**CPSC 526**

**Assignment 3**

**Tutorial: T01**

**Name: Sam Ao**

**ID: 10111222**

**Partner: Artin Rezaee-Anzabee**

**Running the program:**

The program can be run by: python3 server.py <logOption> <replaceOption> <word to be replaced> <replace with> srcPort destAddress destPort.

To connect to server, open a new terminal and write: nc localhost <port>.

You can also use a browser as the client.

Server will bind to the user specified port and start listening for commands. Server will create a connection to the speciefied destAddress via destPort and listen for server data.

The serve works with the back-door server from assignment2

**Connecting to server and handshake details:**

The server will be bound to the specified source port and will create a new connection to the destination address via the specified destination port on every new client.

The server will log the date and time of the new connection

**Supported command:**

Commands supported by the server are:

* -raw
* -strip
* -hex
* -autoN
* -replace

**Server behaviour:**

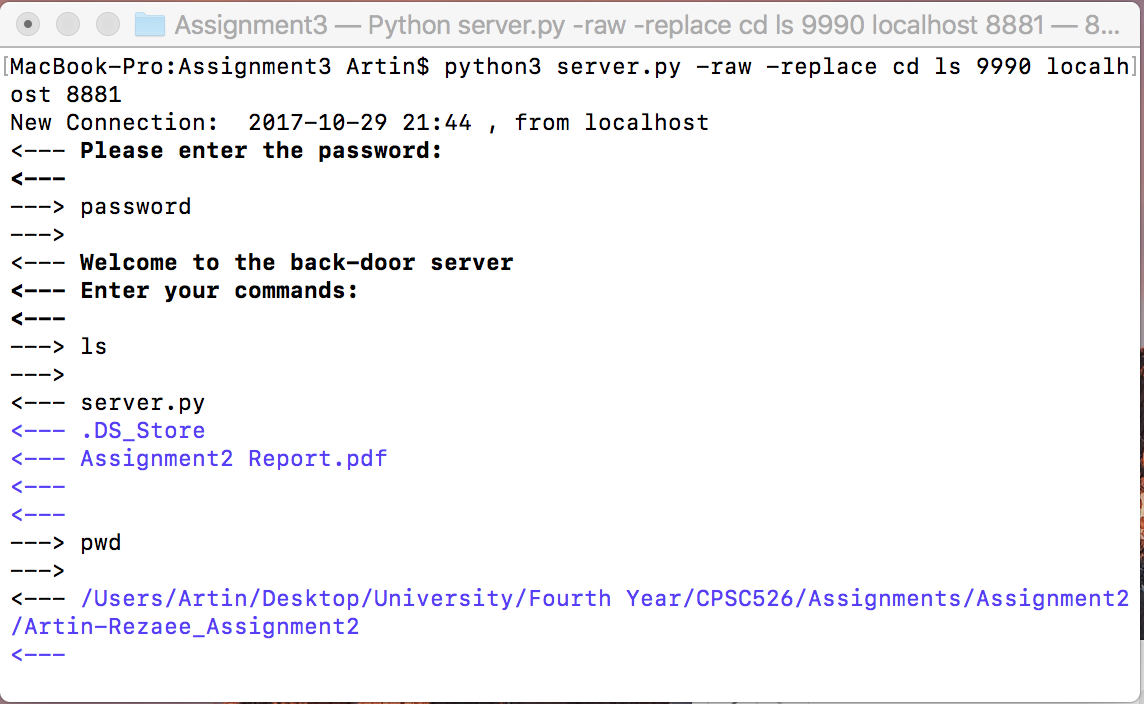
Depending on the log options and the presence of replace options.

