User Acceptance Testing Plan for Network Traffic Profiler Dashboard

Outline

1. Scope

1.1. Objectives and business requirements

The goal of this user acceptance test is to figure out whether our user-base of non-technical experts can properly use our PCAP file analysing system with expected results by our AI model. (Goal)

We're planning to finish our part of the development and allow the project to be continued and finalised by future developers, and finally, to be integrated into the firewall.
(Accomplishment)

We will measure success by evaluating success times of the tasks to be performed, along with the accuracy of the AI from different PCAP file input.
(Success measurement)

1.2. Scope

Our development phase has been nearing completion, and we need to ensure that our system meets client expectations and is ready to be handed over to the client. (Objective)

For this UAT test, we'd like to:

- Verify that the pipeline works as expected (upload file -> run model -> display statistics)
- Get feedback from the team on how accurate the anomality prediction is
- ?

(What are we testing, what not?)

2. Testing team

In this section, list out members of your QA team and what their roles will be during UAT.

Name	**Responsibilities**
1	UAT Coordinator (handle communication between end users and QA team)
2	Design test cases
3	Create test data
4	Write UAT reports

3. Milestones and deliverables

This section contains all deliverables for successful UAT execution.

3.1. Design & testing process

[Link to designs (in Figma or other)]
[Link to wireframes]

Our system consists of a dashboard, where on the left panel the user is prompted to upload their PCAP file.
After the PCAP file is uploaded, the file runs through our trained AI model in the background.
Finally, the user gets shown a set of diagrams on the main panel on the right, displaying:

- a prediction about the anomality of the file
- values that deviate from the norm
- statistics about the file
- descriptions about each displayed diagram

(Designs)

The testing will occur in 4 stages:

1. Staging environment: set up by [name], this environment should closely mirror a real-world scenario in which our system would be used in.
2. Training: UAT testers will be trained by [name] which includes running the user through the system's functionalities and ML model
3. UAT execution: create test cases and have our tester/reporter report on said test cases.
4. Reporting: full data analysis, bug triage, and meeting on what remains to be done.

Deadline for design & testing process:

(How and when does the testing take place)

3.2. Staging environment

Our staging environment will be accessible for the UAT tester on our GitHub [link to GitHub].

Either install dependencies via the requirements.txt or install .exe file/pull the Docker container? Or run the system on one of our personal devices…

Deadline for staging environment:

(Requirements for the staging environment)

3.3. Training

We will be holding the UAT meeting with the user over Teams before meeting them in person, where we will walk them through our system's features, explain the objectives for the UAT and describe what we expect them to report on.

Deadline for training:

(Training for tester)

3.4. UAT Execution

Execution will take about 30 minutes in person. During these, we need to ensure the tester uploads a variety of different PCAP files and analyses their output as critically as possible.

Steps:

1) Onboarding. Onboard the user, help them set up staging, and explain what we expect of them (briefly touched on during training as well).
2) Test case execution. The user will be given specific test cases (see below), and report bugs, feedback and model accuracy.
3) Once done, record quick meeting with the user to get feedback on the experience that we can come back to during QA meeting.

Deadline for UAT execution:

(How and when the UAT takes place, from onboarding to report on the test cases)

3.5. Reporting & data analysis

Full analysis of individual test cases - understand what testers struggled with, what the general feedback is, and areas of improvement.

Deadline for reporting & data analysis:

4. Environmental requirements

4.1. Hardware requirements

Our system does not require special hardware to be run on.

4.2. Software requirements

The tester needs to have the following dependencies installed on their system:

- Python 3.11

5. Features to be tested

This section includes the features to be tested and will be referred to during the UAT.

5.1. Feature 1

5.1.1. Pass/fail criteria

Add a clear description of what the pass and fail criteria is for each feature.

Example:

- **Pass:** the system correctly identified the abnormalities within the PCAP file.
- **Fail:** the system couldn't identify the abnormalities within the PCAP file.

5.1.2. Test cases

Write step-by-step, detailed but concise instructions on how to test the feature.

Example:

1) Run the app through Streamlit on localhost.
2) Upload a PCAP file using the upload button.
3) The PCAP file runs through the trained model in under 10 seconds.
4) Check that the graphs are correctly displayed.

5.2. Feature 2

5.2.1. Pass/fail criteria

5.2.2. Test cases

5.3. Features to avoid testing

Avoid testers being sidetracked by specifying what features must be avoided during testing. This is particularly relevant if you're testing a lot of features at once, or if your software is complex enough that testers might not recognize that they're testing the wrong feature.

5.3.1. Feature 3
5.3.2. Feature 4

6. Signoff

I hereby accept this final product. (Yes/No)

_____

(Signature)

Client Name, Position, Organisation
Date: