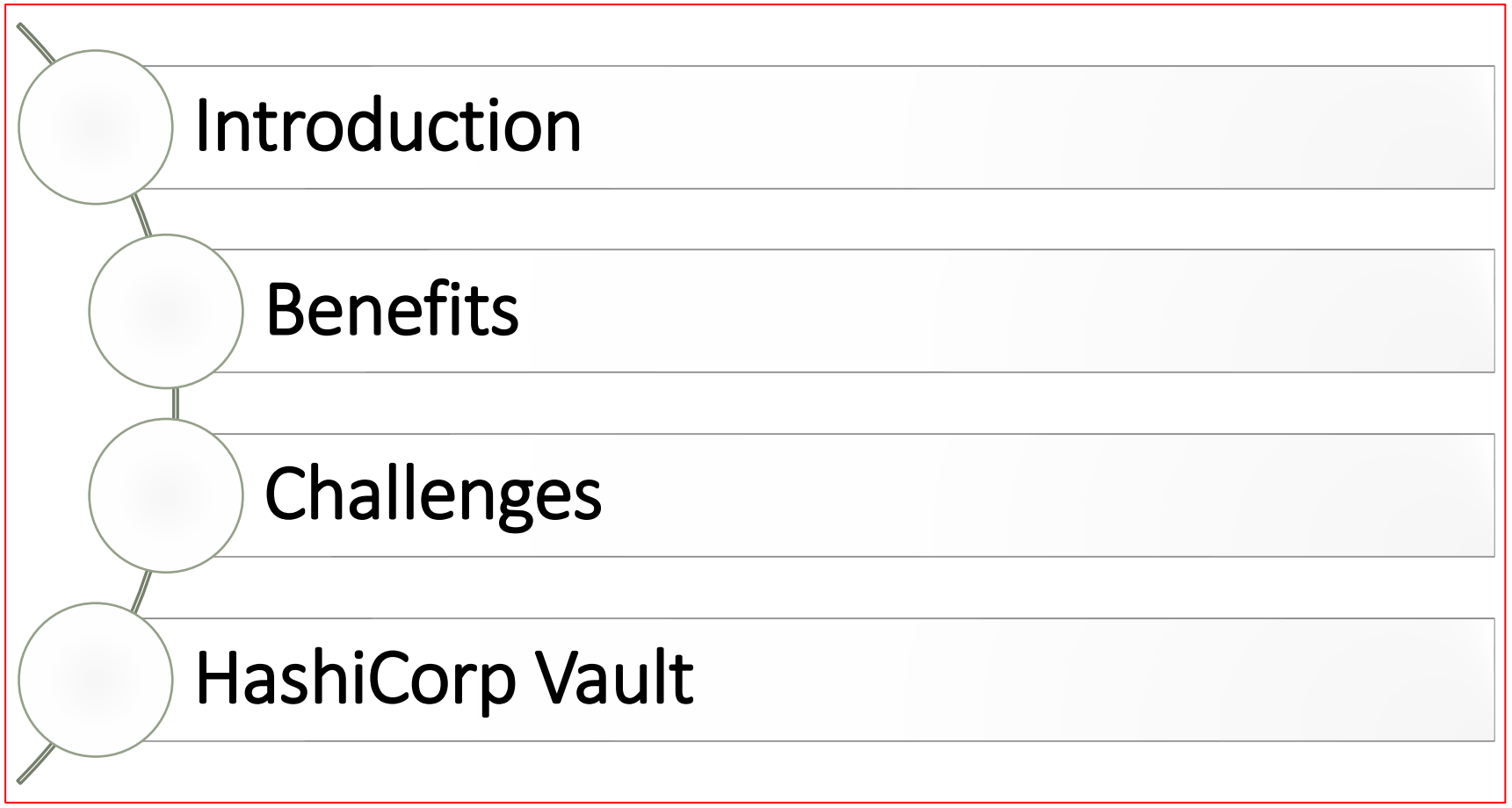
iPMAC
NET YOUR WORK

DEVSECOPS COURSE

SECRET MANAGEMENT

TRAINER: TRAN HUU HOA

AGENDA



INTRODUCTION

Secrets management is a crucial practice for securely handling sensitive information within applications and IT environments. Common types of secrets include:

- Privileged account credentials
- Passwords
- Certificates
- SSH keys
- API keys
- Encryption keys