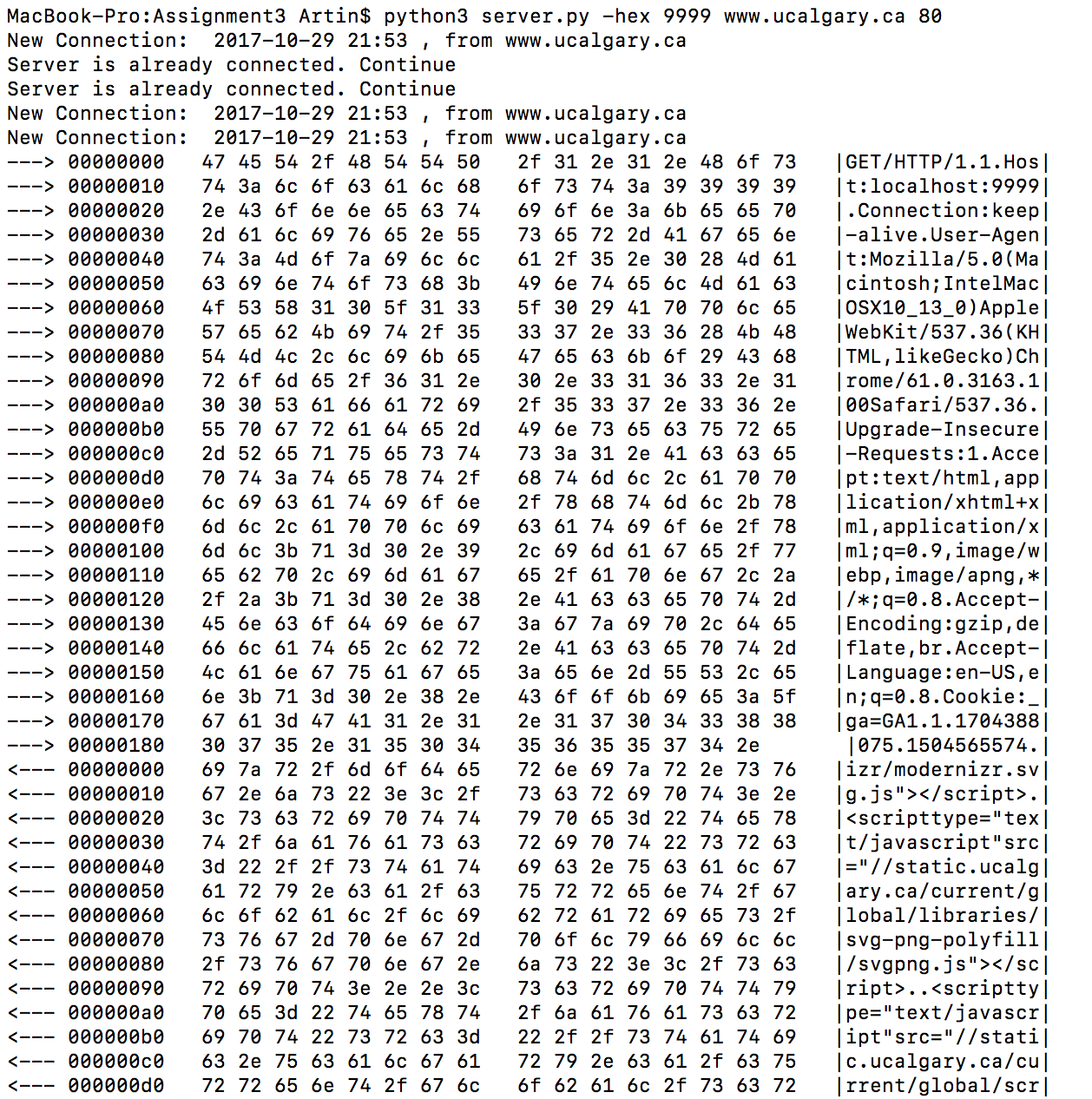
* Server running with the raw command will simply log input and outputs, identified with  and  , in plain string.
* Server running with hex creates a log similar to the Linux hexdump –C command.
* Server running with strip option will replace all the non-printable characters with dots while printable characters are untouched.
* Server running with the autoN command, will divide and log the input and output data in N-byte long chunks and output them on their own line. Each byte in the chunk will be displayed based on its value. If the byte is a backslash, tab, newline or carriage return, it’ll be reported in escaped form, i.e. ‘\\’, ‘\t’, ‘\n’ and ‘\r’ respectively. If the byte is in range 32...127, it will be displayed in raw form. In all other cases the byte will be displayed with a leading slash, followed by a two-digit hexadecimal value of the byte.
* The replace option can be run along with each log option, the replace function will look for the word that it needs to replace and replace it with the user specified word. Server will then output the changed data to its client or server (MITM attack)

All options above will output data both to the client and server of the port forwarding server. All input logs from port-forwarding’s server are identified by  and all outgoing logs from port-forwarding server which it has received from client is identified by 

