vault-keepass-import Documentation

Release 0.0.0

singuliere

Feb 03, 2019

Contents

vault-keepass-import is a CLI to import KeePass secrets (using pykeepass) in Hashicorp Vault (using hvac).

Bugs and feature requests can be found in the issue tracker

The *Title* of the entry is used as the last component of the secret path. For instance if importing an entry with *Title mysecret* in the *mygroup* group, the path *secret/keepass/mygroup/mysecret* will be used.

There may be multiple entries with the same title in a group. The path is made unique by appending the *UUID* of the entry to the title. For instance to entries with the same *title1* in the group *group1* will be imported as *group1/title1 (TJxu0nxlyEuaKYNYpi0NPQ==)* and *group1/title1 (kFl/iRsoVUWDUdmmCDXwJg==)*. The *UUID* is not appended if the title is unique.

- *User name* from the *Entry* tab is imported as is under the key *username*
- *Password* from the *Entry* tab is imported as is under the key *password*
- *URL* from the *Entry* tab is imported as is under the key *password*
- *Notes* from the *Entry* tab is imported as is under the key *notes*
- *Expires* from the *Entry* tab is imported under the key *expiry_time*. It is only imported if set and converted to epoch.
- *Tags* from the *Properties* tab is imported as is under the key *tags*
- *UUID* from the *Properties* tab is imported as is under the key *uuid*
- *String fields* from the *Advanced* tab are imported as is with a key matching their *Name* and a value set to their *Value*
- *File attachments* from the *Advanced* tab are imported with a key set to **id/filename** (for instance if there only is one **foo.txt** attachment, it will have the key **0/foo.txt**) and the value is base64 encoded. For instance, the actual value can be retrieved from the command line with:

```
$ vault kv get -field 8/attached.txt secret/mysecret | base64 --decode
```

- *mtime*, *ctime*, *atime* are always imported and converted to epoch

Quick start

```
$ pip3 install vault-keepass-import
$ export VAULT_ADDR=https://myvault.com:8200
$ export VAULT_TOKEN=mytoken
$ vault-keepass-import --token $VAULT_TOKEN \
                       --vault $VAULT_ADDR \
                       --password kdbxpassword \
                       database.kdbx
$ vault kv list secret/keepass
Keys
----
Group1/
Group2/
secret1
secret2
$ vault kv get secret/keepass/secret1
====== Metadata ======
Key              Value
---              -----
created_time     2019-01-29T13:52:32.79894513Z
deletion_time    n/a
destroyed        false
version          1
==== Data ====
Key      Value
---      -----
atime    1465498383
ctime    1465498332
icon     0
mtime        1527099465
password     strongpassword
username     someuser
uuid         5uCDWvHUQjyGnyBlRw9CFA==
```

Testing the import

- Download and [install Hashicorp Vault](#)

- Run vault in development mode (the storage is reset when it restarts)

```
$ vault server -dev
...
Root Token: s.PTNNfrICGosELrJeX2ojPIS6
...
```

- Assuming the password to the KeePass database is *kdbxpassword*, run an import with:

```
$ vault-keepass-import --token s.PTNNfrICGosELrJeX2ojPIS6 \
                       --vault http://127.0.0.1:8200 \
                       --password kdbxpassword \
                       database.kdbx
```

Command help

```
vault-keepass-import --help
```

Contributions

4.1 Release management

- Prepare a new version

- version=1.3.0 ; perl -pi -e "s/^version.*/version = $version/" setup.cfg ; for i in 1 2 ; do python setup.py sdist ; amend=$(git log -1 –oneline | grep –quiet "version $version" && echo –amend) ; git commit $amend -m "version $version" ChangeLog setup.cfg ; git tag -a -f -m "version $version" $version ; done

- Publish a new version

- python setup.py sdist upload –sign

- git push ; git push –tags

- pypi maintenance

- python setup.py register # if the project does not yet exist

- trim old versions at https://pypi.python.org/pypi/vault-keepass-import