Retrospective Sprint 3 of Group AttackFlow1

Building a dataset of real-world cyber-attacks with Attack Flow

Team members:

Se Jin Yoon: a1706219

Ting-Wei Chin: a1782423

Faisal Hawsawi: a1822781

Lina Nehme: a1802697

Ran Qi: a1675122

Joseph Toubia: a1753547

Zemin Wong: a1780385

Jixing Ye: a1798631

Yu Zheng: a1739446

Snapshots (Group):

1. Client meeting: Wednesday-04-10-2023.
2. Retrospective meeting: Thursday-5-10-2023.

I attended both the above meetings with team members and the client.

*For snapshots see the end of this report

What went well in the sprint (Individually Written)?

Task Allocation and Management:

1.1 The sprint witnessed a clear delineation of tasks among team members, ensuring everyone was well-aware of their responsibilities.

1.2 The progress tracking mechanism implemented proved effective, providing a real-time insight into the project's trajectory towards the deadlines.

1.3 The estimation of due dates was accurate, which is a testament to the team's understanding and experience in managing the workload efficiently.

Adaptability to Requirement Modifications:

2.1 A shift in priority was smoothly executed from the admin page development to more crucial implementations like the annotation page and enhancing the response accuracy of ChatGPT.

2.2 The change in access control for viewing history versions of annotations from all users to only annotators was a strategic move towards enhancing user experience and data security.

2.3 Transitioning from using ChatGPT for technique extraction to metadata extraction only, was aligned with the client’s requirements, showing the team’s flexibility and client-centric approach.

Scrum Master's Proficiency:

3.1 The Scrum Master played a pivotal role in alleviating roadblocks, thus promoting an environment conducive for enhanced team performance.

3.2 The sprint meetings were productive, fostering a collaborative space for ideas, feedback, and strategizing towards sprint goals.

3.3 The agile process was further refined, ensuring that the goals, scope, and domain of the product were well-comprehended by the development team, which is fundamental for the project's success.

Risk Management Amid GitHub Outage:

During the unfortunate event of GitHub going down during Sprint 3, the team was not deterred but promptly switched to an alternative platform as per the predefined risk management plan. This instance showcased the team’s preparedness and resilience in the face of unforeseen technical glitches.

What could be improved (Individually Written)?

Stabilizing Meeting Schedules:
1.1 The irregularity in meeting schedules has been identified as a bottleneck causing the absence of many team members. Adhering to fixed schedules for meetings (e.g., every Monday and Thursday) will ensure better attendance and engagement from all team members.

Incorporating Quality Management Plans:
2.1 Implementing a comprehensive quality management plan is imperative to uphold the coding standards and documentation consistency as outlined in Lecture 15, Slide 18. This would include adherence to a defined coding style, maintaining documentation standards, and other best practices ensuring the software developed is of high quality and maintainable.

Enhanced Commit Control:

3.1 Establishing a structured commit control and code review process is fundamental. For instance, having version control such as v1, v1.2, conducting code reviews, and merging into the main branch only when a new feature is validated and accepted. This practice will contribute to better code integrity and easier tracking of changes.

Robust Testing Framework:
4.1 Developing a well-defined testing plan is paramount to ensure the system's reliability and fault tolerance.
4.2 Making the project fault-tolerant will contribute to a more robust and dependable system capable of handling unexpected scenarios gracefully.
4.3 Incorporating unit tests, with assertive checks, will help in early identification of bugs and inconsistencies, thereby ensuring that each component is functioning as intended.

Comment on your progress this sprint (Individually Written)

What I did:
1. Attend all the meetings and engaged in discussions
2. Worked with Lina to provide technical support on her function of uploading part.

(The main issue was due to the CORS policy. The server-side code was not correctly set up to handle requests from the client-side origin. The upload.js file was a complete Express application, which should have been a router handler instead.)

3. Worked with Yu Zheng to fix his code caused by pdf uploading.

(The upload was only support text, when it comes with pdf, the result would be a mess. It was mainly due to not being equipped with pdf-parse)
(Also, I fixed the annotation link which used the entire text as a query parameter to the URL can cause some problems. There is a limit to the length of the URL, and for very large text, this can cause the URL to be too long and the request to fail. From a security perspective, having a large amount of text as a URL parameter may expose sensitive information because urls may be stored in server logs or browser history. From a performance point of view, processing a large number of URL parameters can increase the burden on the server. I store it on the server and just pass an identifier or URL to the comment page, where I request the text content again.)

What I'm still working on:
1. Setting up chatgpt environment in backend. (I tried frontend but was warned by openai:
   Uncaught Error: It looks like you're running in a browser-like environment.

