COMS 4180 Network Security Group Project

Sean Inouye (si2281) || Michael Pinkham (mlp2185) || Akhilesh Srivastava (as5215) || Matthew Suozzo (ms4249)

NOTE: Certificate and private key paths must be provided the the client and server but may be specified using either the absolute or relative path.
NOTE: *None* of the arguments to client commands (get, put) may contain spaces. This breaks parsing. Other whitespace characters are fine.

How to set up?

To set up the environment in a GCE Ubuntu 16.04 VM:

1. `sudo apt-get install git`
2. Clone this repository (let's say `REPO_PATH=~/netsec-proj`)
3. `sudo apt-get install -y python3-pip`
4. `cd ${REPO_PATH}`
5. `sudo pip3 install -r requirements.txt`

Generating Certificates

The client and server use PEM-formatted keys and certs. The keys and certs we used for development are included in the repository but they can also be generated using the included bash script (a short wrapper around OpenSSL):

- For server: `./create_cert.sh server`
- For client: `./create_cert.sh client`

How to Run?

```
server: ./server.py <port> <cert> <key> <client_cert>
client: ./client.py <host> <port> <cert> <key> <server_cert>

Args:
    <port>      : Port Number on which server is running.
    <host>      : IP Address/FQDN of the server on which the Server is
                      hosted/Client will connect
    <cert>      : The .pem file containing that entity's certificate.
    <key>       : The .pem file containing the private key that signs
                      that entity's certificate.
    <*_cert>    : The .pem file containing the other entity's
                          certificate.
```

Example usage:

server: `./server.py 8000 server_cert.pem server_key.pem client_cert.pem`

client: `./client.py 127.0.0.1 8000 client_cert.pem client_key.pem server_cert.pem`

Supported Commands and Formats

```
1) put <filename> <enc-flag> <opt-password>
        This command encrypts the file and stores it on the server.
2) get <filename> <enc-flag> <opt-password>
        Tries to get the file from the server and stores it locally.
3) stop
        Closes the socket and exits.
4) help OR help <cmd>
        Displays a brief explanation of the command(s).
```

NOTE: The server stores the file and its SHA256 hash in the folder from which it was run or at the relative/absolute path provided by the client.

NOTE: The client stores the file in folder from which it was run with the same caveat as the server.