Improving the Air Traffic Attack Simulator

Edvin Dyremark
Linköping University
Department of Computer and Information Science
Linköping, Sweden
edvdy189@student.liu.se

Tobias Elfstrand
Linköping University
Department of Computer and Information Science
Linköping, Sweden
tobel190@student.liu.se

Rasmus Holmgren
Linköping University
Department of Computer and Information Science
Linköping, Sweden
rasho356@student.liu.se

Adrian Rosén
Linköping University
Department of Computer and Information Science
Linköping, Sweden
adrro045@student.liu.se

Wilmer Segerstedt
Linköping University
Department of Computer and Information Science
Linköping, Sweden
wilse150@student.liu.se

***Abstract*—This report presents enhancements to the Air Traffic Attack Simulator, originally developed on top of the OpenScope Air Traffic Control Simulator. The simulator exposes air traffic controllers to various cyberattack scenarios, helping them understand and handle potential threats. The project aimed to expand the simulator by integrating Swedish airports, introducing advanced attack scenarios, such as GPS jamming, and rouge drones, improving data visualization and analytics, and addressing existing bugs. The new attack scenarios introduce a new challenge for users, while the new analytics improved the analysis of the test results. Bug fixes and refactoring of the admin page improved the quality and maintainability of the simulator's code. Additionally, educational content in the form of tutorial videos for the new attack scenarios was developed. Together, all these updates are a significant improvement to the Air Traffic Attack Simulator.**

***Index Terms*—Air Traffic Controller, Cybersecurity, GPS Jamming, Visual Analytics, Drone intrusion, OpenScope, Attack Simulator**

I. Introduction

This project is based on a previous project where the Air Traffic Attack Simulator was developed on top of the OpenScope Air Traffic Control Simulator [1]. The aim of the previous project was to increase the cyber awareness of Air Traffic Controllers by exposing them to different types of cyber attacks. To increase the potential of the simulator it would be beneficial to add more environments, richer cyberattack scenarios, and refined analytics to aid in evaluating the performance of the users and the simulator. It would also be valuable to reduce the number of bugs in the simulator and develop educational content related to the simulator. This report discusses these improvements and how they were implemented. The simulator is currently used in an experimental phase on air traffic controllers in Sweden to improve their capability to handle cyber attacks. Integrating a Swedish airport would greatly improve the familiarity the air traffic controllers feel with the airport. Some cyber attacks are missing from the roster of attacks that currently exist in the attack simulator, adding these would ensure the traffic controllers get even more relevant training. As part of the experimental phase, it is crucial that statistics are gathered and the administrators of the tests can visualize the results of the tests enabling better analysis.

A. Research questions

The objectives of the project were formulated into the following research questions:

- How can Swedish airports be added to the attack simulator?
- How can new attacks be added to enhance the realism and educational value of the simulator?
- What sophisticated analytics can allow administrators to effectively review and analyze user performance?
- How can the simulator be improved by identifying and rectifying potential flaws and bugs?
- How can cyber awareness be strengthened by developing educational content?

II. Background

This section contains the necessary background information for successfully answering the research questions.

A. Air Traffic Attack Simulator

The Air Traffic Attack Simulator was developed for the reasons mentioned in Section I. To successfully create cyberattack scenarios, the OpenScope ATC simulator was determined as adequate and chosen as the foundation for the Attack

Simulator [1]. The simulator is web-based, mainly written in JavaScript, and runs in a browser. Air Traffic Controllers can using the simulator train to handle complex scenarios for different airports. A score is calculated based on the controller's actions. A positive score is acquired for actions such as successfully landing an airplane, whereas a negative score may be given for actions such as airspace bust, i.e., when an aircraft is not able to land and leaves the airspace. On top of this simulator, the cyber attack component has been added, which allows the Air Traffic Controllers to train on handling different attacks on the communication infrastructure of aviation. Scores are also given for actions related to cyber attack scenarios. Positive scores are given for attacked airplanes successfully identified, and negative scores are given if the attack goes unnoticed. The extended simulator supports different preset test scenarios, which inject attacks of various types with different frequencies. These attacked aircraft can be visualized with one color for each type of attack, as seen in Figure 1. However, during a test, the attacked aircraft is typically not highlighted with colors to force the user to reason about what type of attack affects the aircraft.

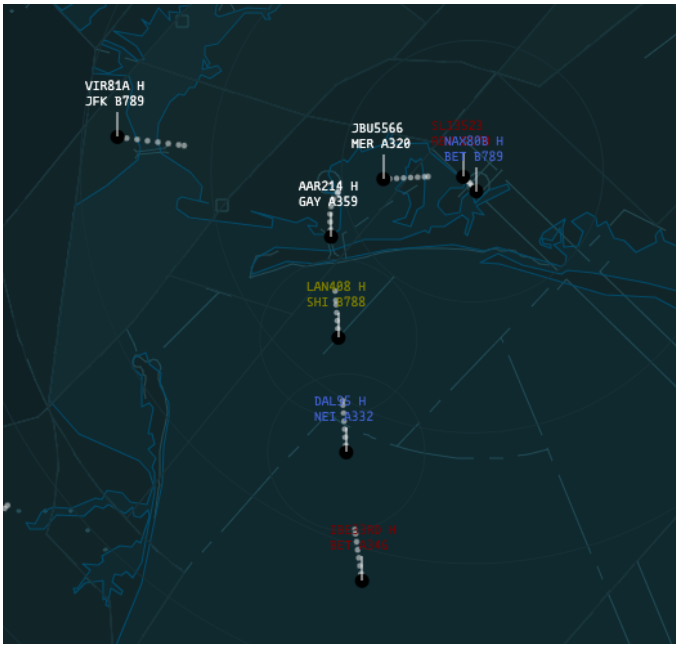

Fig. 1. Image showing the Air Traffic Attack Simulator with attacked aircraft highlighted with colors.

An admin page has been added to the OpenScope Attack Simulator. This page helps test leaders evaluate the performance of the users, as well as the performance of the extended simulator itself. Currently, the admin page contains a table with an entry of each conducted test. For each test, there is a sub-page with additional tables with detailed information for the specific test.