This is disabled by default, as it risks exposing your secret API credentials to attackers.
If you understand the risks and have appropriate mitigations in place,
you can set the `dangerouslyAllowBrowser` option to `true`, e.g.,

new OpenAI({ apiKey, dangerouslyAllowBrowser: true });

https://help.openai.com/en/articles/5112595-best-practices-for-api-key-safety

at new OpenAI (index.ts:128:7)

at chatgpt.mjs:5:16) (Also, It was stuck by either axios or openai, it seems that I cannot apply them both which is still need to be addressed)

2. Use chatgpt to help with annotation. (By providing suggestions on grabing keywords from the uploaded pdf cases)

Snapshots:

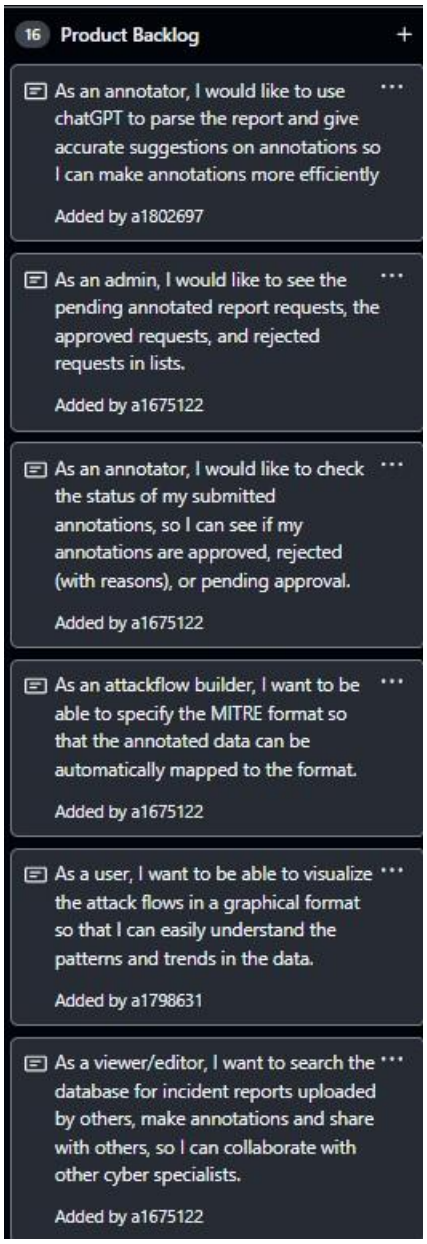
Snapshot Week 7 of Group AttackFlow1

Building a dataset of real-world cyber-attacks with Attack Flow

Se Jin Yoon: a1706219
Ting-Wei Chin: a1782423
Faisal Hawsawi: a1822781
Lina Nehme: a1802697
Ran Qi: a1675122
Joseph Toubia: a1753547
Zemin Wong: a1780385
Jixing Ye: a1798631
Yu Zheng: a1739446

September 17, 2023

Product Backlog



16 Product Backlog +

As an annotator, I would like to use chatGPT to parse the report and give accurate suggestions on annotations so I can make annotations more efficiently

Added by a1802697

As an admin, I would like to see the pending annotated report requests, the approved requests, and rejected requests in lists.

Added by a1675122

As an annotator, I would like to check the status of my submitted annotations, so I can see if my annotations are approved, rejected (with reasons), or pending approval.

Added by a1675122

As an attackflow builder, I want to be able to specify the MITRE format so that the annotated data can be automatically mapped to the format.

Added by a1675122

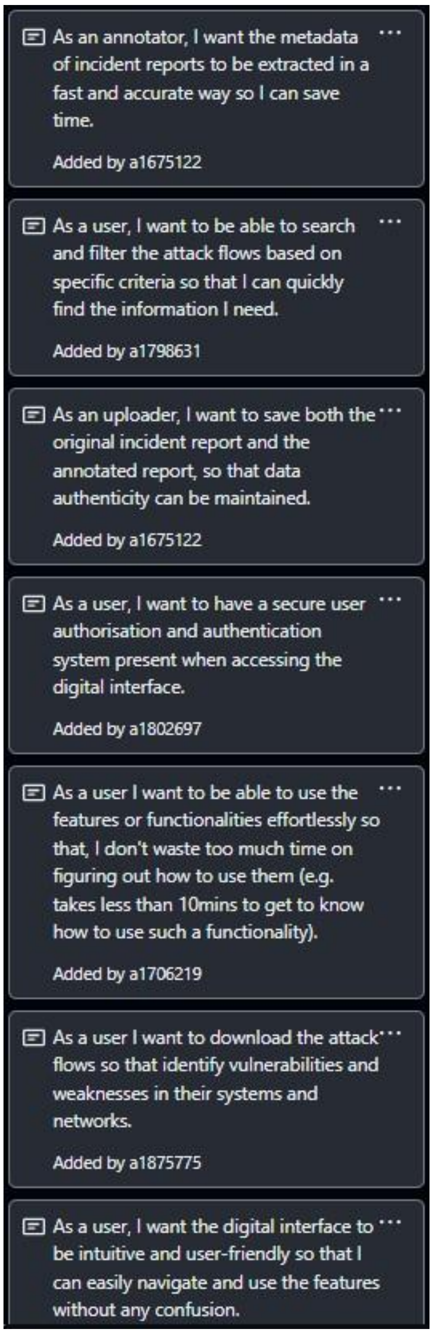
As a user, I want to be able to visualize the attack flows in a graphical format so that I can easily understand the patterns and trends in the data.

Added by a1798631

As a viewer/editor, I want to search the database for incident reports uploaded by others, make annotations and share with others, so I can collaborate with other cyber specialists.

Added by a1675122

Figure 1: Product Backlog (Sprint 3) 1 of 3



As an annotator, I want the metadata of incident reports to be extracted in a fast and accurate way so I can save time.

Added by a1675122

As a user, I want to be able to search and filter the attack flows based on specific criteria so that I can quickly find the information I need.

Added by a1798631

As an uploader, I want to save both the original incident report and the annotated report, so that data authenticity can be maintained.

Added by a1675122

As a user, I want to have a secure user authorisation and authentication system present when accessing the digital interface.

Added by a1802697

As a user I want to be able to use the features or functionalities effortlessly so that, I don't waste too much time on figuring out how to use them (e.g. takes less than 10mins to get to know how to use such a functionality).

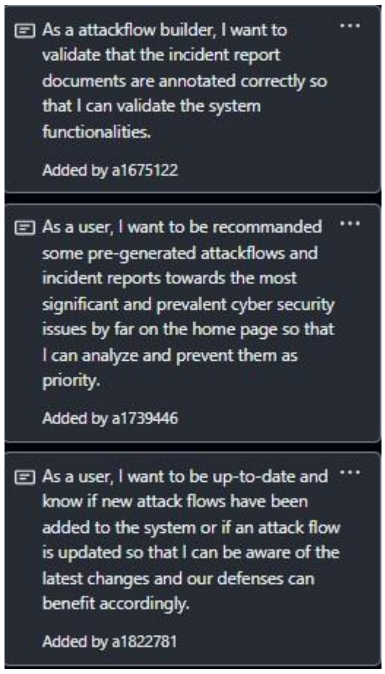
Added by a1706219

As a user I want to download the attack flows so that identify vulnerabilities and weaknesses in their systems and networks.

Added by a1875775

As a user, I want the digital interface to be intuitive and user-friendly so that I can easily navigate and use the features without any confusion.

Figure 2: Product Backlog (Sprint 3) 2 of 3



As a attackflow builder, I want to validate that the incident report documents are annotated correctly so that I can validate the system functionalities.

Added by a1675122

As a user, I want to be recommanded some pre-generated attackflows and incident reports towards the most significant and prevalent cyber security issues by far on the home page so that I can analyze and prevent them as priority.

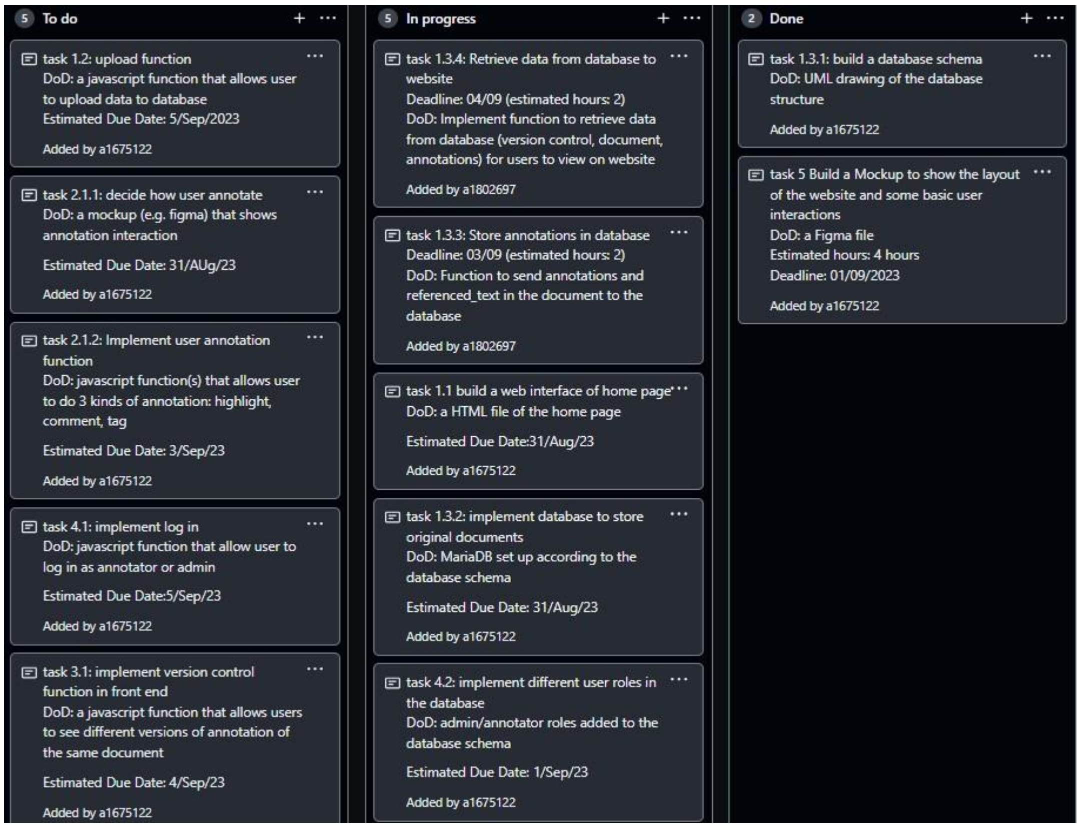
Added by a1739446

As a user, I want to be up-to-date and know if new attack flows have been added to the system or if an attack flow is updated so that I can be aware of the latest changes and our defenses can benefit accordingly.

