**SSH SECRET ENGINE**

The Vault SSH secrets engine provides secure authentication and authorization for access to machines via the SSH protocol. The Vault SSH secrets engine helps manage access to machine infrastructure, providing several ways to issue SSH credentials.

The Vault SSH secrets engine supports the following modes.

* Signed ssh certificates
* One-time ssh passwords
* Dynamic ssh keys

**Signed ssh certificates**

The signed SSH certificates is the simplest and most powerful in terms of setup complexity and in terms of being platform agnostic. By leveraging Vault's powerful CA capabilities and functionality built into OpenSSH, clients can SSH into target hosts using their own local SSH keys.

Before a client can request their SSH key be signed, the Vault SSH secrets engine must be configured. This is done by vault administrator.

**Step-1: enable ssh secret engine with path ssh-client-signer**

vault secrets enable -path=ssh-client-signer ssh

Success! Enabled the ssh secrets engine at: ssh-client-signer/

**Step -2 :**

Configure Vault with a CA for signing client keys using the /config/ca endpoint. If you do not have an internal CA, Vault can generate a keypair for you.

curl --header "X-Vault-Token: s.9NuiX2lkeyVB8YhfC7ct3GDS" --request POST --data @generateca.json <http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/config/ca>

{"request\_id":"7e1a03d9-9148-07cb-e5de-a00bf5925ed2","lease\_id":"","renewable":false,"lease\_duration":0,"data":{"public\_key":"ssh-rsa \n"},"wrap\_info":null,"warnings":null,"auth":null}

**Step-3:**

Add the public key to all target host's SSH configuration

sudo curl -o /etc/ssh/trusted-user-ca-keys.pem <http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/public_key>

Add the path where the public key contents are stored to the SSH configuration file as the TrustedUserCAKeys option.

root@ip-10-44-0-183:/etc/ssh# vi sshd\_config

add the following line

TrustedUserCAKeys /etc/ssh/trusted-user-ca-keys.pem

Restart the SSH service to pick up the changes.

systemctl restart ssh.service

Create a named Vault role for signing client keys.

Because of the way some SSH certificate features are implemented, options are passed as a map. The following example adds the permit-pty extension to the certificate.

curl --header "X-Vault-Token: s.9NuiX2lkeyVB8YhfC7ct3GDS" --request POST --data @ sshrole.json [http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ ssh-client-signer/roles/my-role](http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/config/ca)

**sshrole.json**

ubuntu@ip-10-44-0-183:~$ cat sshrole.json

{

"allow\_user\_certificates": true,

"allowed\_users": "\*",

"allowed\_extensions": "permit-pty,permit-port-forwarding",

"default\_extensions": [

{

"permit-pty": ""

}

],

"key\_type": "ca",

"default\_user": "ubuntu",

"ttl": "2400m0s"

}

**CLIENT SSH AUTHENTICATION**

The following steps are performed by the client (user) that wants to authenticate to machines managed by Vault. These commands are usually run from the client's local workstation.

Locate or generate the SSH public key. Usually this is id\_rsa.pub. If you do not have an SSH keypair, generate one:

**Step-1 : Generate ssh :**

ssh-keygen -t rsa –C ubuntu@

this will generate a private and public key pairs.

cat id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDDF5ADOiw4P61jS8AWk6EYwduoZASs3AiruxvFJK6t7e9askH5/gKo84OQWUeGlYkK8Peqix7Aft65NFSQcx2EsaVCyu4AcNuaC7XCdDjImk6xb/p3QdAbq2NYLu8fCdAsfgHiqs/QSIdDo3+3J0Io8supPqfWniopGSjwh6ss0YEXgGz8PmfI7+7OZlPYRwn5aGbPPhKLqI+6WBCmlvOM+MKjtcVn4NL7kP1lL/MZQ+DwkcMT0cJF7z6Cd+HyKbM5hPozeF3YwKmSIzByVDydDtsOyMNLST3hDTGcAOItD7zZb1qPjTCFqFBUGm6baCxL

cat id\_rsa

-----BEGIN RSA PRIVATE KEY-----





-----END RSA PRIVATE KEY-----

**Step-2 :** Ask Vault to sign your **public key**.

**publickey.json :**

cat publickey.json

{

"public\_key" : "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDDF5ADOiw4P61jS8AWk6EYwduoZASs3AiruxvFJK6t7e9askH5/gKo84OQWUeGlYkK8Peqix7Aft65NFSQcx2EsaVCyu4AcNuaC7XCdDjImk6xb/p3QdAbq2NYLu8fCdAsfgHiqs/QSIdDo3+3J0Io8supPqfWniopGSjwh6ss0YEXgGz8PmfI7+7OZlPYRwn5aGbPPhKLqI+6WBCmlvOM+MKjtcVn4NL7kP1lL/MZQ+DwkcMT0cJF7z6Cd+HyKbM5hPozeF3YwKmSIzByVDydDtsOyMNLST3hDTGcAOItD7zZb1qPjTCFqFBUGm6baCxL73k4qnFTniPq5WyXogrx ubuntu@"

