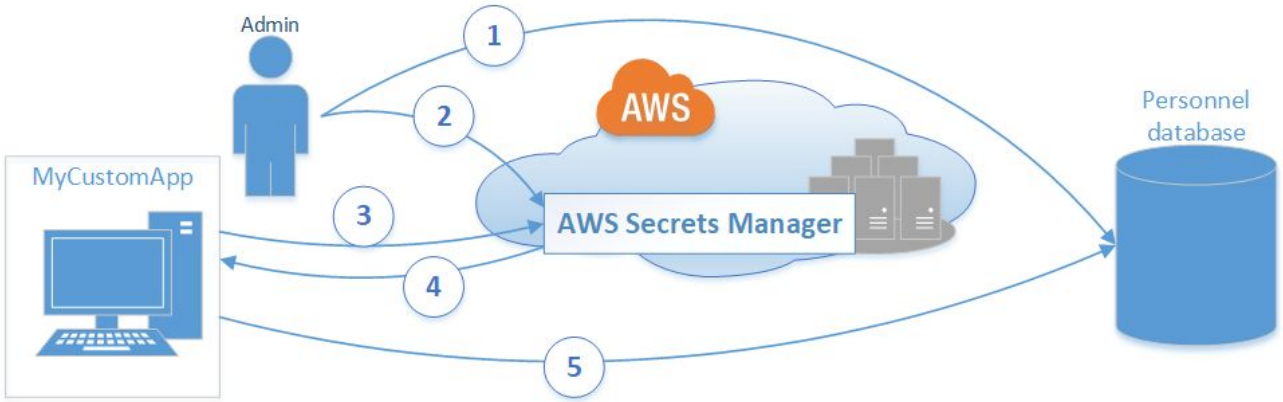
- Centralized Management.
  - Bring all secrets under a Key Management System (KMS)
  - Audit all source code and IT infrastructure for secrets
- Eliminate hardcoded/embedded secrets
  - During development -
    - DevOps tool configurations, build scripts, code files, test builds, production builds, applications.
  - During Production -
    - API calls, system access, cloud environments
- Enforce Password Rules
  - Complexity - length, uniqueness, no words
  - Expiration - from minutes to months
  - Rotation - change on regular interval as people leave organizations.
  - Temporary passwords - one time usage
  - Change once shared.

Secrets Management Best Practices

AUBURN

- Privileged session monitoring
  - Log and audit access
  - SEIM tools can alert on suspicious activities based on logs
  - Recording to capture keystrokes and screens
  - Tools can trigger locks if detects suspicious activity in-progress
- 3rd Parties
  - Temporary employees/contractors, parters,
  - Important they conform to best practices in using and managing secrets.
- Threat analytics
  - Detection of anomalies and potential threats.
  - The more integrated and centralized your secrets management, the better you will be able to report on accounts, keys applications, containers, and systems exposed to risk.
- Secure Development Lifecycle
  - Treat development tools as sensitive systems, control access and secrets
  - Use security testing so that code does not contain embedded secrets

AWS Secret Management Tool



1. Administrator sets up credentials to a database
2. Enters credentials in AWS Secrets Manager
3. Application requiring use of database asks for credentials from AWS Secrets Mgr
4. Credentials are return to application
5. Application uses credentials to access database

- Policies can be set on Secrets Manager to trigger rotation, expiration as well as keep an audit log and history of credentials.

AUBURN UNIVERSITY