Nutzerdatenbank laden / hashen / prüfen

Nutzerdatenbank laden / hashen

Für das Laden und ggf. Rehashen der Nutzerdatenbank (users.toml) kann folgendes Script genutzt werden. Es prüft zunächst, ob die Konfiguration von bffh valid ist. Wenn nicht, wird nichts unternommen. Im Anschluss werden die Nutzer neu geladen und gleichzeitig wieder exportiert, um etwaige unverschüsselte Passwörter automatisch zu hashen.

> Hinweis: Das Script basiert auf einem auf dem System aktiven **systemd** Service namens `bffh.service`

```
mkdir -p /opt/fabinfra/scripts/
vim /opt/fabinfra/scripts/bffh-load-users.sh
```

```
#/!bin/bash
#check config
BASE="/opt/fabinfra/bffh/target/release"
DATA="/opt/fabinfra/bffh-data"
CFG="$DATA/config/bffh.dhall"
USERS="$DATA/config/users.toml"

while [[ $# -gt 0 ]]; do
  case $1 in
    -h|--help)
      echo "-h|--help   - show help"
      echo "-r|--rehash - overwrite users.toml with hashed passwords (ensure secure secrets)"
      exit 1
      ;;
    -r|--rehash)
      REHASH="y"
      shift
      ;;
    -*)
      ;;
  esac
```

```
done

echo "use -h|--help to show additional script options!"

$BASE/bffhd --check --config $CFG > /dev/null
if [ $? == 0 ]; then
	#pre-check bffh.dhall
	echo "Config is valid. Loading users ..."
	$BASE/bffhd --verbose --config $CFG --load=$USERS

	if [ $? == 0 ]; then
		#then load users.toml
		$BASE/bffhd --verbose --config $CFG --dump-users /tmp/users.toml --force
	else
		echo "Error: Newly given users.toml is invalid!"
		exit 1
	fi

	#if this was successful and service is running, restart it, also do nothing
	if [ $? == 0 ]; then
		if [[ $REHASH == "y" ]]; then #overwrite users if --rehash option is given (not null)
			echo "Rehasing users.toml!"
			cat /tmp/users.toml > $USERS
			rm /tmp/users.toml
		fi
		FAS="bffh.service"
		STATUS="$(systemctl is-active $FAS)"
		if [ "${STATUS}" = "active" ]; then
			echo "restarting $FAS"
			systemctl restart $FAS
		else
			echo -e "\n\n$FAS not active/existing. Not restarting bffh service!"
		fi
	fi

else
	echo "Error: Currently loaded users.toml is invalid!"
	exit 1
fi
```

```
chmod +x /opt/fabinfra/scripts/bffh-load-users.sh
```

Nutzerdankenbank prüfen

Folgendes Python-Script erlaubt die Auswertung einer users.toml Datei. Es zählt die Nutzer und deren zugewiesene Rollen, validiert eventuell vergebene Cardkeys und gibt Hinweise bei möglichen Datenbankfehlern.

> Das Script benötigt mindestens Python 3.11. Erst ab dieser Version wird `tomllib` unterstützt!

```
vim /opt/fabinfra/scripts/show-users-toml-stats.py
```

```
'''
This script validates users.toml for several aspects
The script requires at least Python 3.11

Written by Mario Voigt (vmario89) - Stadtfabrikanten e.V. - 2024

ToDos
- enter bffh.dhall path to check roles against users.toml. If our toml contains roles, which bffh does not know, we should also warn!
'''

import argparse
import os
import sys
import tomllib
import uuid

'''
cardkeys for FabAccess use Uuid format in Version v4 (see https://docs.rs/uuid/latest/uuid/struct.Uuid.html)
allowed formattings:
- simple: a1a2a3a4b1b2c1c2d1d2d3d4d5d6d7d8
- hyphenated: a1a2a3a4-b1b2-c1c2-d1d2-d3d4d5d6d7d8
- urn: urn:uuid:A1A2A3A4-B1B2-C1C2-D1D2-D3D4D5D6D7D8
- braced: {a1a2a3a4-b1b2-c1c2-d1d2-d3d4d5d6d7d8}
'''
```

```
def is_valid_uuid(val):
    try:
        _uuid = uuid.UUID(val, version=4)
        return True
    except ValueError:
        return False

def main():

    parser = argparse.ArgumentParser()
    parser.add_argument("--db", type=str, help="path of users.toml user database file")
    args = parser.parse_args()

    if args.db is None:
        print("Error: no users.toml given. Please add with '--db </path/to/users.toml>'")
        sys.exit(1)

    countUsers = 0
    countUsersWithoutCardkeyOrPassword = 0
    uniqueRoles = []
    countUserWithoutRoles = 0
    countPassword = 0
    countPasswordUnencrypted = 0
    countPasswordEncrypted = 0
    countCardkey = 0
    countCardkeyInvalid = 0
    countUnknownKeys = 0

    countWarnings = 0

    #a definition of valid keys within a user section of FabAccess
    knownKeys = ['roles', 'passwd', 'cardkey']

    usertoml = args.db

    print("{} Checking database {}\n".format("*"*25, "*"*25))

    file_stats = os.stat(usertoml)
    #print(file_stats)
    print("Database size: {} Bytes ({:0.5f} MB)".format(file_stats.st_size, file_stats.st_size / (1024 * 1024)))
```

```
    if file_stats.st_size == 0:
        print("Error: File size is zero! Database is corrupted!")
        sys.exit(1)

    print("\n")

    with open(usertoml, "rb") as f:
        try:
            data = tomllib.load(f)
        except Exception as e:
            if "Cannot declare" in str(e) and "twice" in str(e):
                print("Error: found at least one duplicate user. Cannot parse database. Please fix and try again.
Message: {}".format(str(e)))
            elif "Invalid value" in str(e):
                print("Error: Some user contains a key without value (e.g. 'passwd = '). Cannot parse database. Please
fix and try again. Message: {}".format(str(e)))
            elif "Expected '=' after a key" in str(e):
                print("Error: Found an incorrect key/value mapping. Cannot parse database. Please fix and try again.
Message: {}".format(str(e)))
            else:
                print(str(e))
            sys.exit(1)

        for user in data:
            print("--- {}".format(user))

            for key in data[user].keys():
                if key not in knownKeys:
                    print("Warning: User '{}' contains unknown key '{}' (will be ignored by BFFH server)".format(user,
key))
                    countWarnings += 1
                    countUnknownKeys += 1

            if "roles" in data[user]:
                roles = data[user]["roles"]
                for role in roles:
                    if role not in uniqueRoles:
                        uniqueRoles.append(role)
                if roles is None: #if role key is defined but empty
                    countUserWithoutRoles += 1
```

```
            else: #if role key is not existent
                countUserWithoutRoles += 1

            if "passwd" in data[user]:
                passwd = data[user]["passwd"]
                countPassword += 1
                if passwd.startswith("$argon2") is False:
                    print("Warning: Password for user '{}' is not encrypted!".format(user))
                    countWarnings += 1
                    countPasswordUnencrypted += 1
                else:
                    countPasswordEncrypted += 1

            if "cardkey" in data[user]:
                cardkey = data[user]["cardkey"]
                if is_valid_uuid(cardkey) is False:
                    print("Warning: Cardkey for user '{}' contains invalid cardkey (no UUID v4)".format(user))
                    countCardkeyInvalid += 1
                    countWarnings += 1

                countCardkey += 1

            if "passwd" not in data[user] and "cardkey" not in data[user]:
                countUsersWithoutCardkeyOrPassword += 1

            countUsers += 1
            print("\n")

        print("\n")

        if countUsers == 0:
            print("Error: Database does not contain any users!")
            sys.exit(1)

        print("{} Database statistics {}\n".format("*"*25, "*"*25))
        print("- Total users: {}".format(countUsers))
        print("- Total unique roles: {}".format(len(uniqueRoles)))
        print("- Total passwords: {} (encrypted: {}, unencrypted: {})".format(countPassword,
countPasswordEncrypted, countPasswordUnencrypted))
        print("- Total cardkeys: {}".format(countCardkey))
```

```
        print("\n")

        print("{} Important information {}\n".format("*"*25, "*"*25))
        if countUnknownKeys > 0:
            print("- {} unknown keys (will be ignored by BFFH server)".format(countUnknownKeys))

        if countUserWithoutRoles > 0:
            print("- {} users without any roles. They won't be able to do something as
client!".format(countUserWithoutRoles))

        if len(uniqueRoles) == 0:
            print("- Globally, there are no roles assigned for any user. They won't be able to do something as client!")

        if countCardkeyInvalid > 0:
            print("- {} invalid cardkeys in your database. They won't be able to authenticate at BFFH server by
keycard!".format(countCardkeyInvalid))

        if countUsersWithoutCardkeyOrPassword > 0:
            print("- {} users without both: password and cardkey. They won't be able to login
anyhow!".format(countUsersWithoutCardkeyOrPassword))

        if countWarnings > 0:
            print("- {} warnings in total. You might need to optimize your user database!".format(countWarnings))

if __name__ == "__main__":
    main()
```

Script benutzen:

```
python3 /opt/fabinfra/scripts/show-users-toml-stats.py --db /opt/fabinfra/bffh-data/config/users.toml
```

Version #18
Erstellt: 14 November 2024 16:33:57 von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 6 Dezember 2024 00:45:52 von Mario Voigt (Stadtfabrikanten e.V.)