ktcal2 Documentation

Release 0.1

cr0hn

August 01, 2014

Contents

CHAPTER 1

What's this project?

This project aims to perform a library/tool make a SSH brute force password attack that you can use **as a library as a command line tool**.

The goal of ktcal2 is that it uses new **non-blocking I/O AsyncIO framework**, included **Python 3.4**.

Some links:

- **Documentation:** http://ktcal2.readthedocs.org (currently not working).
- AsyncSSH: This project use AsyncSSH library internally.

CHAPTER 2

Licence

This project is **BSD**... Copy it! And, if you remember, please mention me in credits :)

CHAPTER 3

How to install

3.1 PIP

```
sudo python3.4 -m pip install ktcal2
python3.4 kt-cal2 -h
```

3.2 Manually

```
git clone https://github.com/cr0hn/ktcal2.git ktcal2
cd ktcal2
sudo python3.4 -m pip -r requirements.txt install
cd ktcal2/bin
python3.4 kt-cal2.py -h
```

CHAPTER 4

How use it?

You can use this project in command line tool or as a library, in your Python projects.

4.1 As a tool

You can test SSH passwords, using a wordlist or brute forcer password generation.

4.1.1 Using wordlist

Basic usage:

```
python3.4 ktcal2.py --password-wordlist my_password_list.txt -u root 127.0.0.1
```

Using user name wordlist:

```
python3.4 ktcal2.py --password-wordlist my_password_list.txt --user-wordlist user_names.txt 127.0.0.1
```

4.1.2 Using password wordlist brute force

ktcal2 can generates all combinations of wordlist based in rules.

If we want to generate all combinations, with 4 word length **(–max-length 4)** using only **numbers (-N), 0000-9999**:

```
python3.4 ktcal2.py -u root --max-length 4 -N 127.0.0.1
```

All combinations. 2 max and minimum length, only numbers 00-99:

```
python3.4 ktcal2.py -u root -N --max-length 2 --min-length 2 127.0.0.1
```

All combinations. 2 max and minimum length. Using numbers, low and upper letters (00..aa..AA):

```
python3.4 ktcal2.py -u root -N -c -C --max-length 2 --min-length 2 127.0.0.1
```

4.2 As a library

```
from ktcal2.api import run
from ktcal2.lib.data import GlobalParameters

def custom_display(message):
    """Displays debug info in a custom way"""
    print("----->>> %s <<<-----" % message)


if __name__ == "__main__":
    # Configure password generator, for brute forcer mode.
    password_config = PasswordConfig(low_chars=True,
                                     numbers=True,
                                     special=True)

    # Configure global parameters
    config = GlobalParameters(target="127.0.0.",
                              verbosity=2,

                              # If we wan to display info
                              display_function=custom_display,

                              # Net options
                              concurrency=20,

                              # Credentials
                              username_list=("root",),
                              password_config=password_config)

    main(config)
```

CHAPTER 5

API

Content:

5.1 api

5.2 bin.kt_cal2

5.3 lib.data

5.4 lib.ssh_brute

CHAPTER 6

Indices and tables

- *genindex*
- *modindex*
- *search*