III. Research Methodology

A. Integrating Swedish airports

Preliminary research was conducted to determine an airport of a suitable size to be included in the Air Traffic Attack Simulator. Smaller airports are easier to implement, but may not yield substantial benefits for learning and assessment. However, larger airports provide excellent learning and evaluation opportunities, but their implementation can require more time and resources than what is available within the project's timeline. Initially, a suitable airport is found, and then a combination of looking at the repository's README files and examples of previously defined airports in the repository are used to figure out the structure. Relevant data for defining the Swedish airport can be accessed at AROWeb.

B. Implementation of advanced analytics

The admin page has been reworked to provide easy-to-read analytics for user performance, both overall and on a per-test-basis. This was done by using visualization tools such as graphs and pie charts. Furthermore, the admin page was expanded with additional analytics and information. Another feature that was added was the ability to select a number of tests and display information about them simultaneously.

The following is a list of the analytics that have been implemented:

- A bar diagram that compares the scores of different users.
- A pie chart displaying the percentages and frequencies of different attacks in a test.
- A pie chart displaying the percentages of correctly identified cyber attacks, together with the correct/incorrect actions for each attack.
- A bar diagram displaying the score of a user at different time intervals. The score will be divided into four categories, where the score for actions related to air traffic control and actions related to cyber attacks will be separated into their respective positive and negative categories.
- A diagram displaying at what time during the test the cyber attacks were injected into the environment.

The existing code base is first studied to determine if the analytics can be implemented right away, or if it requires any refactoring of the existing code.

The admin pages and sub-pages provide much information about each test in a table format which is hard to read, especially when looking for a quick overview. For example, take a look at the table for event logs in the sub-page for each test which can be seen in Figure 2.

However, due to the large amount of data contained in these tables and logs, it can be challenging for a supervisor to gain an overview of tester performance. To make it easier to visualize the test outcomes, graphs can be used to represent various cyber attacks and their frequencies. Additionally, to better understand the testees' performance, pie charts can illustrate the number of correct versus incorrect guesses made by the user.

The current admin page for the simulator is built in JavaScript. Chart.js is a widely used open-source library for JavaScript,

Timestamp	Test ID	Event Type	Message	Score	Total Score
6.8	1	UI_LOG	Test started, hard, dyre, Test ID: 1	0	0
7.1	1	TRAFFIC_SETTING_CHANGE	The traffic setting departure was changed from 1 to 4.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting probabilityOfNewAircraft was changed from 0 to 0.6.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting nonResponsiveProbability was changed from 0 to 0.1.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting falseInformationProbability was changed from 0 to 0.15.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting impersoniteProbability was changed from 0 to 0.1.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting transponderCodeAlterProbability was changed from 0 to 0.15.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting duplicateProbability was changed from 0 to 0.15.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting nonMovingProbability was changed from 0 to 0.15.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting virtualTrajectoryAlterProbability was changed from 0 to 0.1.	0	0
7.1	1	ATTACK_SETTING_UPDATE	The attack setting positionAlterProbability was changed from 0 to 0.1.	0	0
7.1	1	TRAFFIC_SETTING_CHANGE	The traffic setting windDirection was changed from 280 to 60.	0	0
7.1	1	TRAFFIC_SETTING_CHANGE	The traffic setting windSpeed was changed from 4 to 20.	0	0
7.1	1	TRAFFIC_SETTING_CHANGE	The traffic setting arrival was changed from 1 to 4.	0	0

Fig. 2. Event logs in table format for a selected test

designed to simplify the creation of charts and graphs for web applications [8]. This project is used to create the charts explained above. Figures of the actual charts can be found in Section IV-C.

C. Identifying bugs and refining the simulator

To ensure effective use of the simulator, it was essential that all team members initially became proficient in its operation. During the use of the simulator, any issues identified by the group as potential bugs were systematically documented. As development progressed, team members selected and addressed these previously identified bugs, thus improving the overall quality of the Air Traffic Attack Simulator. Furthermore, any new features introduced by the group were rigorously tested for potential bugs to maintain the simulator's high standard of quality.

D. Integration of new cyber attacks

Two new cyber attacks were to be implemented, an intruding rogue drone and GPS jamming.

1) Intruding drone attack: Drones are becoming increasingly more common, leading to a rise in incidents where they unintentionally or maliciously enter areas with restricted airspace [2]. The planned scenario simulates a cyber attack where a rogue drone breaches the airport's controlled airspace, creating immediate safety risks, and highlighting the need for rapid response and countermeasures to maintain safety and operational integrity. Initial research was carried out to identify the procedures airports implement to ensure passenger safety and maintain operational continuity in the event of drone intrusions. In most cases, the airports choose to ground all flights in the area but this cyber attack simulation will focus on the few cases when they only choose to restrict a certain area around where the drone has been spotted.

To extend the attack simulator with an intruding drone scenario, a distinct drone icon and a red circle to represent the surrounding restricted airspace will be added. The drone can follow a predefined path or remain stationary, allowing for various scenarios.

The point system will impose a penalty for any aircraft that enters the restricted area. Positive points will only be awarded for aircraft that were originally scheduled to route through the restricted area but successfully rerouted around it. This encourages strategic decision-making in response to the drone threat.

2) GPS Jamming: In recent years, external interference as a vulnerability of the Global Positioning System (GPS) and the Global Navigation Satellite System (GNSS) has gained significant attention, especially in safety-critical applications like aviation due to the effects of e.g. ongoing wars. Among these vulnerabilities are GPS jamming and spoofing. GPS jamming involves transmitting signals at the same frequency as GPS signals to prevent accurate positioning, while GPS spoofing transmits false signals to mislead the receiver into calculating an incorrect position controlled by the attacker.

As part of this project, we explored various GPS jamming techniques to understand their impact on an aircraft's positioning systems. Specifically, GPS jamming, which blocks or distorts the GPS signal, was researched as it is more widespread due to it being trivial to implement. The effects of GPS jamming on aircraft positioning were informed by prior research in this field, namely in [3] and [4]. The final implementation was based on a simplified approach primarily inspired by [4].

E. Development of educational content

Additional informative educational content was created to strengthen cyber awareness training. This included several video tutorials showcasing the additional features that were added to the simulator. Along with the educational videos, a document explaining the videos and the functionality of the attacks was created.

