PUBLIC REPOSITORY

msvechla (/u/msvechla/)/vaultbot (/r/msvechla/vaultbot/) ☆

Last pushed: 3 days ago

Short Description

Vaultbot is a Hashicorp Vault PKI client, built for infrastructure certificate automation.

Full Description

(https://gitlab.com/msvechla/vaultbot/commits/master) 

(https://gitlab.com/msvechla/vaultbot/commits/master)

Vaultbot

Lightweight Hashicorp Vault (https://www.vaultproject.io/) PKI client, built for infrastructure automation. Automatically request and renew certificates generated inside vault via the PKI backend (https://www.vaultproject.io/docs/secrets/pki/index.html).

By default Vaultbot will only renew certificates that are due for renewal within a specified period. Therefore Vaultbot is ideal for running at a fixed interval (e.g. crontab). This tool is also inspired by the well-known certbot (https://github.com/certbot/certbot) for letsencrypt.

Available Tags

- `${MAJOR}`
- `${MAJOR}.${MINOR}`
- `${MAJOR}.${MINOR}.${PATCH}`

Getting Started

Requesting and renewing a certificate is straightforward. See the following self-explanatory example:

```
./vaultbot --vault_addr=http://localhost:1234 --vault_token=myroot --pki_mount=pki --pki_role_name=example-dot-com  --pki_common_name=myd
```

You can also see further usage information by running `./vaultbot --help`

Get the latest release

In addition to the Docker Image (https://hub.docker.com/r/msvechla/vaultbot/), there are automated builds for all major platforms. You can find all releases in the tags (https://gitlab.com/msvechla/vaultbot/tags) section. Simply click on downloads and select "artifacts" to get the desired version. Alternatively feel free to build the latest release from master yourself.

Configuration

You can configure Vaultbot by specifying command line options or the corresponding environment variables.

```
Usage:
  vaultbot [OPTIONS]

Application Options:
  -v, --verbose                   Show verbose debug information

Vault Options:
      --vault_addr=               The address of the Vault server expressed as a URL and port (default: http://127.0.0.1:8200) [$VAULT_A
      --vault_cacert=             Path to a PEM-encoded CA cert file to use to verify the Vault server SSL certificate. [$VAULT_CACERT]
      --vault_capath=             Path to a directory of PEM-encoded CA cert files to verify the Vault server SSL certificate. If VAULT_(
                                  take precedence. [$VAULT_CAPATH]
      --vault_client_cert=        Path to a PEM-encoded client certificate for TLS authentication to the Vault server. [$VAULT_CLIENT_CE
      --vault_client_key=         Path to an unencrypted PEM-encoded private key matching the client certificate. [$VAULT_CLIENT_KEY]
      --vault_client_timeout=     Timeout variable for the vault client. [$VAULT_CLIENT_TIMEOUT]
      --vault_skip_verify         If set, do not verify Vault's presented certificate before communicating with it. Setting this variabl
                                  testing. [$VAULT_SKIP_VERIFY]
      --vault_tls_server_name=    If set, use the given name as the SNI host when connecting via TLS. [$VAULT_TLS_SERVER_NAME]
      --vault_max_retries=        The maximum number of retries when a 5xx error code is encountered. [$VAULT_MAX_RETRIES]
      --vault_token=              The Vault authentication token. [$VAULT_TOKEN]
      --vault_renew_token         If set, vaultbot tries to automatically renew the current token [$RENEW_TOKEN]

PKI Options:
      --pki_mount=                Specifies the PKI backend mount path (default: pki) [$PKI_MOUNT]
      --pki_role_name=            Specifies the name of the role to create the certificate against [$PKI_ROLE_NAME]
      --pki_common_name=          Specifies the requested CN for the certificate [$PKI_ROLE_NAME]
      --pki_alt_names=            Specifies requested Subject Alternative Names, in a comma-delimited list [$PKI_ALT_NAMES]
      --pki_ip_sans=              Specifies requested IP Subject Alternative Names, in a comma-delimited list [$PKI_IP_SANS]
      --pki_ttl=                  Specifies requested Time To Live [$PKI_TTL]
      --pki_exclude_cn_from_sans  If set, the given common_name will not be included in DNS or Email Subject Alternate Names (as appropr
      --pki_renew_percent=        Percentage of requested certificate TTL, which triggers a renewal when passed (>0.00, <1.00) (default:
      --pki_renew_time=           Time in hours before certificate expiry, which triggers a renewal (e.g. 12h, 1m). Takes precedence ove
                                  [$PKI_RENEW_TIME]
      --pki_force_renew           If set, the certificate will be renewed without checking the expiry [$PKI_FORCE_RENEW]
      --pki_cert_path=            Path to the requested / to be updated certificate (default: cert.pem) [$PKI_CERT_PATH]
      --pki_cachain_path=         Path to the CA Chain of the requested / to be updated certificate (default: chain.pem) [$PKI_CACHAIN_PA
      --pki_privkey_path=         Path to the private key of the requested / to be updated certificate (default: key.pem) [$PKI_PRIVKEY_
  -y, --auto_confirm              If set, user prompts will be auto confirmed with yes [$AUTO_CONFIRM]

Help Options:
  -h, --help                      Show this help message
```

Renewing existing certificates

When Vaultbot is run and `pki_cert_path` points to an existing certificate, the certificate is only renewed and overwritten when specific criteria are met.

You can either specify `pki_renew_percent` (e.g. 0.75), to renew the certificate after 75% of its lifespan has been reached. Otherwise you can specify `pki_renew_time` to set a fixed amount of time before the expiry-date, which will trigger a renewal when passed.

If you want to renew the certificate on every run, you can specify the `pki_force_renew` flag.

Sanity checks and user user confirmation

By default Vaultbot performs a small set of sanity checks before overwriting an existing certificate at the `pki_(cert/cachain/privkey/_path)` locations.

If the newly requested certificate data (common name, dns alternative names, ip SANS) differs from the data specified in the existing certificate at the location, the user will be asked for confirmation.

If you want to skip these checks in automated environments, you can specify the `y` or `auto_confirm` flag.

Contributing

Please read CONTRIBUTING.md () for details on our code of conduct, and the process for submitting pull requests to us.

Versioning

We use SemVer (http://semver.org/) for versioning. For the versions available, see the tags on this repository (https://gitlab.com/msvechla/kubehiera/tags) or take a look at the CHANGELOG.md (./CHANGELOG.md)

Authors

- **Marius Svechla** - *Initial work*

See also the list of contributors (https://gitlab.com/msvechla/kubehiera/graphs/master) who participated in this project.

License

Acknowledgments

- The official vault go client (https://github.com/hashicorp/vault/tree/master/api)

Docker Pull Command

```
docker pull msvechla/vaultbot
```

Owner

msvechla