**CRYPTOCAT**

Create a Python script called **cryptocat.py** able to add encryption and decryption functionalities to the standard **netcat** linux command. Encryption and decryption must be done using the **openssl enc** command.

**Example of *netcat* command**

* Server side

netcat –l port

–l 🡪 Listen for an incoming connection rather than initiating a connection to a remote host.

port 🡪 Specify the source port **netcat** should use, subject to privilege restrictions and availability

* Client side

netcat <hostname> port

hostname 🡪 is the IP address of the server you want to connect to

port 🡪 integer, specify the source port listening on the server side

**Example of *openssl enc* command**

**The openssl enc** command can be used to *encrypt* and *decrypt* data blocks using a large set of cryptographic algorithms.

The synopsis of the command is the following:

openssl enc [-algorithm] [-e] [-d] [-k key] [-in file] [-out file]

-algorithm 🡪 specify the encryption algorithm must be used (openssl enc –list for a full list of the algorithms)

-e 🡪 encrypt a file/text

-d 🡪 decrypt a file/text

-k 🡪 can be used to specify a secret key for the encryption

-base64 🡪 out text in **base64** format (useful when data must be sent through the network)

The **cryptocat.py** script should be invoked as follows:

Cryptocat [options][hostname] port

|  |  |  |  |
| --- | --- | --- | --- |
| **Options** | **Type** | **Optional** | **Description** |
| --listen | boolean | **YES** | if *True*: run the script in **server** mode  if *False*: run the script in **client** mode  **default:** *False* |
| --key | str | **YES** | Specify the secret key for the encryption/decryption  **default:** *empty string* |
| --algorithm | str | **YES** | Specify the encryption/decryption algorithm  **default:** *-pbkdf2* |
| --hostname | str | **YES** | In client mode, specify the ip address of the server  **default:** *localhost* |
| port | int | **NO** | Specify the port number of the server |

When the script is executed in **server mode** it has to

1. Receive encrypted stream from a client
2. Decrypt data using the specified algorithm and secret key
3. Show the decrypted text on STDOUT

When the script is executed in **client mode** it has to

1. Connect to the specified ip address of the server
2. Read from input some text
3. Encrypt the text in a stream using the specified algorithm and secret key
4. Send the encrypted stream to the server