ATTACK DEFENSE

by PentesterAcademy

Name	Vault: OTP Based SSH Access
URL	https://www.attackdefense.com/challengedetails?cid=2360
Type	DevSecOps Basics: Secrets Management

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

Hashicorp Vault allows the user to securely store the secrets (e.g. tokens, passwords, certificates, encryption keys). The user or applications can interact with it using web UI, CLI, or HTTP API.

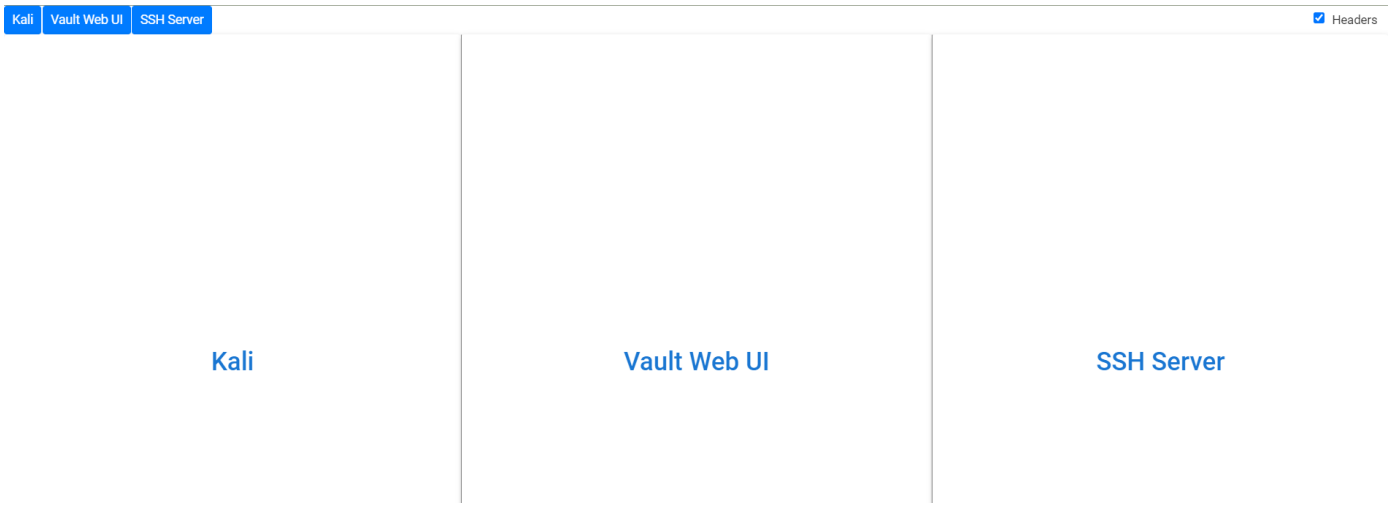
In this lab, a Vault server, vault CLI client, and an SSH server configured with vault-based OTP authentication are provided.

The Vault server is using the token: welcome123 (will be required to interact with the Vault server)

Objective: Follow the manual to learn how to use vault to generate and use credentials to securely login into a remote server using SSH!