**Examples:**

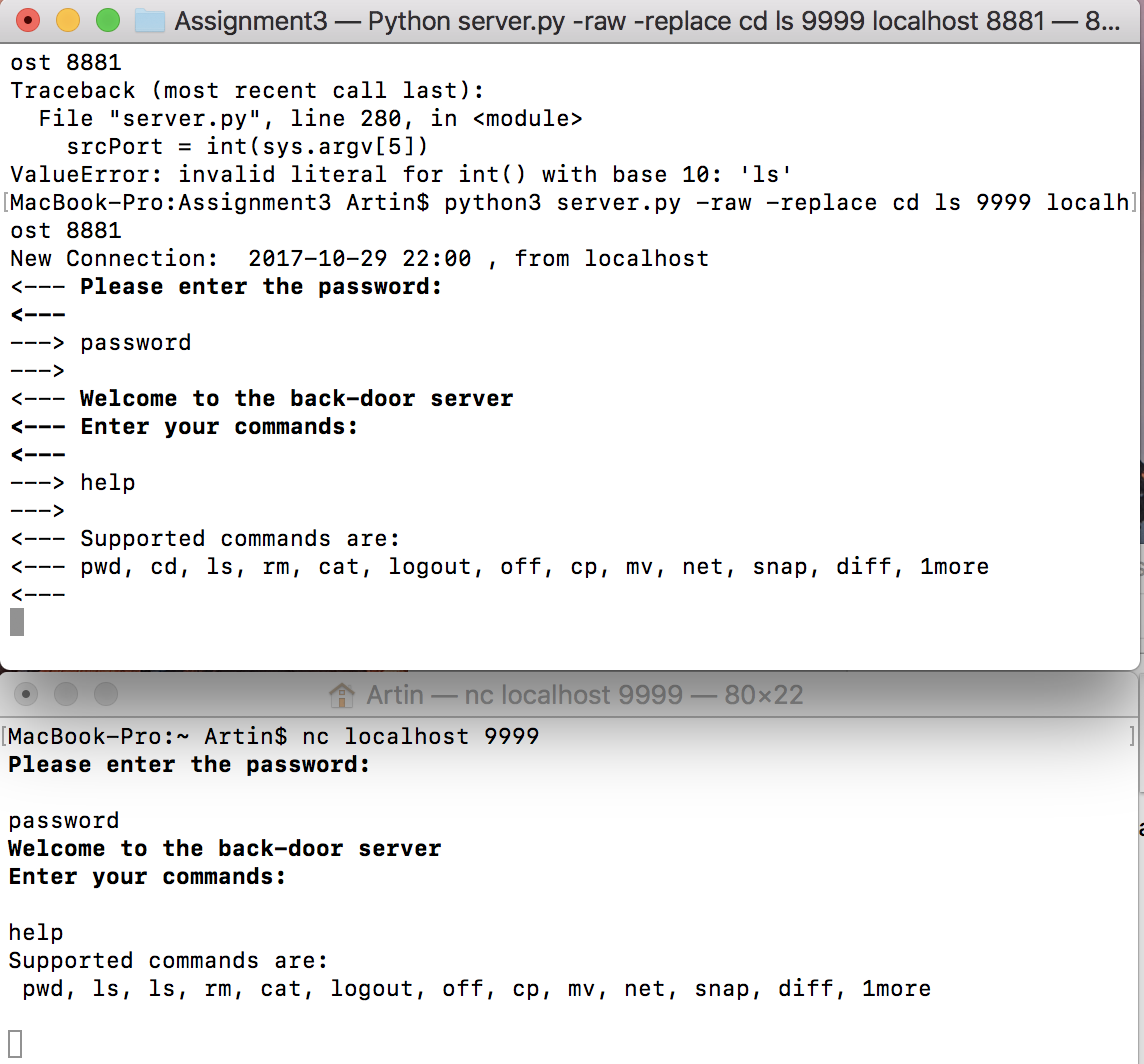
**-Raw:** port-forwarding server view



**-Hex:** port-forwarding server view 

**-Replace:** Both client and port-forwarding server view

The server has replaced all occurrences of cd with ls. Hence, the result of the help command includes 2 instances of ls



**-Strip:** Port-forwarding server and client view



**-AutoN:** Port-forwarding server and client view