Added by a1822781

Figure 3: Product Backlog (Sprint 3) 3 of 3

Task Board



5 To do

task 1.2: upload function
DoD: a javascript function that allows user to upload data to database
Estimated Due Date: 5/Sep/2023
Added by a1675122

task 2.1.1: decide how user annotate
DoD: a mockup (e.g. figma) that shows annotation interaction
Estimated Due Date: 31/AUg/23
Added by a1675122

task 2.1.2: Implement user annotation function
DoD: javascript function(s) that allows user to do 3 kinds of annotation: highlight, comment, tag
Estimated Due Date: 3/Sep/23
Added by a1675122

task 4.1: implement log in
DoD: javascript function that allow user to log in as annotator or admin
Estimated Due Date:5/Sep/23
Added by a1675122

task 3.1: implement version control function in front end
DoD: a javascript function that allows users to see different versions of annotation of the same document
Estimated Due Date: 4/Sep/23
Added by a1675122

5 In progress

task 1.3.4: Retrieve data from database to website
Deadline: 04/09 (estimated hours: 2)
DoD: Implement function to retrieve data from database (version control, document, annotations) for users to view on website
Added by a1802697

task 1.3.3: Store annotations in database
Deadline: 03/09 (estimated hours: 2)
DoD: Function to send annotations and referenced_text in the document to the database
Added by a1802697

task 1.1 build a web interface of home page
DoD: a HTML file of the home page
Estimated Due Date:31/Aug/23
Added by a1675122

task 1.3.2: implement database to store original documents
DoD: MariaDB set up according to the database schema
Estimated Due Date: 31/Aug/23
Added by a1675122

task 4.2: implement different user roles in the database
DoD: admin/annotator roles added to the database schema
Estimated Due Date: 1/Sep/23
Added by a1675122

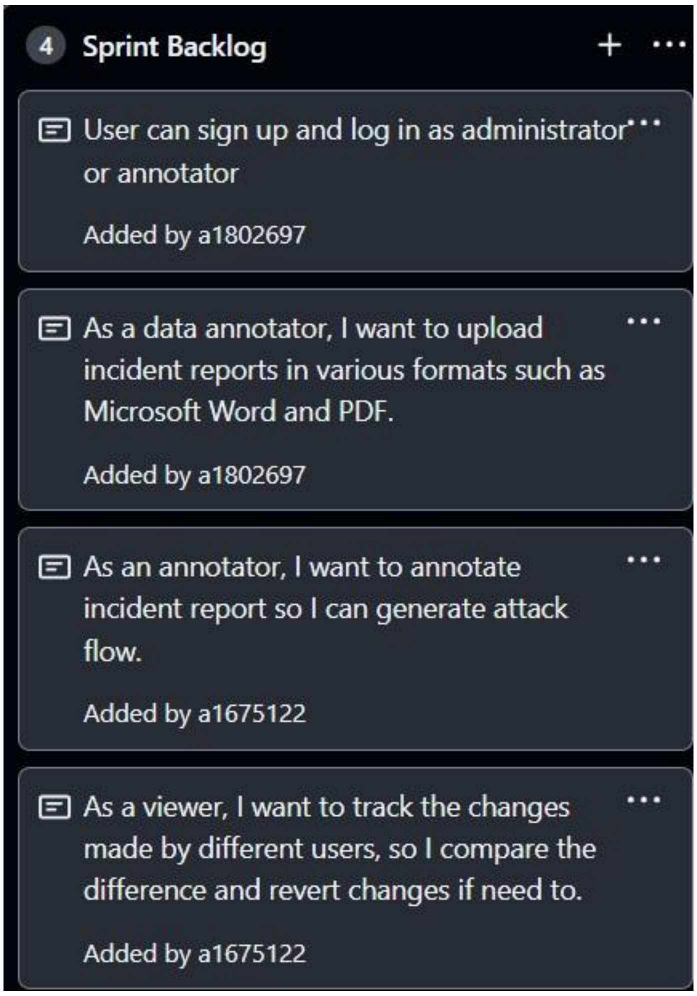
2 Done

task 1.3.1: build a database schema
DoD: UML drawing of the database structure
Added by a1675122

task 5 Build a Mockup to show the layout of the website and some basic user interactions
DoD: a Figma file
Estimated hours: 4 hours
Deadline: 01/09/2023
Added by a1675122

Figure 4: Task board (Sprint 3)

Sprint Backlog



4 Sprint Backlog

User can sign up and log in as administrator or annotator

Added by a1802697

As a data annotator, I want to upload incident reports in various formats such as Microsoft Word and PDF.

Added by a1802697

As an annotator, I want to annotate incident report so I can generate attack flow.

Added by a1675122

As a viewer, I want to track the changes made by different users, so I compare the difference and revert changes if need to.

Added by a1675122

Figure 5: Sprint Backlog (Sprint 3)

User Stories

We have identified that some user stories like upload and annotate selected from the product backlog for sprint 2 remain incomplete due to underestimating the time complexity needed to complete the tasks. Therefore, we have selected the user stories document upload and annotate, as well as user sign ups and logins for this sprint. From this, we have determined the following tasks for sprint 3:

- “As a data annotator, I want to upload an incident report and save it so that I can further edit the file.”

  – Related tasks:
  task 1.2: upload function
  task 1.3.2: implement database to store original documents

- “As an annotator, I want to annotate incident report so I can generate attack flow.”

  – Related tasks:
  task 2.1.1: decide how user annotate
  task 2.1.2: implement user annotation function
  task 2.1.3: integrate ChatGPT to parse report and give accurate annotation suggestions

- “User can sign up and log in as administrator or annotator

  – Related tasks:
  task 4.1: implement sign up
  task 4.2: implement log in
  task 4.3: implement different user roles in the database

Definition of Done

The individual DoD of each task can be seen on the screenshot of the task board.
In summary, the goals are:

- Testing the integrated backend functions works as we expected.
- Code is well-documented and adheres to our coding standards.
- Explain and ensure that all the scrum members understand the specifics of the tasks and how to construct the implementation.

Summary of Changes

The last snapshot focused on the functional requirements upload and annotate. While this snapshot will continue to focus on upload and annotate, we will also emphasise the functional requirements of user sign ups and logins. Some additional aspects that have changed since our last snapshot are:

- **Added new user stories**- During our meeting with the tutor we discovered new user stories. The first user story involves enabling annotators to check the status of their submitted annotation which can be either approved, rejected (with reasons) or pending approval. We also discovered a second user story regarding admins being able to see the pending annotated report requests, the approved requests, and rejected requests in lists. Lastly, we have clarified that ChatGPT will be integrated to parse the report and give suggestions to accurate annotations for the annotator to select. Thus, we have added a user story to reflect this requirement.

- **Refined our user stories to better align with the tutor's requirements.** We have updated our login user story to include both user logins and sign ups only for administrator and annotator after determining that end users from the general public do not need to make their own account to interact with the web application.

- **We have simplified our user story for uploading reports from the previous snapshot**. After discussions with our tutor, we determined that the functionality of saving incomplete annotated reports presubmission for further edits was unnecessary. Therefore, we have updated our upload user story to exclude this functionality and enabled users to only upload, annotate and submit their reports without saving incomplete annotations during the annotation process.

- **This snapshot emphasises the major role of ChatGPT in parsing the report and providing accurate annotation suggestions.** We have integrated ChatGPT in our web application to give standard responses. Our next steps involve learning the mechanisms that enable ChatGPT to deliver consistent and accurate annotation suggestions including technique IDs and associated tags based on our inputs.

Snapshot Week 8 of Group AttackFlow1

Building a dataset of real-world cyber-attacks with Attack Flow

Se Jin Yoon: a1706219
Ting-Wei Chin: a1782423
Faisal Hawsawi: a1822781
Lina Nehme: a1802697
Ran Qi: a1675122
Joseph Toubia: a1753547
Zemin Wong: a1780385
Jixing Ye: a1798631
Yu Zheng: a1739446