}

**Curl command to sign the key :**

curl --header "X-Vault-Token: s.9NuiX2lkeyVB8YhfC7ct3GDS" --request POST --data @publickey.json [http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/sign/my-role](http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/sign/my-role%20)  > signed-cert.pub

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 2725 0 2317 100 408 59410 10461 --:--:-- --:--:-- --:--:-- 69871

**Using cli you can use the following command:**

vault write -field=signed\_key ssh-client-signer/sign/my-role \

public\_key=@$HOME/ubuntu/id\_rsa.pub > signed-cert.pub

You will get a signed public key from the above command with which you can ssh into your system.

To View enabled extensions, principals, and metadata of the signed key.

ssh-keygen -Lf sign-cert.pub

**output:**

signed-cert.pub:

Type: ssh-rsa-cert-v01@openssh.com user certificate

Public key: RSA-CERT SHA256:bYxSlPQkRvzKlh7Vfnh7kDOv1LZAAc3Ut9+P4ffmU58

Signing CA: RSA SHA256:rMjg1OLURLDL5SMGk1rx7lPxyaYNO6inaIZ1Kp4y6yM

Key ID: "vault-root-6d8c5294f42446fcca961ed57e787b9033afd4b64001cdd4b7df8fe1f7e6539f"

Serial: 17277563621986071282

Valid: from 2020-10-14T04:58:10 to 2020-10-14T05:28:40

Principals:

ubuntu

Critical Options: (none)

Extensions:

permit-pty

signed-cert.pub:2: invalid key: invalid format

**ssh into your machine:**

SSH into the host machine using the signed key. You must supply both the signed public key from Vault **and** the corresponding private key as authentication to the SSH call.

ssh -i signed-cert.pub -i id\_rsa [ubuntu@10.44.0.183](mailto:ubuntu@10.44.0.183)

The authenticity of host '10.44.0.183 (10.44.0.183)' can't be established.

ECDSA key fingerprint is SHA256:AjO/KC3anL4bKwt5cd6qYhG16ulkXgMEf3wqTd0h0TQ.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.44.0.183' (ECDSA) to the list of known hosts.

Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1025-aws x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

System information as of Wed Oct 14 05:03:43 UTC 2020

System load: 0.0 Processes: 106

Usage of /: 3.4% of 96.88GB Users logged in: 1

Memory usage: 64% IP address for eth0: 10.44.0.183

Swap usage: 0%

\* Kubernetes 1.19 is out! Get it in one command with:

sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

\* Canonical Livepatch is available for installation.

- Reduce system reboots and improve kernel security. Activate at:

https://ubuntu.com/livepatch

6 packages can be updated.

0 updates are security updates.

New release '20.04.1 LTS' available.

Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Oct 14 04:55:25 2020 from 157.48.199.38

**HOST KEY SIGINING**

For an added layers of security, it is recommend to enable host key signing. This is used in conjunction with client key signing to provide an additional integrity layer. When enabled, the SSH agent will verify the target host is valid and trusted before attempting to SSH. This will reduce the probability of a user accidentally SSHing into an unmanaged or malicious machine.

**Step-1** : Mount the secrets engine. For the most security, mount at a different path from the client signer.

vault secrets enable -path=ssh-host-signer ssh

**Step-2** : Configure Vault with a CA for signing host keys using the /config/ca endpoint. If you do not have an internal CA, Vault can generate a keypair for you.

curl --header "X-Vault-Token: s.9NuiX2lkeyVB8YhfC7ct3GDS" --request POST --data @generateca.json [http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ ssh-host-signer/config/ca](http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/%20ssh-host-signer/config/ca)

**output:**

{"request\_id":"133887da-5038-8a02-3701-49b62676ea8d","lease\_id":"","renewable":false,"lease\_duration":0,"data":{"public\_key":"ssh-rsa \n"},"wrap\_info":null,"warnings":null,"auth":null}

you can extend the host key ttls using following command :

vault secrets tune -max-lease-ttl=87600h ssh-host-signer

**Step-3 :**

Create a role for signing host keys. Be sure to fill in the list of allowed domains, set allow\_bare\_domains, or both.

vault write ssh-host-signer/roles/hostrole \

key\_type=ca \

ttl=87600h \

allow\_host\_certificates=true \

allowed\_domains="localdomain,example.com" \

allow\_subdomains=true

**Step-4 :**

Sign the host's SSH public key

In publickey.json place the “ssh\_host\_rsa\_key.pub”

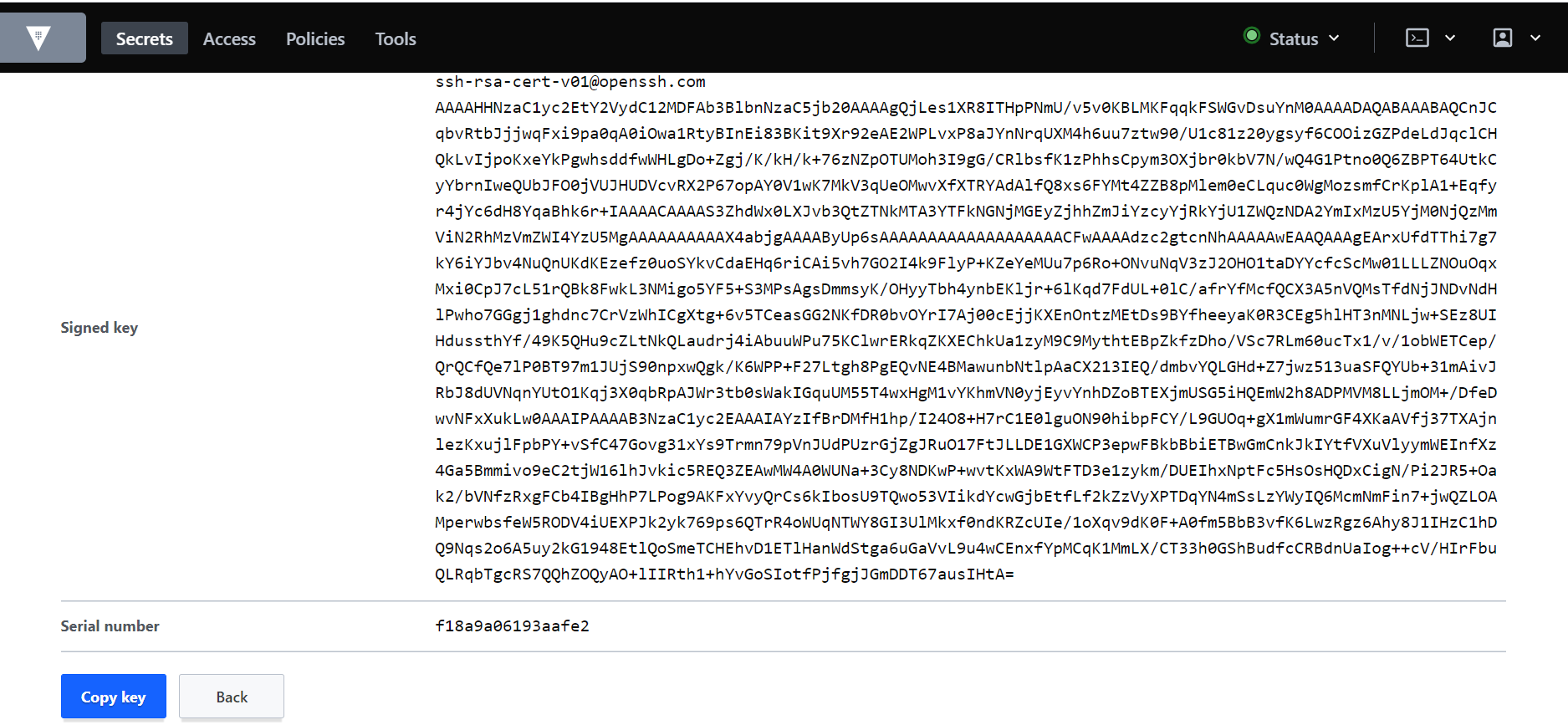
curl --header "X-Vault-Token: s.9NuiX2lkeyVB8YhfC7ct3GDS" --request POST --data @publickey.json [http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/sign/my-role](http://1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com/v1/ssh-client-signer/sign/my-role%20)  > /etc/ssh/ssh\_host\_rsa\_key-cert.pub

To do it from UI :

1. Goto ssh-host-signer/
2. Click on hostrole
3. In certificate type select host as shown in the following screen.



1. Paste you **ssh\_host\_rsa\_key.pub** in the public key.
2. Click on sign and then copy the generated signed key.

****

Set permissions on the certificate to be 0640 :

chmod 0640 /etc/ssh/ssh\_host\_rsa\_key-cert.pub