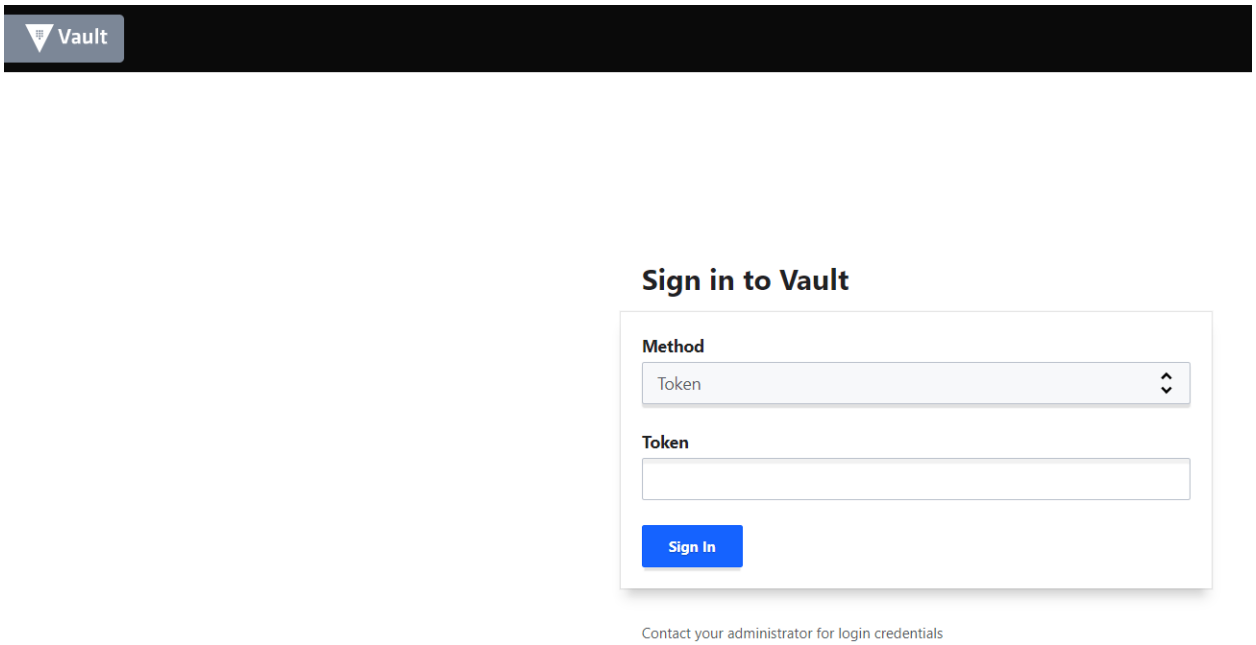
Lab Setup

On starting the lab, the following interface will be accessible to the user.



On choosing (clicking the text in the center) left left panel, **Kali GUI** will open in a new tab

On selecting the center panel, a web UI of **Vault** will open in a new tab.



On selecting the right panel, a CLI to **SSH server** will open in a new tab.

Solution

The SSH server is configured to use OTP token (from Vault server) for authentication.

We will generate the token using the Vault server and then use that to login into the SSH server.

Step 1: On Kali machine, export the following environment variables.

Commands:
export VAULT_ADDR=http://vault:8200
export VAULT_TOKEN=welcome123

```
root@kali:~# export VAULT_TOKEN=welcome123
root@kali:~#
root@kali:~# export VAULT_ADDR=http://vault:8200
root@kali:~#
```

The mapping for vault, sshserver is present in /etc/hosts file.

Command: cat /etc/hosts

```
root@kali:~# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.156.80.3    kali
127.0.0.1 AttackDefense-Kali
192.156.80.2 HelloWorld
192.156.80.3 kali
192.156.80.4 vault
192.156.80.5 sshserver
```

Step 2: Check the status of the Vault server by using the vault command.

Commands: vault status

```
root@kali:~# vault status
Key             Value
---             -----
Seal Type       shamir
Initialized     true
Sealed          false
Total Shares    1
Threshold       1
Version         1.7.2
Storage Type    inmem
Cluster Name    vault-cluster-f2d4e2a9
Cluster ID      a591e521-f5a9-2390-de4a-dbe9f9b12bd0
HA Enabled      false
```

Step 3: Enable the ssh Secret engine backend

Commands: vault secrets enable ssh

```
root@kali:~# vault secrets enable ssh
Success! Enabled the ssh secrets engine at: ssh/
root@kali:~#
```

Step 4: Check the IP address of the Kali machine's eth0 interface.

Command: ip addr

```
root@kali:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
236: eth0@if237: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:9c:50:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.156.80.3/24 brd 192.156.80.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Note down this IP address.

Use this IP address range while creating a vault role for user “root”

Command: vault write ssh/roles/otp_role key_type=otp default_user=root cidr_list=192.156.0.0/16

```
root@kali:~# vault write ssh/roles/otp_role key_type=otp default_user=root cidr_list=192.156.0.0/16

Success! Data written to: ssh/roles/otp_role
root@kali:~#
```

Step 5: The SSH server machine is on 192.156.80.5. The same can be verified by running “ping sshserver” command.

Obtain an OTP to login into SSH server (i.e. sshserver)

Command: vault write ssh/creds/otp_role ip=192.156.80.5

```
root@kali:~# vault write ssh/creds/otp_role ip=192.156.80.5
Key                 Value
---                 -----
lease_id            ssh/creds/otp_role/eMLYgQsB8ob5jCEohDBYu2Q9
lease_duration      768h
lease_renewable     false
ip                  192.156.80.5
key                 8e2ed5df-21b0-deb3-8a2c-f77e5d68b70b
key_type            otp
port                22
username            root
```

Step 6: Use this to login into the SSH Server.

Command: ssh root@sshserver

Provide the copied token when prompted for password.