September 17, 2023

Product Backlog

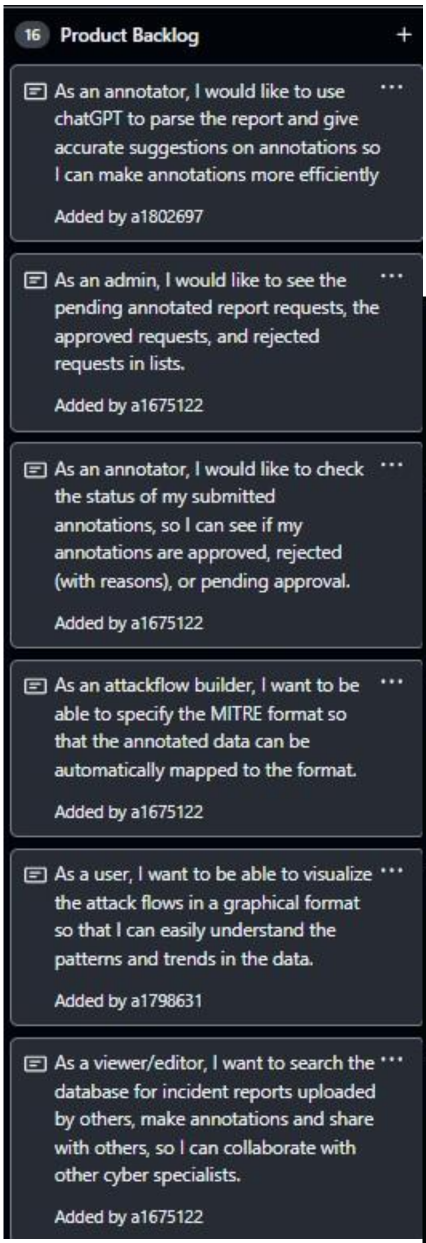


Figure 1: Product Backlog (Sprint 4) 1 of 3

1

16 Product Backlog +

As an annotator, I would like to use chatGPT to parse the report and give accurate suggestions on annotations so I can make annotations more efficiently

Added by a1802697

As an admin, I would like to see the pending annotated report requests, the approved requests, and rejected requests in lists.

Added by a1675122

As an annotator, I would like to check the status of my submitted annotations, so I can see if my annotations are approved, rejected (with reasons), or pending approval.

Added by a1675122

As an attackflow builder, I want to be able to specify the MITRE format so that the annotated data can be automatically mapped to the format.

Added by a1675122

As a user, I want to be able to visualize the attack flows in a graphical format so that I can easily understand the patterns and trends in the data.

Added by a1798631

As a viewer/editor, I want to search the database for incident reports uploaded by others, make annotations and share with others, so I can collaborate with other cyber specialists.

Added by a1675122

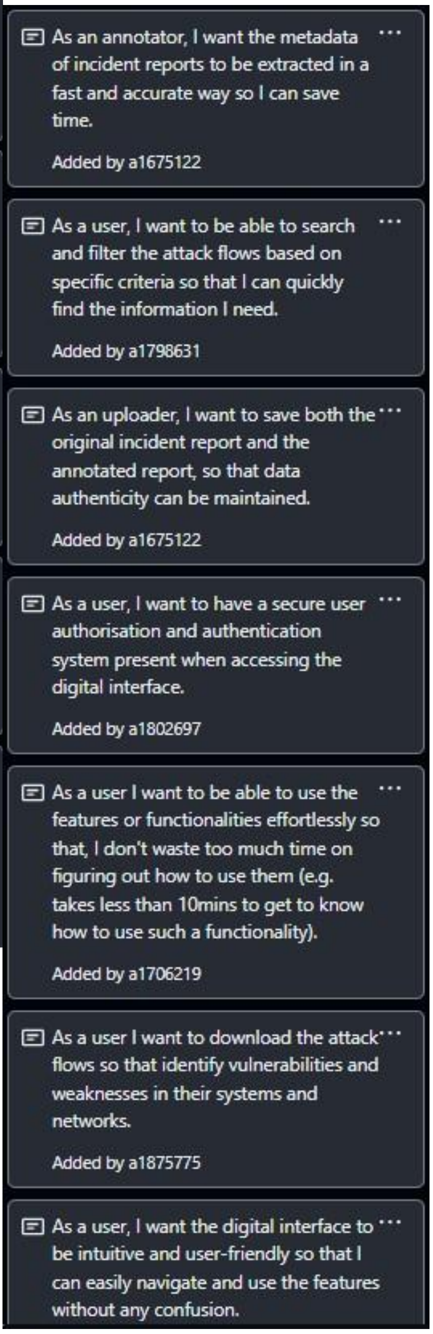


Figure 2: Product Backlog (Sprint 4) 2 of 3

2

As an annotator, I want the metadata of incident reports to be extracted in a fast and accurate way so I can save time.

Added by a1675122

As a user, I want to be able to search and filter the attack flows based on specific criteria so that I can quickly find the information I need.

Added by a1798631

As an uploader, I want to save both the original incident report and the annotated report, so that data authenticity can be maintained.

Added by a1675122

As a user, I want to have a secure user authorisation and authentication system present when accessing the digital interface.

Added by a1802697

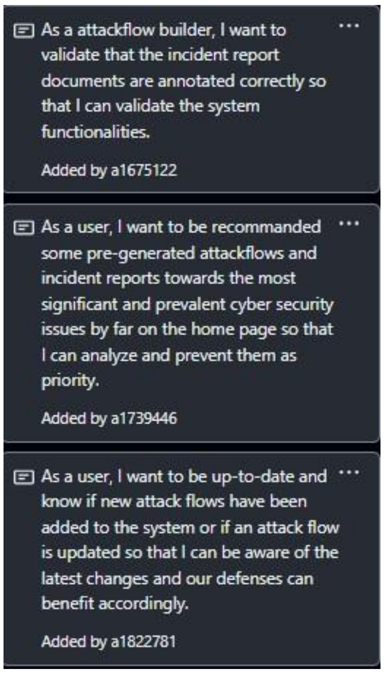
As a user I want to be able to use the features or functionalities effortlessly so that, I don't waste too much time on figuring out how to use them (e.g. takes less than 10mins to get to know how to use such a functionality).

