User Acceptance Testing Plan for Network Traffic Profiler Dashboard

Outline

Outline 2
1. Scope 4
1.1. Objectives and business requirements 4
1.2. Scope 4
2. Testing team 5
3. Milestones and deliverables 6
3.1. Design & testing process 6
3.2. Staging environment 7
3.3. Training 7
3.4. UAT Execution 7
3.5. Reporting & data analysis 8
4. Environmental requirements 8
4.1. Hardware requirements 8
4.2. Software requirements 8
5. Features to be tested 8
5.1. PCAP file upload 8
5.1.1. Pass/fail criteria 8
5.1.2. Test cases 8
5.2. Action classification 9
5.2.1. Pass/fail criteria 9
5.2.2. Test cases 9
5.3. CSV file download 9
5.3.1. Pass/fail criteria 9
5.3.2. Test cases 9
5.4. Error handling 9
5.4.1. Pass/fail criteria 9
5.4.2. Test cases 9
b) Features to avoid testing 9
6. Signoff 10

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 intuitively and properly use our PCAP file analysing system, with our AI model correctly classifying the contained user actions within the file.

We're planning to finish our part of the development and leave the project to be continued and finalised by future developers, with the objective of being integrated into the firewall.

We will measure success by evaluating upload times and the accuracy of the AI classification, using different PCAP files for the input.

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.

For this UAT test, we'd like to:

- Verify that the pipeline works as expected (upload file -> run model -> display statistics)
- Evaluate the accuracy of the AI action classification

PCAP files that don't contain expected actions, such as Liking/Commenting on a YouTube video, cannot be classified by our AI model, but still display metadata on the dashboard.

2. Testing team

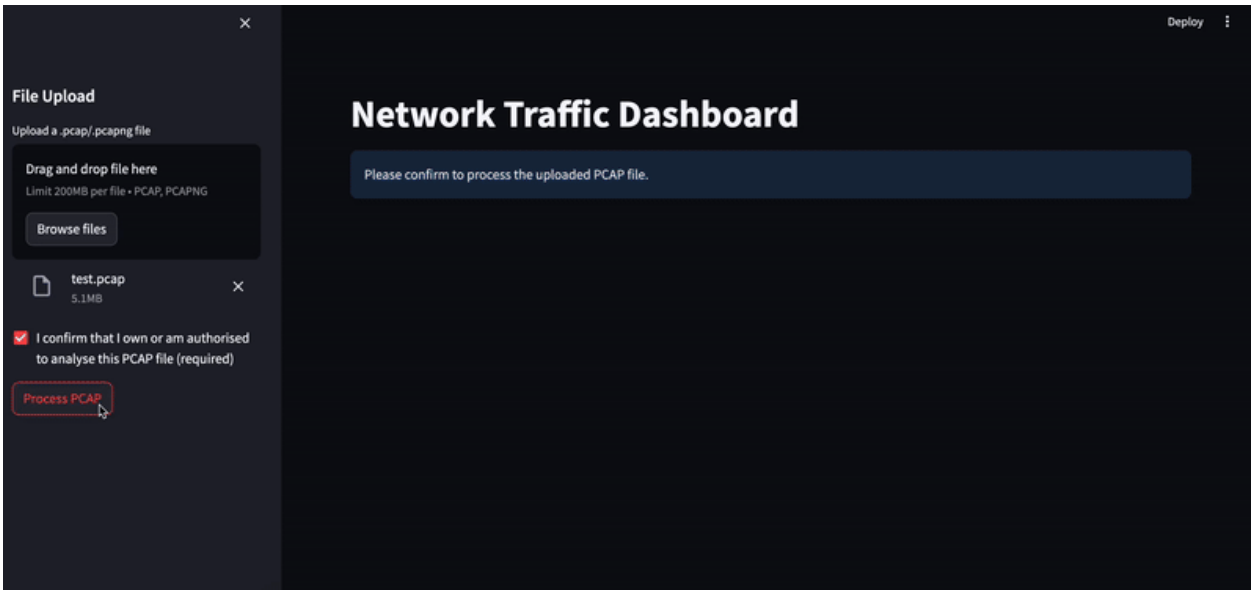
In this section, members of the QA team along with their roles during the UAT are listed.

