Distributed Client Simulating Cluster v0.5 beta:

Written By: Shaun Munshi
Language: Python
OS: Ubuntu Linux (virtualization through Oracle VM VirtualBox)

Summary:

This product is a REST API that, upon being called, will execute a bash command given by a client. There are two sides to this API: a server side and a client side. Upon starting the program, the server side listens for any calls from clients. Furthermore, many security assets including token-based authentication, access control list, etc allow for a more established and secure session between the client and server. Finally, all calls made to the server (whether authorized or unauthorized) are recorded in their respective log files.

Routes:

This product has two different URL routes; a /login and a /bashcall route. First, the client will send a login request to the server through the /login URL. The client inputs the password, and if correct, the server issues a token (JSON string made of mixed letters), which is valid for the next 30 minutes. Then, using the /bashcall route, the client inputs a bash command to be run on the server side. If the token is valid AND the command is included in the access control list, the server runs the command and returns the output to the client. However, if the token is invalid OR the command is outside the access control list, the server will not run the command and will return an JSON string error message of “Unauthorized Request” or “Operation not Permitted”, respectfully. (NOTE: The command from the client can have any number or type of flags included; as long as the base command is included in the access list, the server will run the command with flags)

Logs:

This product generates two different log files: authorizedlog.txt and unauthorizedlog.txt. Any call from a client who provides both a valid token AND a command that is included within the access control list will be recorded in the authorizedlog.txt file. If a call has a missing/incorrect token or gives a command outside of the access control list, the call will get recorded in the unauthorizedlog.txt file. Both files have three attributes: Timestamp, command attempted, and source IP address. The timestamp records when the call was submitted (in UTC time), the command is simply the bash command that was passed in the call, and the source IP address logs the IP of the client that submitted the call. (NOTE: In the case of the unathorizedlog.txt file, if a wrong token is submitted, the “command attempted” field will record “NO TOK” (no token).

Sample Client-Server Communication:

(NOTE: Blue: Client Entered Green: Server Statements)
Sample Access Control list: {“ifconfig”, “echo”}

```
        $ curl -u username http://127.0.0.1:5000/login
                Password: *******
        {“token”: “eyeefiefefefnejijfwfncklnklenekcncklewnkffefwoa,cnalridysbfowbfpwndwndowifew8dnewodeefejdbwodnKDMEJpkDpw-DDJDOWDDJWLdoqjdnxlwjdoshqjoeE”}

        $ curl -X POST -H “Content-Type: application/json” -d ‘{“command”: “ifconfig -s”}’
http://127.0.0.1:5000/bashcall?token=eyeefiefefefnejijfwfncklnklenekcncklewnkffefwoa,cnalridysbfowbfpwndwndowifew8dnewodeefejdbwodnKDMEJpkDpw-DDJDOWDDJWLdoqjdnxlwjdoshqjoeE
eth0      Link encap:Ethernet  HWaddr 00:80:C8:F8:4A:51
          inet addr:192.168.99.35  Bcast:192.168.99.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:190312 errors:0 dropped:0 overruns:0 frame:0
          TX packets:86955 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:30701229 (29.2 Mb)  TX bytes:7878951 (7.5 Mb)
          Interrupt:9 Base address:0x5000

        $ curl -X POST -H “Content-Type: application/json” -d ‘{“command”: “echo device”}’
http://127.0.0.1:5000/bashcall?token=eyeefiefefefnejijfwfncklnklenekcncklewnkffefwoa,cnalridysbfowbfpwndwndowifew8dnewodeefejdbwodnKDMEJpkDpw-DDJDOWDDJWLdoqjdnxlwjdoshqjoeE

Device

        $ curl -X POST -H “Content-Type: application/json” -d ‘{“command”: “ls -al”}’
http://127.0.0.1:5000/bashcall?token=eyeefiefefefnejijfwfncklnklenekcncklewnkffefwoa,cnalridysbfowbfpwndwndowifew8dnewodeefejdbwodnKDMEJpkDpw-DDJDOWDDJWLdoqjdnxlwjdoshqjoeE

Operation Not Permitted!

        $ curl -X POST -H “Content-Type: application/json” -d ‘{“command”: “echo device”}’
http://127.0.0.1:5000/bashcall?token=123

{“Message”: “Unauthorized Request”}
```

Sample Log File:

```
authorizedlog.txt:
        Timestamp:                   Command Ran:            Source IP Address
        6/17/2020 13:23:45           ifconfig -s             127.0.0.1
        6/18/2020 16:45:54           echo device             127.0.0.1

unauthorizedlog.txt:
        Timestamp:                   Command Ran:            Source IP Address
        6/17/2020 13:23:45           ls -al                   127.0.0.1
        6/18/2020 16:45:54           NO TOK                   127.0.0.1
```