BENEFITS

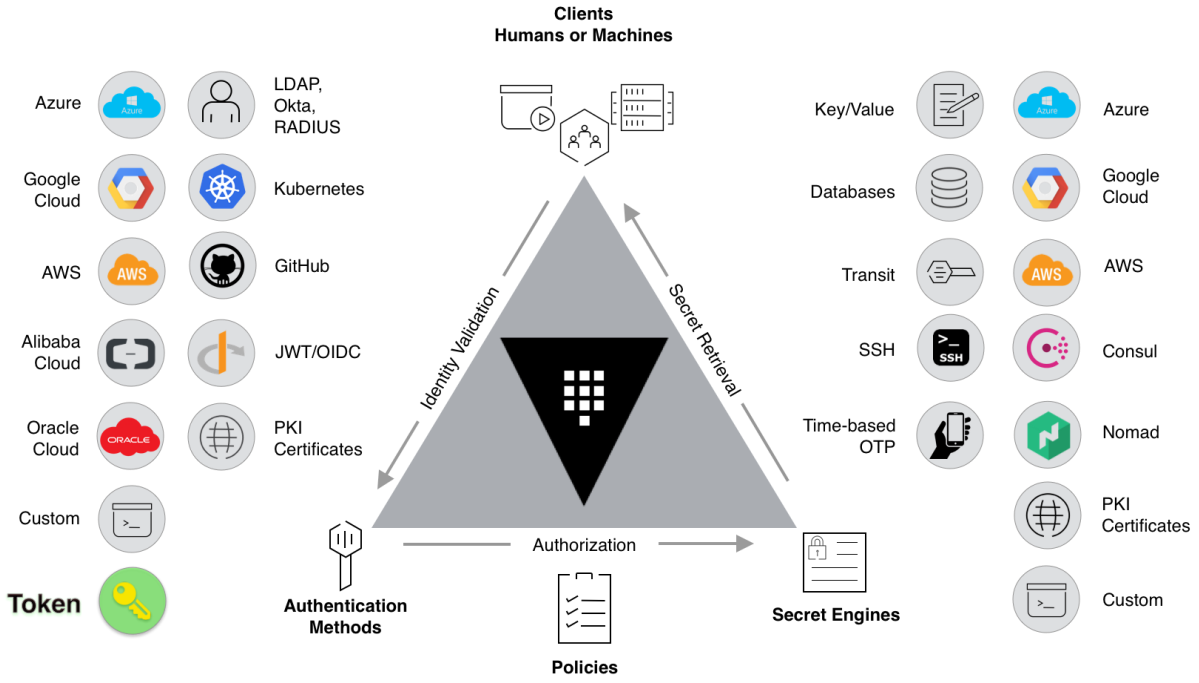
- Consistently enforce security policies for non-human identities.
- Authenticate all access requests using non-human credentials.
- Protect against cyber-attacks and mitigate risks.

CHALLENGES

- Non-human identities (such as automated processes) rely on secrets to access resources.
- Cyber attackers target secrets to gain unauthorized access.
- Secrets are widespread across various tools, applications, and environments.

HASHICORP VAULT

HashiCorp Vault is an identity-based secrets and encryption management system. It allows you to securely manage sensitive data, such as API encryption keys, passwords, and certificates



HASHICORP VAULT

Key features:

- Secrets Management: Vault lets you centrally store, access, and distribute secrets programmatically.
- Certificates: Generate, rotate, and revoke certificates on demand.
- Keys: Distribute, rotate, enable, and disable keys.
- Data Protection: Protect data both in transit and at rest.
- Reduce Risk: Prevent credential exposure, eliminate secret sprawl, and block unauthorized users.
- Developer Efficiency: Automate secret creation, consumption, expiration, and rotation using a single API.
- Operational Efficiency: Scale secrets access across large IT environments, consolidate applications for secrets storage, and automate credential rotation

HASHICORP VAULT

Authentication methods:

- Token-Based Authentication: Users or applications authenticate using tokens issued by Vault.
- Username/Password: Vault supports traditional username/password authentication.
- AppRole: This method is suitable for applications. An AppRole ID and secret are used for authentication.
- GitHub, LDAP, or OIDC: Vault integrates with external identity providers for user authentication.
- AWS IAM: If your application runs on AWS, you can authenticate using IAM roles.
- TLS Certificates: Clients can authenticate using X.509 certificates.

HASHICORP VAULT

Best practices:

- Least Privilege: Follow the principle of least privilege when defining policies.
- Secret Rotation: Regularly rotate secrets (e.g., database credentials, API keys). Vault provides automated rotation mechanisms. Set up appropriate TTLs (time-to-live) for secrets.
- Dynamic Secrets: Use dynamic secrets whenever possible. Generate secrets on-demand for databases, cloud services, and other systems. This reduces exposure and enhances security.
- Audit Logging: Enable Vault's audit logging to track who accessed secrets and when. This helps with compliance and troubleshooting.
- High Availability: Deploy Vault in a highly available (HA) configuration. Use multiple nodes across different availability zones or data centers.
- Backup and Disaster Recovery: Regularly back up Vault data and store backups securely. Test disaster recovery procedures.
- Secure Communication: Ensure secure communication between Vault clients and servers using TLS certificates.
- Monitor Metrics: Monitor Vault's performance, resource usage, and health.
- Automate Deployment: Use infrastructure as code (IaC) tools to automate Vault deployment and configuration.
- Test and Validate: Regularly test Vault's functionality, including secret retrieval, rotation, and authentication.