IV. Results & Discussion

The following subsections give an overview of the results. Each subsection pertains to a specific research question outlined in Section I-A.

A. Integrating Swedish airports

Gathering the data for the Swedish airports was mostly through AROWeb [5]. AROWeb is a system created by the Swedish Civil Aviation agency, Luftfartsverket. What is needed for adding a new airport in the simulator is defined in the project documentation [6]. There are two levels of airports that are defined in the OpenScope simulator, the "standard" level and the "premium" level [7]. For it to be a premium airport, the following criteria have to be met:

- Relevant airways.
- Full documentation on facility's airspace.
- Airspace and traffic flow diagrams.
- Accurate airspace stratification.
- Exceptionally high realism.

These items will not be taken into consideration when creating the first Swedish airport, the reasoning for this is to create a minimal working version as a first step. In the future if more resources is allocated the standard airport could later be upgraded into a premium version.

Using the guide on how the airports should be formatted and AROWeb as a basis where most information could be found. The information that was either missing or not stated clearly enough for someone without domain-specific knowledge was the following

1) Standard Instrument Departure Route (sid)
2) airways around the airport
3) The map

The map is specific to the simulator and is usually created using ARTCC (Air Route Traffic Control Centers) on VATSIM (Virtual Air Traffic Simulation Network) as the foundation. Most European airports are not covered within this network and only the bigger ones such as potentially Arlanda. Since Arlanda is Sweden's largest airport integrating Arlanda was seen as too big of a scope to cover within the limits of this course. Instead focus was spent on working on the Malmö airport. Because of the mentioned missing information the result of this work was that malmö airport could not be added to the simulator. The openscope slack group was contacted to try to find some missing gaps in this information, but to no success.

In an older version of OpenScope, Luleå airport was integrated and working, everything was refactored 5 years ago meaning some data is missing or formatted in the wrong way for the airport to work with the refactored version of OpenScope. Some effort was directed toward getting this airport working since that could then be used as a foundation on how to integrate other Swedish airports later. The result of the work done on Luleå airport was unfortunately not successful, which means that the airport did not work on the simulator.

In parallel to this task, a guide on how to add Swedish airport and how to apply the information found in AROWeb to the OpenScope format a guide was to be created. Since there was no success in implementing a Swedish airport this guide will also be incomplete, this could help some information gathering in the future and contains some explanations for what different keywords mean. This is the main result of the sub-task to the research question.

B. Integration of new cyber attacks

1) Intruding drone attack: A new intruding drone attack was implemented in the Air Traffic Attack Simulator to simulate a rouge drone breaching an airport's controlled airspace. The attack aims to mimic a real world scenario where a drone, potentially operated with malicious intent, enters the restricted area surrounding an airport and thereby forcing air traffic controllers to react quickly to ensure passenger safety and operational continuity.

The attack is visualized in the simulator as a distinct drone icon with a surrounding red circle appearing on the map, as seen in Figure 3. The circle indicates the area that aircraft must avoid and stay clear of. As previously specified, points will be deducted for any aircraft that enters the restricted area, and points

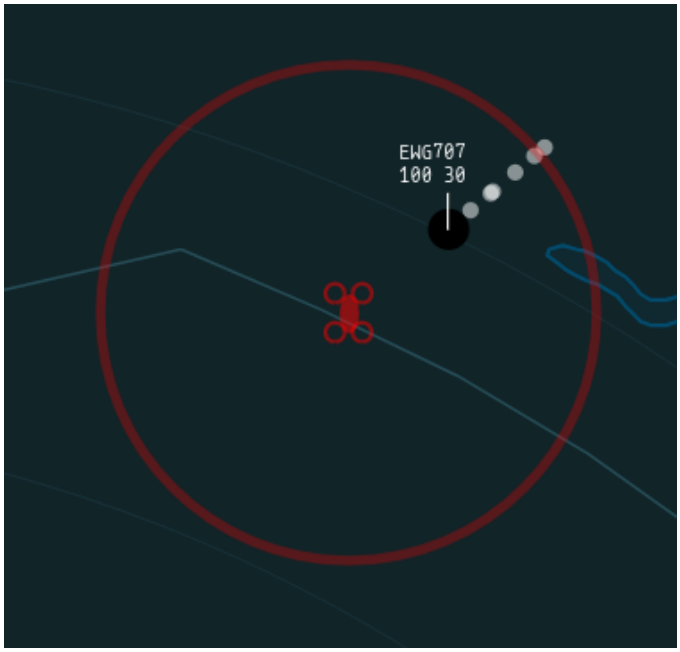

Fig. 3. Image displaying an aircraft entering the restricted drone zone in the attack simulator.

will be awarded for any aircraft that were originally scheduled to route through the restricted area that are successfully re-routed by the user. In its current implementation, 50 points are awarded for successfully re-routing aircraft and -500 are given as a penalty for aircraft that enter the restricted drone zone. Unlike previous attacks, this attack does not directly impact aircraft. Instead, it solely awards or deducts points based on user action.

Fig. 4. Image displaying moving drone zone spawn point with North-West heading.

Fig. 5. Image displaying aircraft affected by drone attack, colored in aqua blue.

The drone attack is categorized into two modes: a stationary drone attack mode, illustrated in Figure 3, and a moving drone attack mode, as seen in Figure 4. Both attack modes, like previous attacks, target a single aircraft. However, the effects of the attack extend to all aircraft. In Figure 5, the aircraft targeted by the drone attack is highlighted in aqua blue. In its current implementation, only one intruding drone attack can occur at a time. This limitation is in place primarily because managing multiple simultaneous drone attacks has proven to be challenging.

The stationary attack mode's spawn location is determined by analyzing the waypoints in the aircraft's aircraftModel to identify a suitable waypoint near the airport center. The attack controller iterates through the waypoints, evaluating their positions relative to the airport center, and selects the closest waypoint that meets the minimum distance requirement as the target location. This ensures that the stationary drone spawn location is placed near the airport center and allows the attack to impact multiple aircraft, as several aircraft often share the same waypoint routes.

The spawn location of the drone in the moving attack mode is determined by the drone attack heading setting, accessible in the drone attack options menu shown in Figure 8. This setting can be set to one of the four directions, North-West, North-East, South-West, and South-East or be set to random. If set to random, one of the four directions is chosen at random. The drone spawns in the corner opposite the flight heading and travels in the selected direction, crossing the airport center. In this mode, no points are awarded for re-routing aircraft, however, points are deducted for any aircraft entering the moving drone zone.

Regardless of the attack mode, the drone attack ends once the targeted aircraft has arrived or left the airspace. The duration of which the attack remains active after the targeted aircraft has disappeared can be adjusted in the attack settings.

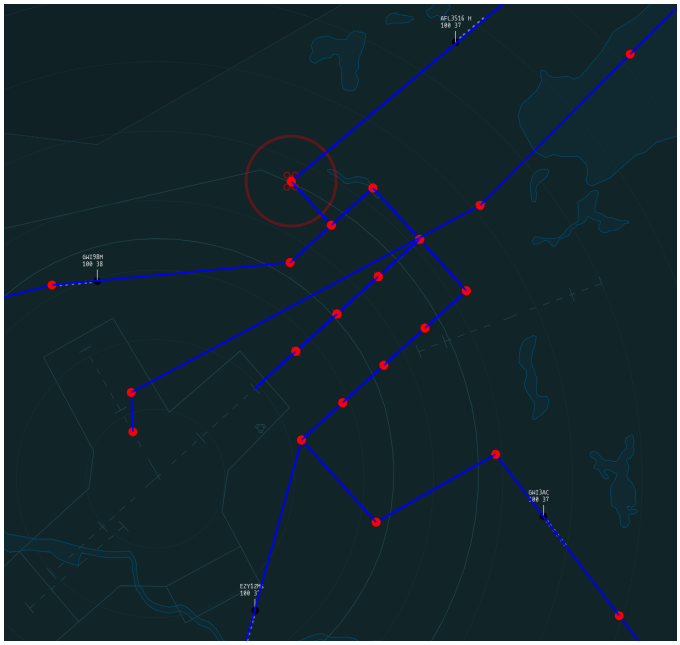

Fig. 6. Image showing aircraft routes passing through the restricted drone zone in the attack simulator, with the aircraft waypoint debug function enabled. Red dots represent waypoints, while blue lines connect the waypoints to illustrate the flight path.

Significant effort was put into determining which aircraft would pass through the drone zone. Aircraft in the attack simulator follow predefined waypoint-based paths when entering controlled airport airspace. This waypoint data was extracted and visualized by developing a debugging function that draws the waypoints on the map, as seen in Figure 6. While not part of the drone attack, this debugging feature was helpful in understanding the waypoint structure and ensuring accurate detection. This was important to make certain that points were only awarded for aircraft scheduled to route through the drone zone, avoiding false positives.

In practice, this was implemented in the intruding drone attack by iterating through the AircraftModels and extracting the waypoint lists for each simulated aircraft. These lists are used to draw lines between adjacent waypoints, which are then checked for intersections with the drone zone circle. Improving this implementation required considerable effort, as initial results were imprecise and did not align correctly with what was actually projected on the map.

The intruding drone attack can be easily activated or deactivated by adjusting its probability in the attack settings menu, depicted in Figure 7. It is fully integrated into the attack framework and functions seamlessly, just like any other attack in the attack simulator.

The attack is user-configurable, with several parameters that can be adjusted to meet specific requirements. These parameters include the probability of the moving drone attack mode,

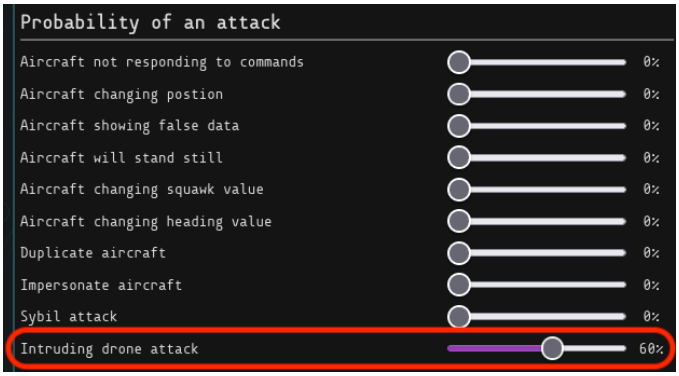

Fig. 7. Image highlighting the drone attack probability setting in the attack settings menu.

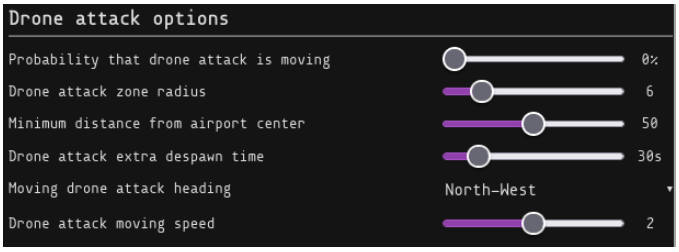

Fig. 8. Image displaying the drone attack option settings in the attack settings menu.

the drone attack zone radius, the minimum distance from airport center, additional de-spawn time, the flight heading of the moving attack mode, and the speed of the moving drone. These options, as illustrated in Figure 8, allow users to modify various aspects of the attack.

The moving drone attack probability setting determines the likelihood of the drone attack being stationary or moving. If set to maximum, only moving drone attacks will occur and if set to minimum, only stationary attacks will take place. The drone attack zone radius can be adjusted to increase or decrease the size of the restricted area in the attack. The minimum distance from the airport center setting specifies how close the drone attack can spawn to the airport center, which is particularly useful when adapting the simulation to different airports.

The extra de-spawn time option allows users to extend the duration of the attack after the targeted aircraft has arrived or departed the airspace. The flight heading of the moving attack mode can be customized to one of the following directions, North-West, North-East, South-West, South-East, or set to random, where one of the four directions is chosen randomly. Finally, the moving speed of the drone can be adjusted to increase or decrease its speed during the moving attack mode.

2) GPS Jamming: Two GPS jamming attacks were implemented in the simulator, namely a high-powered and a low-powered GPS jamming attack. These attacks can be similar to the already existing ADS-B false positioning information and the ADS-B standing still attack. To separate the GPS jamming attacks, a setting was added to choose whether to make the jamming zones visible in the map. Specifically, the low-powered attack is then represented on the map as a yellow circle, see Figure 9, and the high-powered attack as a red circle, see Figure 10. The color difference underscores the difference in intensity and impact of the two attacks.

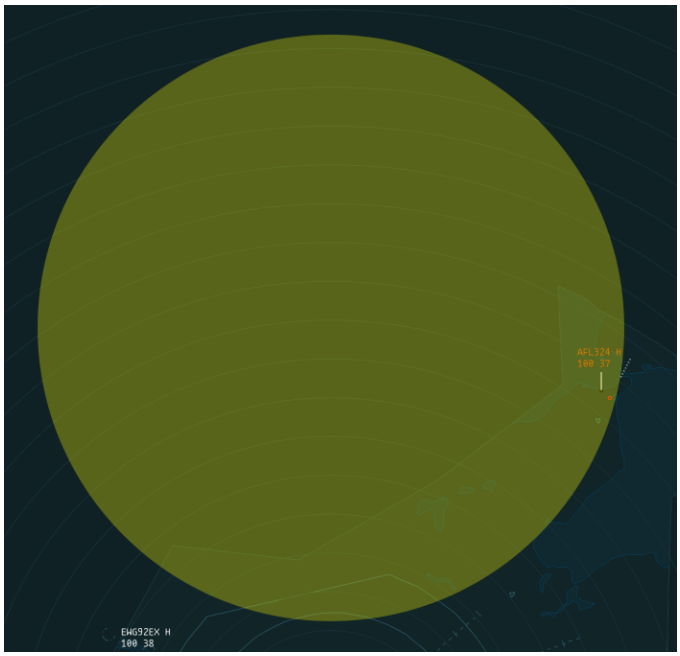

Fig. 9. Image showing low-powered GPS jamming zone in OpenScope.

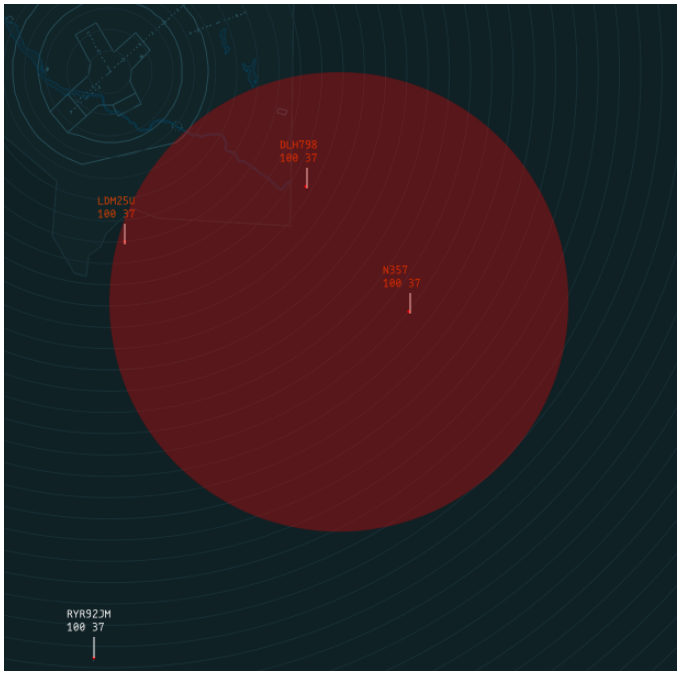

Fig. 10. Image showing high-powered GPS jamming zone in OpenScope.

The jamming attacks affect the positioning of the airplane by using a simplified version of what is described in [4],

to simulate realistic failure scenarios. Under the low-powered attack, the reported GPS-position of the airplane has an error of a few kilometers, resulting in a deviation from the true position. In the high-powered attack, the GPS receiver in the aircraft will not receive a new position during the entire time that the true position of the aircraft is within the red circle. Instead, the last updated position of the airplane will be shown, i.e. where the aircraft entered the jamming zone. This false position is stationary while the aircraft is within the jamming zone and is continuously transmitted to the air traffic controller. Once the aircraft is outside the jamming zone, it will receive a GPS update with a true position and hence send the true position again.

Figure 11 further illustrates how the low-powered attack affects the aircraft's visible position. In this figure, the small red circle shows the aircraft's actual position. The position that is visible to the user in the simulator, i.e. what the aircraft thinks is its true position, is indicated by the black circle. The erroneous position is updated every second to give the perception of positioning faults in the aircraft due to GPS jamming.

Fig. 11. Image showing low-powered GPS jamming zone in OpenScope with the aircraft's real position as a small red circle.

The user is expected to recognize that a GPS-jamming attack is taking place by making a guess and receiving points if they correctly identified the attack as a GPS-jamming attack. If the user does not guess or guess incorrectly, they will receive a penalty of -100 points. Further, points are deducted if they fail to identify the attack as a GPS jamming attack. The two attacks further complicate the work of the user by not allowing them to see the aircraft's true position. This affects multiple aspects such as routing, landing and making sure the different aircraft keep a regulated distance between them to avoid collisions.

In addition to the two GPS-jamming attacks, configurable settings were added in the attack settings menu, see Figure 12. These settings include the ability to toggle the attacks on and off. Additionally, a setting for whether the aircraft's true position is shown on the map during GPS-jamming was added. The later setting is useful for showing the different effects the two attacks have on the aircraft's positioning. Further, settings for how many low-powered and high-powered attacks appear were added, allowing the test leader to tailor the attacks for different scenarios. The test leader can also change the radius of the low-powered and high-powered attacks independently with the settings in the menu. Finally, a setting for whether the jamming zones are visible in the map for the user was added. This setting allows the test leader to show the effects of the different zones and can allow the user to easily differentiate between the previously mentioned ADS-B attacks. When toggling the attacks on or off, the positions of the jamming zones are randomized and the number of zones and their extent is taken from the settings menu. However, this randomization can be improved to take map parameters into consideration, such as where the landing strips are located, how large the airspace is, etc. This would make the GPS-jamming attacks more tailored to the map the user is being trained on since all incoming traffic might be coming from a particular place on the map. Then, placing the zones near this point would give the user more of a challenge then if the zones were placed far outside of the map. The attacks can also be improved further by randomizing when they appear, instead of the test leader having to toggle the attacks on or off. This approach would enable more realistic scenarios where the user remains unaware of when GPS-jamming occurs, creating a need for them to respond in real-time on their own accord. By removing prior knowledge of when the attacks are active, the scenario better simulates real-world conditions and enhances the evaluation of user responses.

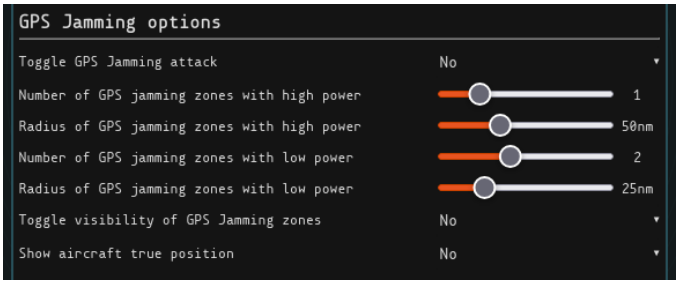

Fig. 12. Image showing the settings for GPS jamming attack in the simulator.

C. Implementation of advanced analytics

The admin main page has been expanded with functionality which allows the user to select a number of tests to compare against each other. The scores for each user is displayed on the y-axis, with the user id's of selected tests on the x-axis, see Figure 13.

Scores for the selected test will then be shown in a bar graph which gives a clear overview. See Figure 14 for an example.

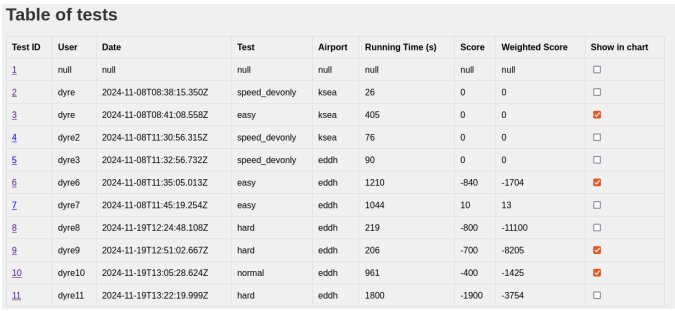

Table of tests

Test ID	User	Date	Test	Airport	Running Time (s)	Score	Weighted Score	Show in chart
1	null	null	null	null	null	null	null	☐
2	dyre	2024-11-08T08:38:15.350Z	speed_devonly	ksea	26	0	0	☐
3	dyre	2024-11-08T08:41:08.558Z	easy	ksea	405	0	0	☑
4	dyre2	2024-11-08T11:30:56.315Z	speed_devonly	ksea	76	0	0	☐
5	dyre3	2024-11-08T11:32:56.732Z	speed_devonly	eddh	90	0	0	☐
6	dyre6	2024-11-08T11:35:05.013Z	easy	eddh	1210	-840	-1704	☑
7	dyre7	2024-11-08T11:45:19.254Z	easy	eddh	1044	10	13	☐
8	dyre8	2024-11-19T12:24:48.108Z	hard	eddh	219	-800	-11100	☐
9	dyre9	2024-11-19T12:51:02.667Z	hard	eddh	206	-700	-8205	☑
10	dyre10	2024-11-19T13:05:28.624Z	normal	eddh	961	-400	-1425	☑
11	dyre11	2024-11-19T13:22:19.999Z	hard	eddh	1800	-1900	-3754	☐

Fig. 13. Test selection for graph comparison

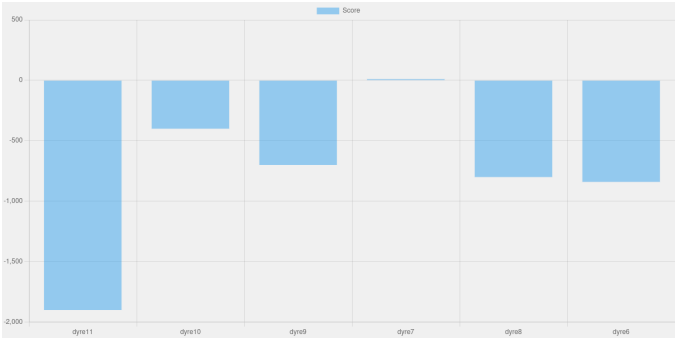

Fig. 14. Admin page bar-chart

The sub-pages for selected tests have been expanded with two pie charts. One shows what attacks occurred during the test and their frequencies, which can be seen in Figure 15. The other shows user guesses for these attacks in three categories, Correct, Incorrect or No-Guess, displayed in Figure 16.

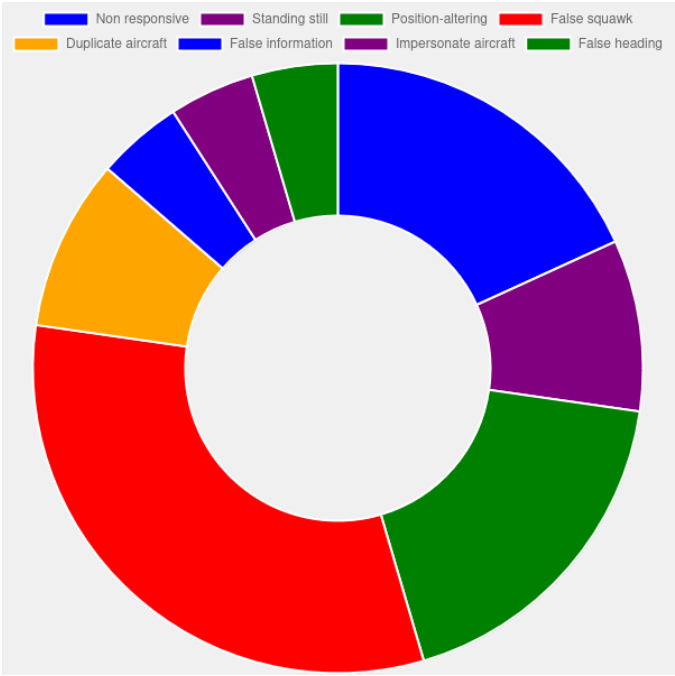

Fig. 15. Pie-chart displaying cyber attacks & their frequencies

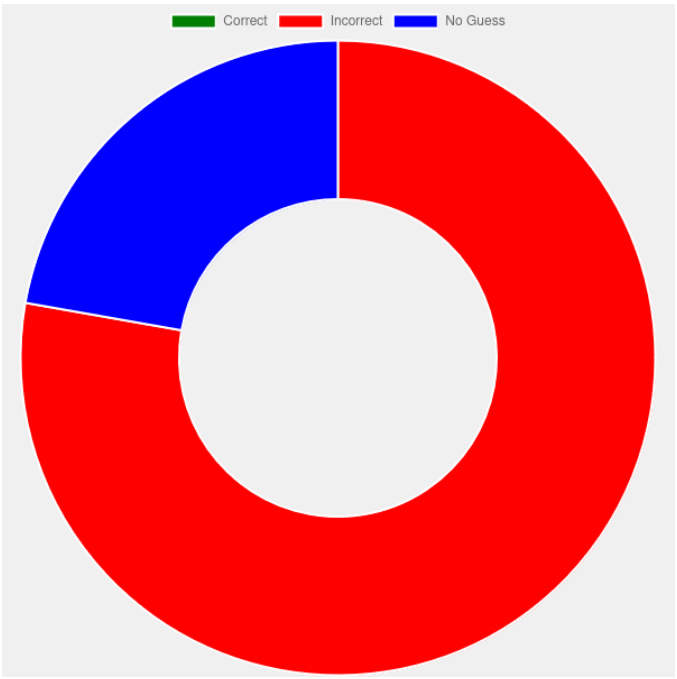

Fig. 16. User guesses on cyber attacks

The admin sub-pages have also been expanded with graphs to show what score the user has been awarded at different times throughout the test, with four categories. Positive/Negative for air traffic management and Positive/Negative for cyber attacks, see Figure 17.

Finally, the number of cyber attacks injected into the simulator at different intervals has also been added to the test sub-page and can be seen in Figure 18.

Fig. 17. User Score at different time intervals

After some investigation, it was determined that the existing code for the admin page and test sub-pages would need some refactoring to allow the addition of the new analytics in a good way. This refactor resulted in more modular code, to increase the overall maintainability, with separate files for the HTML and the JavaScript code. The charts were implemented using the selected open-source Chart.js library [8].

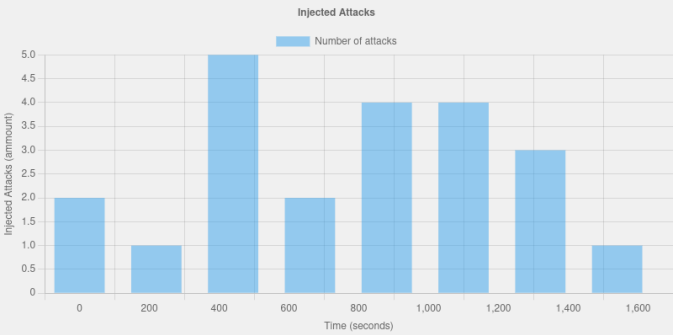

Fig. 18. Number of cyber attacks injected at different intervals

D. Identifying bugs and refining the simulator

When working with the simulator and testing new features, quite a few bugs have been discovered in the simulator. In order to create a better user experience and a fully functional program these bugs have been investigated and largely fixed by the project group. The group used a two-step plan to organize this work. When a bug was first observed, it was saved in a shared document and continuously updated over the course of the project. These identified bugs were then added to GitLab as issues with the bug fix label, and assigned to a group member. With this approach, most bugs were investigated and eventually fixed.

Some bugs were found to be occurring since the open scope commit that the attack simulator is forked from is older and the attack simulator has not been rebased on the newest version. This meant that some bugs existed in the attack simulator but not in the OpenScope simulator.

One such bug that existed in the attack simulator but not in the OpenScope simulator was that the tower-controlled departure was not working. This was because a working implementation existed in the OpenScope simulator but the attack simulator's implementation did not change behavior based on that setting at all. This issue was fixed by updating the aircraft controller to check if this specific option was set. If so, the program was updated to automatically run all commands required for takeoff on departing planes upon spawning.

Some reported bugs were determined to be false positives, specifically those related to attack probabilities. Increasing the time-warp setting to speed up the simulator, running multiple test sessions, and using the newly created visual analytics for the test helped with debugging. It was found that some guessed bugs were actually not bugs, and the simulator was working as expected.

One example was two attacks that never occurred in the tests, namely:

- Sybil
- Airplane Flooding

After thorough testing and investigation, it was found that the probabilities for these attacks were not configured in the test settings, which defaulted to 0, preventing the attacks from happening. Adding probabilities for these attacks in the test configurations would result in the attacks occurring as expected, and thus the issue was closed.

Another issue falsely identified as a bug involved an attack probability setting in the user interface. In a similar manner to the bug above, this was likely reported as working incorrectly when changing the settings in a test scenario. After some experimentation in the free-play mode, the probabilities turned out to be working as intended. Additional testing was performed in the test modes. It was found that starting tests automatically reset the selected attack probabilities to the defaults of the selected test. However, increasing the attack setting probabilities after the test had started seemed to work as expected. It was also found that some attacks were happening more often than others even with the same probabilities. This was due to hard-coded injected attacks at different timestamps in the test configurations. Eventually, these functionalities were considered indented behavior, and as such this issue was closed without further investigation.

One bug that was remedied was error logs that occurred on the admin page. The error logs were related to the function allowing filtering of tests but did not affect any functionality of the admin page.

A bug related to the penalty received for missed guesses was found and fixed. The user only received a penalty for missing to guess on an attacked airplane when the airplane was removed from the simulator, such as when leaving the airspace. However, it was also expected that the user would receive a penalty for missing to guess on an attacked airplane when a test ends and the airplane is still in the airspace. It was decided to also add a threshold of 8 minutes without a guess before giving the penalty. This assures that if an airplane leaves the airspace within 8 minutes after the start of the attack, it will not result in the user receiving a penalty.

An additional bug related to users guesses on airplanes was that users could guess on uncontrollable airplanes outside the airspace. The expected behavior was that users should not be allowed to guess on these airplanes. The bug was fixed by additional checks and an error message when a user attempted to guess on airplanes outside the airspace.

Most other bugs were related to commands being available or working even in unexpected situations. For example, it was possible for users to make a guess on aircraft outside the airspace. This was not the expected behavior and it was fixed by restricting guesses on aircraft outside the controllable airspace. Another similar bug was the `stoptest` command being accepted even if no test was currently running. This was fixed with a simple check and an error message being displayed to the user when attempting the command if no test had been started.

E. Development of educational content

The OpenScope tutorial was extended to include help for configuring GPS-jamming attacks. Further, the attack descriptions in OpenScope were extended to explain how the GPS-jamming attacks affect the positioning of aircraft visible to the user. Additionally, educational videos were produced showcasing the GPS-jamming attacks and their effects as well as how the intruding drone behaves. These videos can be used in training sessions with air traffic controllers to further their knowledge of modern-day attacks on air traffic. A document explaining the videos and how the attacks behave was also created. This document can be used by both the test leader to prepare for user testing, as well as the user to further their knowledge of GPS jamming and how aircraft behave when exposed.

V. Conclusions

The conclusions of each research question are described in the subsections below.

A. Integrating Swedish airports

Integrating a Swedish airport is possible through the work of a team with experience in both how json files work but also experience in the airplane/airport/airsim community.

B. Integration of new cyber attacks

The newly added cyber attacks further improve the simulator's ability to train air traffic controllers in their knowledge of cyber attacks. With intruding drones and GPS-jamming becoming more prevalent in the world, the knowledge of how to handle these two attacks are an important tool in the controllers toolbox.

To further improve these attacks, the GPS-jamming should be configurable to be enabled randomly instead of a test leader having to manually enable and disable the attack. Tailoring the GPS-jamming attacks to the currently used map would also further improve the educational value of the simulator. If the attacks were tailored to the maps in the simulator, the air traffic controller being trained would be subject to a more realistic scenario than randomly appearing GPS-jamming zones. Another type of GPS-jamming mode can also be added that combines the two existing methods to try to more accurately mimic real world scenarios since there exists no real hard line between *low-powered* and *high-powered* attacks. Real world GPS-jamming attacks can vary largely in their respective power, creating more complex scenarios where aircraft behave differently than in the simulator.

Additionally, the intruding drone attack could be enhanced by allowing users to pre-determine the drone zone location either by dragging it on the map or by entering coordinates in the settings menu. This would allow for further customization of the attack. The moving drone zone could also be improved to calculate aircraft trajectories and analyze wether they intersect, enabling the awarding of points for aircraft that re-route during the moving attack mode. Furthermore, expanding the range of directions in which the moving drone attack can travel would enhance the overall customizability of the attack, enabling a wider variety of attack scenarios.

C. Implementation of advanced analytics

The test analysis was improved with the help of visual analytics in the form of pie charts and bar charts. As a result of the implementation process, the overall maintainability of the admin page was improved.

Future work expanding on these results would be refactoring of the tutorial, and feedback page to also improve on their maintainability, similar to what have been done for the admin page. After evaluation with real air traffic controllers, the analytics can possibly be extended with additional visuals as needed.

D. Identifying bugs and refining the simulator

Some bugs were fixed during the course of this project and they will be summarized in the list below.

- Airplanes not automatically departing with the setting "automatically controlled takeoff on".
- Stoptest command can be executed even when no test is active.
- Attack probability setting not working as intended.
- Some attacks never appear.
- Error logs related to the search of tests on the admin page.
- A user did not receive a penalty at the end of the test when missing to guess on an attacked airplane.
- A user could guess on an uncontrollable airplane outside the airspace.

Recommendation for future work in this area would be to spend the effort to rebase the attack simulator on the most recent OpenScope release. This could lead to fewer bugs in the simulator but it will also be easier in the future when OpenScope adds new functionality to get that into the attack simulator.

E. Development of educational content

Instructional videos and a document describing the newly added attack functionality was created to enhance the cyber awareness training. Future work in this area may include additional educational content covering in more detail how GPS-jamming works on a technical level, and why it affects the aircraft as shown in the simulator.

Acknowledgment

Special thanks to Gurjot Singh Gaba who helped us immensely with this project. He helped us pick well-thought-out research questions and kept us on track toward reaching the goal during the entirety of the project. We also want to thank our supervisor Andrei Gurtov who helped us find domain-specific knowledge about airports and airplanes. Finally, we want to thank Ulf Kargén, the examiner in this course. For his guidance, instructions, and timely feedback.

REFERENCES

[1] A. Blåberg, G. Lindahl, A. Gurtov and B. Josefsson, "Simulating ADS-B Attacks in Air Traffic Management", 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 2020, pp. 1-10, doi: 10.1109/DASC50938.2020.9256438.

[2] M. Makar, J. Ekman, J. Granlund, A, Johnsson. "Arlanda stängdes efter drönarlarm: 'Misstänker medveten handling"', SVT Nyheter [Internet], (2024-09-16), https://www.svt.se/nyheter/lokalt/stockholm/arlanda-stangs-flygplan-dirigeras-om-till-skavsta

[3] K. Zhang, "Investigating GPS Vulnerability", Dissertation, KTH, School of Electrical Engineering, Stockholm, Sweden, 2013. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1537307/FULLTEXT01.pdf

[4] Faria, Lester and Silvestre, Caio and Correia, Marcelino, "GPS-Dependent Systems: Vulnerabilities to Electromagnetic Attacks" 2016 Journal of Aerospace Technology and Management, pp. 423-430, doi: 10.5028/jatm.v8i4.632.

[5] Luftfartsverket, (25-11-2024), "esms-airport", https://aro.lfv.se

[6] Openscope, (25-11-2024), "Airport-format.md", github https://github.com/openscope/openscope/blob/develop/documentation/airport-format.md

[7] Openscope, (25-11-2024), "Airport-file-standards.md", github https://github.com/openscope/openscope/blob/develop/documentation/airport-file-standards.md

[8] Chartjs Contributors, (25-11-2024), "Chart.js", https://www.chartjs.org