Dynamic Secrets: Unleashing the Thor's Hammer ⚒️ of FOSS Security

Sunday, 25 August 2024 12:41 PM

Real Life Example

1)Login Authentication : Bank Website

2)AWS account verification 2MFA

3)Transaction Confirmation : Credit card / Debit card

4)Email Verification

5)Microsoft Authenticator – Company Portal

****

Prerequisites :(covered in PPT)

1. Awareness on AWS Bucket, AWS IAM – User, Assumed Roles, IAM Policy.
2. Hashi Corp Vault
3. Client Server Architecture
4. Secret Engine

****

INSTALLATION

Vault installation: Install I Vault I HashiCorp Developer
Extract the downloaded zip into your respective folder (ex: C:\Users\snasha\Desktop\projects)
1) Start vault in developer mode : vault server -dev
copy values VAULT_ADDR and Root Token
2) open another command prompt window :
Now keep this cmd open this would act as your Vault Server operating in Dev Mode
Open new CMD window

- set VAULT_ADDR=http://127.0.0.1:8200
- set VAULT_TOKEN= returned value when you started the vault in dev mode

Now you are all set to use this command window to talk to the other vault server engine
3) Secrets list : shows all the enabled secrets within vault server and their mount paths
vault secrets list

****

Performing Basic CRUD operations in HashiCorp Vault
CREATE
vault kv put -mount=secret myAwesomeApp/creds dbid=admin dbpid=Password@124
READ
vault kv get -mount=secret myAwesomeApp/creds
UPDATE
vault kv patch -mount=secret myAwesomeApp/creds version=2 dbid=admin dbpid=Password@901
GET the updated pwds -> version 2
vault kv get -mount=secret -version=1 myAwesomeApp/creds
DELETE
vault kv delete -mount=secret myAwesomeApp/creds
PRO TIP
Vault kv put -mount=secret MyNewApp token=-
This would allow you to type the pwd without displaying it and once you are done type ^Z to exit

****

Getting started with AWS Secret Engine

Install GITBASH : https://git-scm.com/downloads

1) Bucket name
bucket1 : my-foss-bucket1
2) IAM USER
my-foss-user1
3) Policy name: Policy-my-foss-user1
4) Credentials
Bucket1 -> Static Creds

AWS_ACCESS_KEY_ID=**********
Access Key: Secret key : **********

In GitBash >
export AWS_ACCESS_KEY_ID=AKIAURI3MYJWPOEQJQYT
export
AWS_SECRET_ACCESS_KEY=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
export AWS_DEFAULT_REGION=us-east-1

aws s3 ls s3://my-foss-bucket1
aws sts get-caller-identity
aws s3 ls

These credentials are static secrets and you can see the arn from the get caller identity
-> I have also listed the bucket contents
****

ENABLING AWS SECRET ENGINE
vault secrets enable aws
vault secrets list
****

Configure AWS Secrets Engine
>vault write aws/config/root access_key=AKIAURI3MYJWPOEQJQYT
secret_key=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 region=us-east-1
> vault read aws/config/root
Configure Vault Role
vim policy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-foss-bucket1/*",
        "arn:aws:s3:::my-foss-bucket1"
      ]
    }
  ]
}
```

vault write aws/roles/foss-role-Sakshi credential_type=iam_user
policy_document=@policy.json
foss-role-Sakshi is the unique role name

vault read aws/roles/foss-role-Sakshi

****

READING AND GENERATING DYNAMIC CREDENTIALS
vault read aws/creds/foss-role-Sakshi
The accesskey and secret key can be used to perform any s3 operation(specified within the role permissions)

Operations using Dynamic Creds
In GitBash >

Got from Hashicorp Vault
access_key AKIA**********W4
secret_key ZSn**********LTE9ZK

Static creds from Aws ->

Bucket1 -> Static Creds
AWS_ACCESS_KEY_ID=AKI**********QYT

Access Key: Secret key : sguP**********2j4N9

Use hashicorp once
export AWS_ACCESS_KEY_ID=AKI**********W4
export AWS_SECRET_ACCESS_KEY=ZSn2**********LTE9ZK
export AWS_DEFAULT_REGION=us-east-1
aws s3 ls s3://my-foss-bucket1
aws sts get-caller-identity
aws s3 ls

The credentials in the above picture are static secrets and you can see the arn from the get caller identity -> I have also listed the bucket contents
Now I would use the above generated dynamic Credentials and list bucket contents I am using git bash for the aws cli commands Set the parameters

export AWS_ACCESS_KEY_ID=<replace with vault your-access-key>
export AWS_SECRET_ACCESS_KEY=<replace with vault your-secret-key>
Check ARN from the get caller identity: aws sts get-caller-identity
"This is a prove that I am not using permanent static creds"
****

Perform opertions:
aws s3 cp "filepath" s3://my-foss-bucket1/ -> Allowed
aws s3 ls -> not allowed
aws s3 ls s3://my-foss-bucket1 -> allowed
The above operations are based on the permissions you mentioned in the policy.json
To recollect Look at the policy_document:
vault read aws/roles/foss-role-yourname
To list all of your buckets, you must have the s3:ListAllMyBuckets permission.
****

Lease: renewal, revocation and inspection
vault read aws/creds/foss-role-Sakshi
Copy the full path of lease_id : This is used for renewal, revocation and inspection
LEASE LOOKUP
vault lease lookup aws/creds/foss-role-Sakshi/S9*****94xd

767h52m45s is almost 32 days
REVOKING
vault lease revoke aws/creds/foss-role-Sakshi/S9*****94xd
vault lease lookup aws/creds/foss-role-Sakshi/S9*****94xd
Once you have revoked it will take time to actually delete or inactive the keys

****

Time To Live TTL (Expiry) Of Token
Want to keep token only for 20 mins ?
Default is 768hr Tokens generated for services or applications usually have a default lease_duration of 768 hours (32 days) if not otherwise specified in Vault's configuration.
TTL is only available for assumed_role, federation_token, and session_token credential types

>vault write aws/roles/assumed-role-sakshi credential_type=assumed_role policy_document=@policy.json default_sts_ttl=20m num_uses=3 max_sts_ttl=20m role_arns=arn:aws:iam::311979459180:role/assumed-role-Sakshi1

assumed-role-sakshi is unique role name in hashicorp

arn of assumed role= from aws = arn:aws:iam::311979459180:role/assumed-role-Sakshi1
example : arn:aws: iam:: 311979459180:role/asssumed-rolel-foss
Change policy of the bucket to generate and use assume role

Read role
>vault read aws/roles/assumed-role-sakshi

Read the Dynamic Credentials : vault read aws/creds/assumed-role-sakshi

vault read aws/creds/foss-role-yourname
Few moments later : vault lease lookup aws/creds/paste your lease_id -> expired secrets

In GITBASH -> Export the session token also along with accessKey and secretKey If you don't export session token then listing the bucket contents won't work

****

Steps to disable AWS Secret Engine in HashiCorp Vault

vault secrets disable aws
verify -> vault secrets list

STop server ctrl c

Don't forget to delete AWS bucket, Role, user, and policy
****

More to explore

Auto Rotation policy : https://developer.hashicorp.com/hcp/docs/vault-secrets/auto-rotation

Extension to Databases : https://developer.hashicorp.com/vault/tutorials/db-credentials/database-secrets

****

Feel free to ping in case of doubts : (3) Sakshi Nasha ☁️ | LinkedIn

> **"You got the incredible power of FOSS through the Thor's Hammer to protect your the Asgard (Applications) from potential threats "**

Thank you