Added by a1706219

As a user I want to download the attack flows so that identify vulnerabilities and weaknesses in their systems and networks.

Added by a1875775

As a user, I want the digital interface to be intuitive and user-friendly so that I can easily navigate and use the features without any confusion.



As a attackflow builder, I want to validate that the incident report documents are annotated correctly so that I can validate the system functionalities.

Added by a1675122

As a user, I want to be recommanded some pre-generated attackflows and incident reports towards the most significant and prevalent cyber security issues by far on the home page so that I can analyze and prevent them as priority.

Added by a1739446

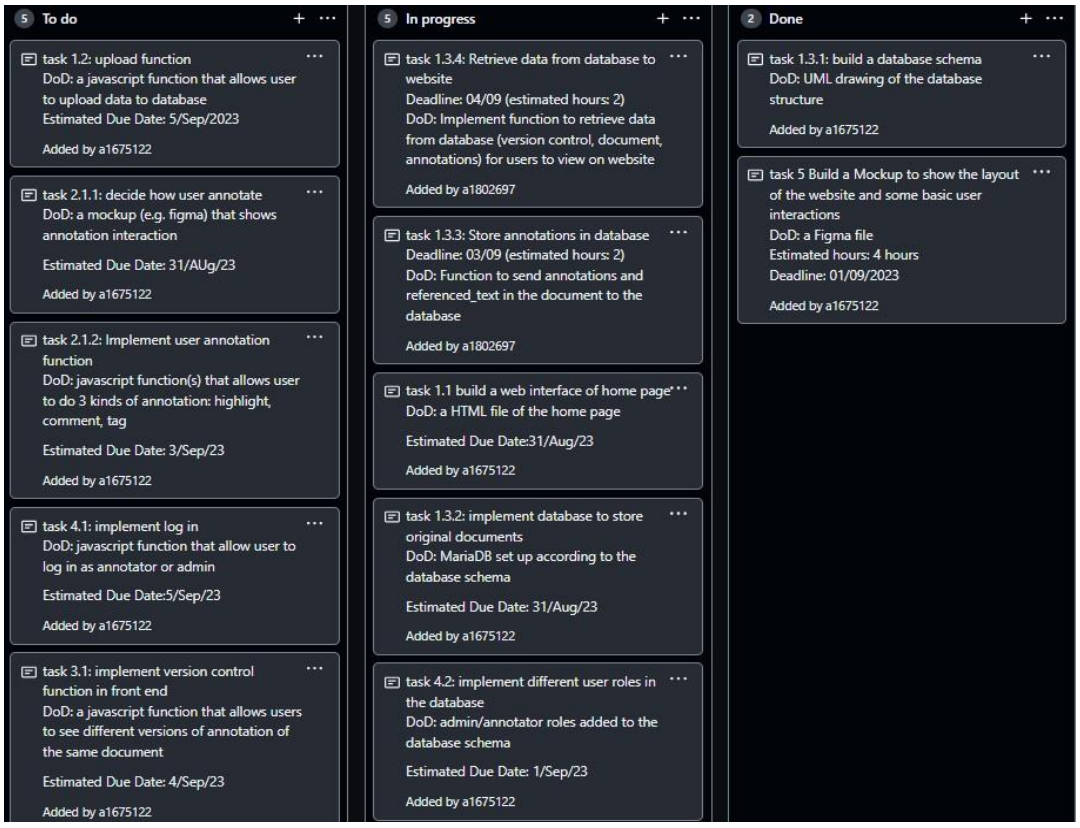
As a user, I want to be up-to-date and know if new attack flows have been added to the system or if an attack flow is updated so that I can be aware of the latest changes and our defenses can benefit accordingly.

Added by a1822781

Figure 3: Product Backlog (Sprint 4) 3 of 3

Task Board

Figure 4: Task board (Sprint 4)



5 To do

task 1.2: upload function
DoD: a javascript function that allows user to upload data to database
Estimated Due Date: 5/Sep/2023
Added by a1675122

task 2.1.1: decide how user annotate
DoD: a mockup (e.g. figma) that shows annotation interaction
Estimated Due Date: 31/AUg/23
Added by a1675122

task 2.1.2: Implement user annotation function
DoD: javascript function(s) that allows user to do 3 kinds of annotation: highlight, comment, tag
Estimated Due Date: 3/Sep/23
Added by a1675122

task 4.1: implement log in
DoD: javascript function that allow user to log in as annotator or admin
Estimated Due Date:5/Sep/23
Added by a1675122

task 3.1: implement version control function in front end
DoD: a javascript function that allows users to see different versions of annotation of the same document
Estimated Due Date: 4/Sep/23
Added by a1675122