```
root@kali:~# ssh root@sshserver
The authenticity of host 'sshserver (192.156.80.5)' can't be established.
ECDSA key fingerprint is SHA256:cFV99PGfdcbUNsA4xXM8qU75E0iR6DZLpkiVLsiaWr0
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'sshserver,192.156.80.5' (ECDSA) to the list of
Password:

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@sshserver:~#
```

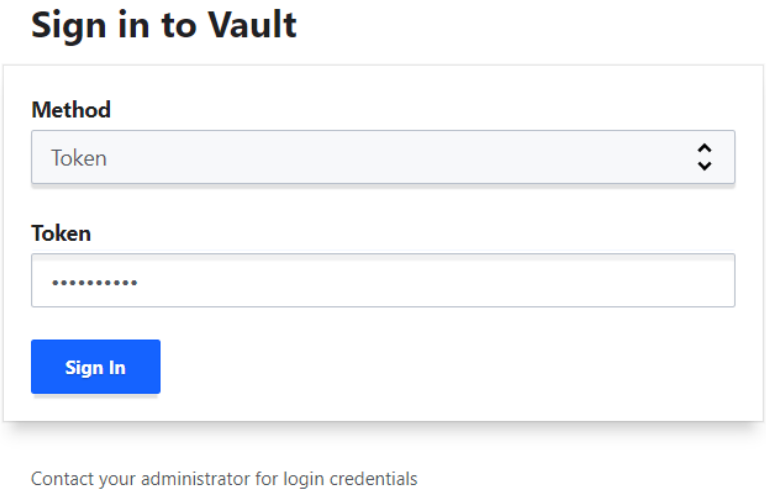
Please remember that this is an OTP and won't work again. One has to generate the token again in order to login into the SSH server.

Alternate Approach

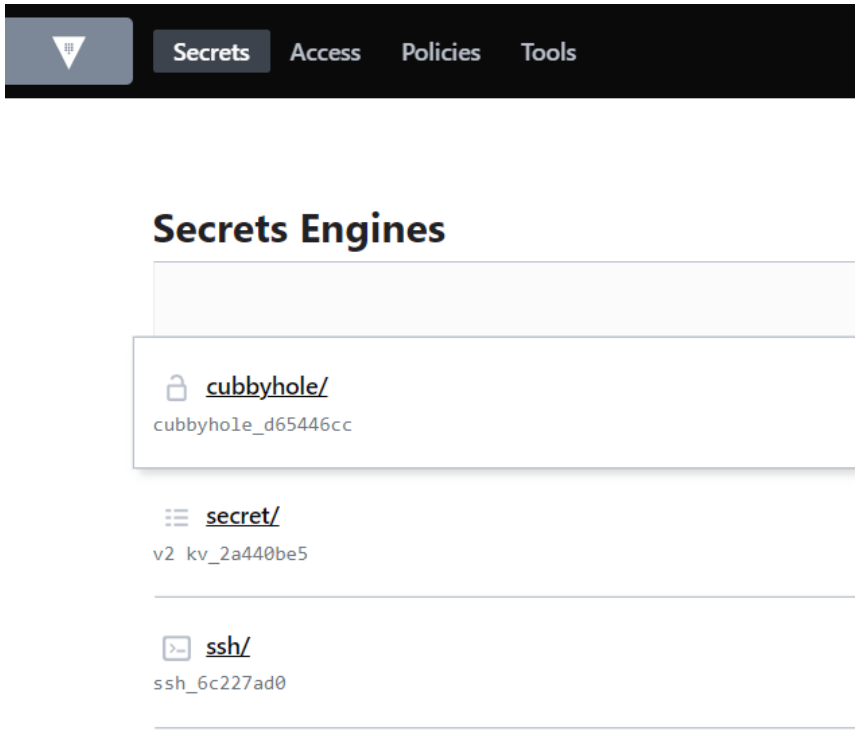
The user can also use the Vault web UI to generate a token and use it to perform SSH into the SSH server.

Step 1: Sign into Vault server web UI.

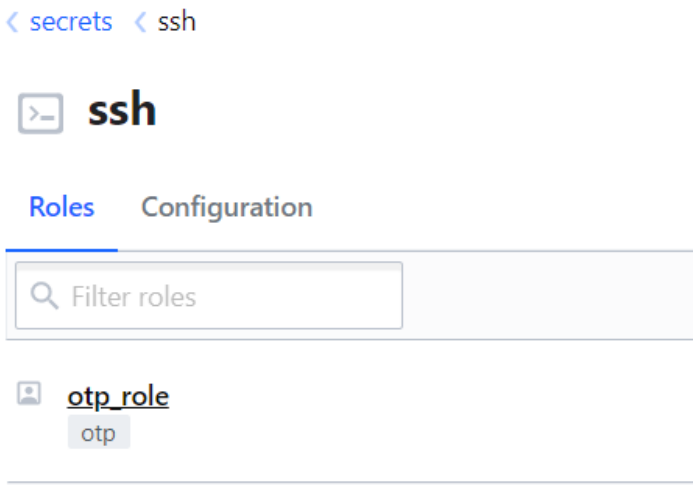
User “Token” method and token is welcome123



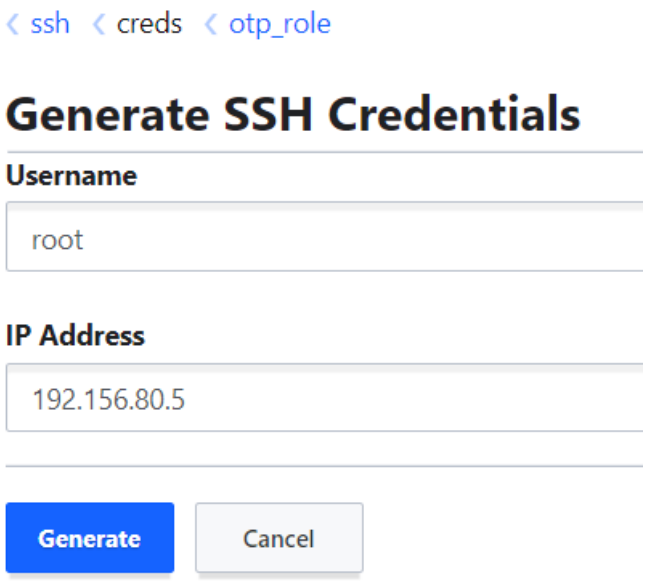
The backend Secret Engines will be visible now



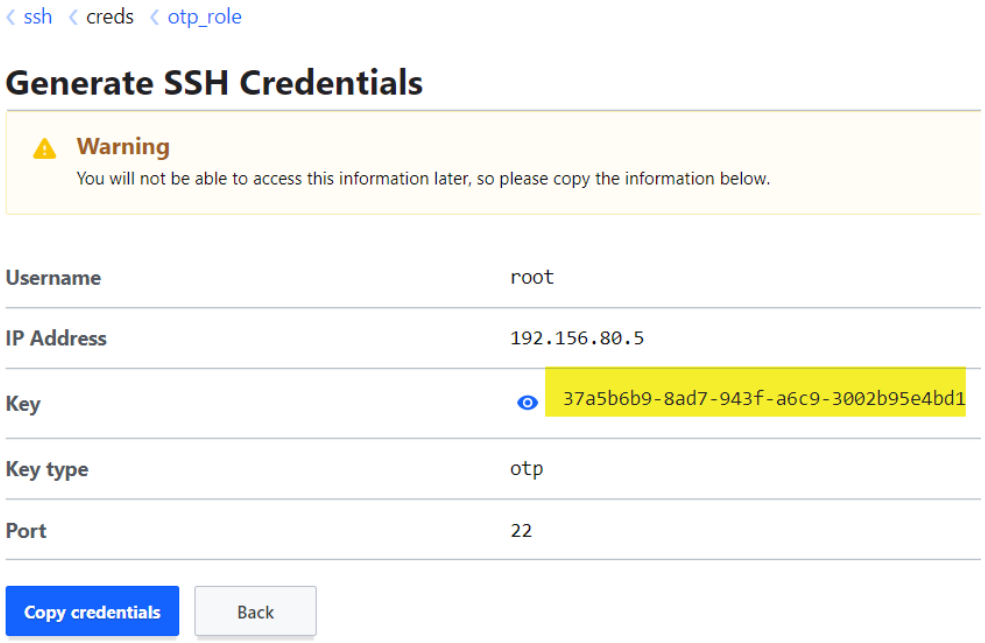
Step 2: Select the **ssh** secret engineer by clicking on it.



Click on **otp_role** and then fill the following fields:
Username: root
IP address: 192.156.80.5
And, click on the Generate button.



This will generate an OTP token which can be used to SSH into the SSH server.



Step 3: Use this to login into the SSH Server.

Command: ssh root@sshserver

Provide the copied token when prompted for password.

```
root@kali:~# ssh root@sshserver
The authenticity of host 'sshserver (192.156.80.5)' can't be established.
ECDSA key fingerprint is SHA256:cFV99PGfdcbUNsA4xXM8qU75E0iR6DZLpkiVLsiaWr0
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'sshserver,192.156.80.5' (ECDSA) to the list of
Password:

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@sshserver:~#
```

In this manner, Vault can be used to login into SSH server securely.

References

- Manage SSH with HashiCorp Vault (https://www.youtube.com/watch?v=bKe4BkDfdvI)
- Vault SSH (https://github.com/errygg/devopsdays-denver-2018)
- Vault documentation (https://www.vaultproject.io/api-docs/secret/ssh)