Client Meeting: API Threat Assessment Tool (AT-AT)

Date: 13th May 2025
Time: 15:00
Platform: Google Meets
Attendees:

- BlueVision ITM: Mr. Ivan Burke
- Skill Issue: All Present
- COS301 Staff: tbd

Agenda:

- **Introductions**
- **Our vision**
    - **Basics of the app**
        - The program takes in an API specification.
            - This can be done using an API specification.
            - A scan can be made based on common endpoints and heuristics based on the type of business.
            - A predefined template to allow users to enter details of more complex API's.
        - Users can switch between pre-made or user made testing profiles that vary in complexity and scale.
        - The program will run the API through a suit of scans both static and dynamic looking for vulnerabilities and abnormal responses as well as outdated vulnerable packages.
            - Current focus is OWASP API top 10 2023 but this can be extended later on.
        - Once the scan is completed a report is generated providing a summary of the scan, any vulnerabilities found, recommended actions, compliance outcomes and a security score.
            - For threat detection and recommended actions a database could be consulted, client suggestions on where to look for this information would be welcomed.
        - We've identified these as the core components of the app but we still need to do more research into additional features, client suggestions are welcome here as well.

- **Technology**
    - Version Control: *GitHub*.
    - Back end: *python*.
    - Front end: *react*.
    - Database: *postgresSQL*.
- **Requirements**
    - **Functional Requirements**
        - Use Registration.
        - User Login.
        - Form Validation.
        - Input via known API specification.
        - Discovery via Heuristic Scan.
        - Ability to select various pre-configured testing profile
        - API scanning and testing.
            - Identify vulnerabilities
            - Report outdated packages.
            - Detect Abnormal responses.
        - Generate an Assessment Report.
    - **Non-Functional Requirements**
        - Scan should be completed in a feasible amount of time.
        - The system should accurately identify vulnerabilities with a low rate of false positives (Goal is 1% and under).
        - The application should be easy to use.
            - Minimal training should be required to use the threat assessment tool.
        - The report should be clear, concise and easy to understand.
            - There could be an option for more technical reports.
        - The scan should not consume a lot of resources and have a reliable execution time.
        - The app should be secure and follow the guidelines of the CIA triad ensuring that sensitive information is protected.
    - **Use cases**
        - Load in user API file via import.
        - Initiate API discovery.
        - Select testing profile
        - Execute API scan.
        - View / export Threat assessment report.

    - Register User.
    - Login User.
  - **Development Road map**
    - Next 2 weeks (Demo 1)
      - Core components / Proof of concept build.
      - Basic UI and back end with 3 working use cases.
        - Login, Register, API specification import.
      - Basic documentation and architecture diagrams.
      - Research report.
    - Documentation and Testing
      - Documentation will be created and updated throughout the development life cycle.
      - Stability and Quality will be ensured through both automated and manual testing.
    - Initial Project Road map
      - Define internal API specification with support for imports.
      - API scanning engine.
      - Security reporting.
      - Enhanced UX.
      - Improvement of scan accuracy and performance.
      - Expand vulnerability coverage.
      - Maintainability and App Security evaluation and remediation.
      - Advanced and WOW features.
- **Client Vision**
  - Overview and feedback.
  - Additional Requirements.
  - Additional Use Cases.
  - Technical review.
- **Communication**
  - Discuss how often client meetings should occur.
  - Preferred platform.
- Discuss what needs to be done for the next meeting.
- **Closing**

```
            AT-AT

                 ________
            _.-'::'\____`.
          ,'::::'  |,------.
         /::::'    ||`-..___;
        ::::'      ||   / ___\
        |::       _||  [ [___]]
        |:   __,-'  `-._\__._/
        :_,-\  \| |,-'_,. . `.
        | \  \  | |.-'_,-\ \   ~
        | |`._`-| |,-|    \ \    ~
        |_|`----| ||_|     \ \     ~              _
        [_]     |_|[_]     [[_]      ~        __(  )
        | |    [[_]| |     `| |        ~    _(   )   )
        |_|    `| ||_|      |_|          ~ (    ) ) ))
        [_]     | |[_]      [_]          (_       _))
       /___\    [ ] __\    /___\           (( \   ) )
jrei          /___\                        (     ) )
                                             (  #  )
```