Name	**Responsibilities**
Amelia	UAT Coordinator (handle communication between end users and QA team)
Sophia	Design test cases and UAT plan
Tim	Create test data
Tomi	Write UAT reports

3. Milestones and deliverables

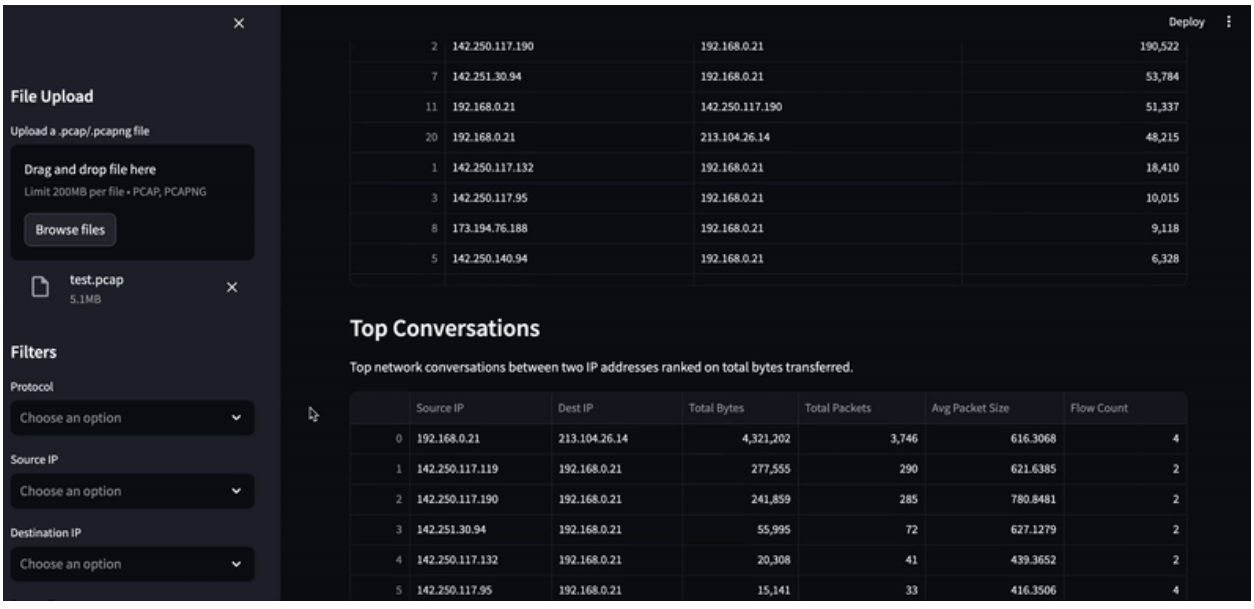
This section contains all deliverables for successful UAT execution.

3.1. Design & testing process



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 is shown a set of diagrams on the main panel on the right, displaying:

- Summary of the PCAP file activity
- Metadata about the file, such as traffic volume, top endpoints and conversations
- Segmented flows and their action classification

The testing will occur in 4 stages:

1. Staging environment: set up by our team at the client meeting, 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 Amelia which includes running the user through the system's functionalities
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: Sprint 5

3.2. Staging environment

Our staging environment will be accessible for the UAT tester on our team's GitHub repository. The tester will be asked in advance to install the requirements, or they can use one of our personal devices. PCAP files used for the tests will be provided in the repository's /pcap folder and recorded by one of our group members, using Wireshark.

Deadline for staging environment: Sprint 5 before the actual testing

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: Sprint 5 before the actual testing

3.4. UAT Execution

Execution will take about 30 minutes in person. During this, 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 (discussed during training)
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: Sprint 5

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: End of Sprint 5 to allow for improvements in Sprint 6

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
- Python libraries from requirements.txt
- PCAP files to upload

5. Features to be tested

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

5.1. PCAP file upload

5.1.1. Pass/fail criteria

- **Pass**: file finished uploading in under 15 seconds.
- **Fail**: file failed to upload or took longer than 15 seconds to upload.

5.1.2. Test cases

1. Run the app through Streamlit on localhost (streamlit run dashboard.py).
2. Upload a PCAP file using the upload button.
3. The dashboard returns visualised statistics.

5.2. Action classification

5.2.1. Pass/fail criteria

- **Pass:** the system correctly identified most actions (>80%) within the PCAP file.
- **Fail:** the system couldn't (correctly) identify the actions within the PCAP file.

5.2.2. Test cases

1. Gather PCAP file samples and insert them into the dashboard.
2. Compare with the expected result (e.g. PCAP file containing traffic of liking a YouTube video should be classified as such).

5.3. CSV file download

5.3.1. Pass/fail criteria

- **Pass**: CSV file gets downloaded to user's device on click and contains correct data.
- **Fail**: CSV file does not get downloaded and/or contains incorrect data.

5.3.2. Test cases

1. Upload PCAP file and let it go through the main pipeline.
2. Download the generated CSV file containing the cleaned and classified data.

5.4. Error handling

5.4.1. Pass/fail criteria

- **Pass**: All errors get handled correctly and non-allowed user actions are not permitted.
- **Fail**: Errors do not get corrected and non-allowed actions are bypassed.

5.4.2. Test cases

1. Try uploading a file that is over 200MB.
2. Try uploading a file that is not a .PCAP file.
3. Try uploading multiple .PCAP files.
4. Try bypassing the GDPR warning by not ticking it off.

b) Features to avoid testing

PCAP files that do not contain actions that our model has been trained to classify cannot be correctly classified and therefore should not be included in the testing.

6. Signoff

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

_____

(Signature)

Client Name, Position, Organisation
Date: