Review OMesh: By A1851002

I reviewed the code both manually and using a tool, the tool I used was bandit, bandit is a tool designed to find common security issues in Python code. The README file was very useful and well put together. I found the addition of a table of contents to be very useful and a good idea. The instructions for the set up and running of the code was well put together and informative enabling an easy understanding of how to use access the implementation. The group also did a good job of outlining the function of their implementation and what exactly everything does. Here are some of the things I found while reviewing the code:

Security issues in server.py:

Issue: [B104:hardcoded_bind_all_interfaces] Possible binding to all interfaces.
Severity: Medium Confidence: Medium

```
42      # Read environment variables
43      BIND_ADDRESS = os.environ.get('BIND_ADDRESS', '0.0.0.0')
44      CLIENT_WS_PORT = int(os.environ.get('CLIENT_WS_PORT', 8765))          # Client WS
port
```

Issue: [B104:hardcoded_bind_all_interfaces] Possible binding to all interfaces.
Severity: Medium Confidence: Medium

```
234             await runner.setup()
235             site = web.TCPSite(runner, '0.0.0.0', self.http_port)
236             await site.start()
```

Security issues in client.py

Issue: [B311:blacklist] Standard pseudo-random generators are not suitable for security/cryptographic purposes.
Severity: Low Confidence: High

```
428                     ]
429                     message_entry['message'] = random.choice(random_messages)
430                     if MESSAGE_EXPIRY_TIME != 0:
```

Issue: [B104:hardcoded_bind_all_interfaces] Possible binding to all interfaces.
Severity: Medium Confidence: Medium

```
515         # Start Flask app
516         app.run(host='0.0.0.0', port=5000, debug=False, use_reloader=False)
517
```

Those were all of the issues I was able to find, the group did a good job of making their vulnerabilities challenging to find and a more skilled tester than myself may have been able to find more vulnerabilities.