5 In progress

task 1.3.4: Retrieve data from database to website
Deadline: 04/09 (estimated hours: 2)
DoD: Implement function to retrieve data from database (version control, document, annotations) for users to view on website
Added by a1802697

task 1.3.3: Store annotations in database
Deadline: 03/09 (estimated hours: 2)
DoD: Function to send annotations and referenced_text in the document to the database
Added by a1802697

task 1.1 build a web interface of home page
DoD: a HTML file of the home page
Estimated Due Date:31/Aug/23
Added by a1675122

task 1.3.2: implement database to store original documents
DoD: MariaDB set up according to the database schema
Estimated Due Date: 31/Aug/23
Added by a1675122

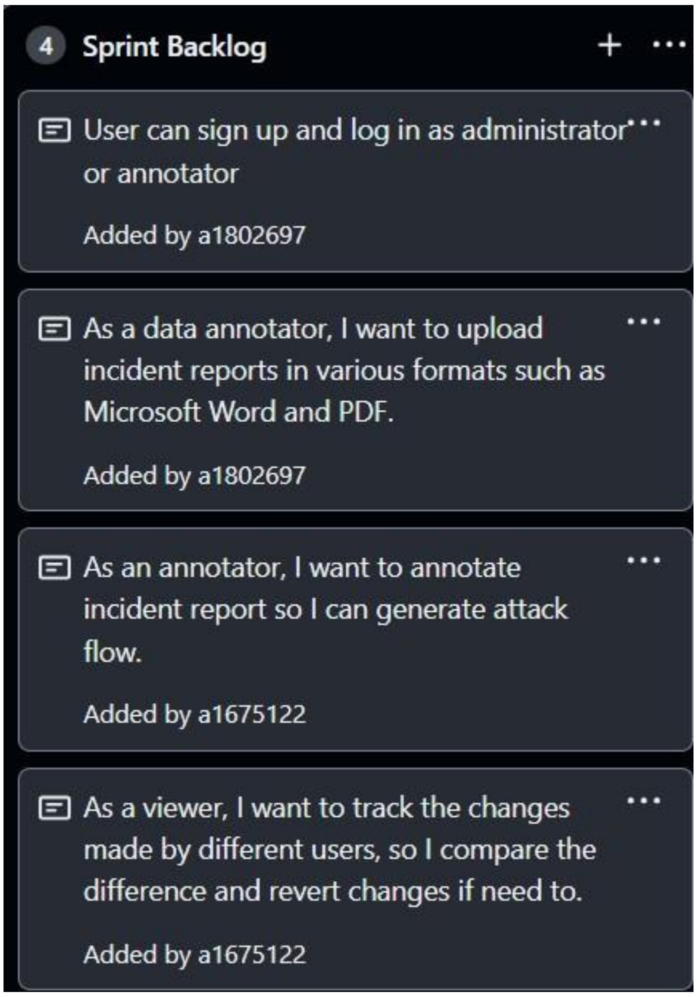
task 4.2: implement different user roles in the database
DoD: admin/annotator roles added to the database schema
Estimated Due Date: 1/Sep/23
Added by a1675122

2 Done

task 1.3.1: build a database schema
DoD: UML drawing of the database structure
Added by a1675122

task 5 Build a Mockup to show the layout of the website and some basic user interactions
DoD: a Figma file
Estimated hours: 4 hours
Deadline: 01/09/2023
Added by a1675122

Sprint Backlog


4 Sprint Backlog

User can sign up and log in as administrator or annotator

Added by a1802697

As a data annotator, I want to upload incident reports in various formats such as Microsoft Word and PDF.

Added by a1802697

As an annotator, I want to annotate incident report so I can generate attack flow.

Added by a1675122

As a viewer, I want to track the changes made by different users, so I compare the difference and revert changes if need to.

Added by a1675122

Figure 5: Sprint Backlog (Sprint 4)

User Stories

We have identified that some user stories like upload and annotate selected from the product backlog for sprint 2 and 3 remain incomplete due to underestimating the time complexity needed to complete the tasks. Therefore, we have selected the user stories document upload and annotate, as well as user sign ups and logins for this sprint. From this, we have determined the following tasks for sprint 3 and 4:

- “As a data annotator, I want to upload an incident report and save it so that I can further edit the file.”

  – Related tasks:
  task 1.2: upload function
  task 1.3.2: implement database to store original documents

- “As an annotator, I want to annotate incident report so I can generate attack flow.”

  – Related tasks:
  task 2.1.1: decide how user annotate
  task 2.1.2: implement user annotation function
  task 2.1.3: integrate ChatGPT to parse report and give accurate annotation suggestions

- “User can sign up and log in as administrator or annotator

  – Related tasks:
  task 4.1: implement sign up
  task 4.2: implement log in
  task 4.3: implement different user roles in the database

Definition of Done

The individual DoD of each task can be seen on the screenshot of the task board.
In summary, the goals are:

- Testing the integrated backend functions works as we expected.
- Code is well-documented and adheres to our coding standards.
- Explain and ensure that all the scrum members understand the specifics of the tasks and how to construct the implementation.

Summary of Changes

The last snapshot focused on the upload and annotate functions also there is an emphasis on the functional requirements of user sign ups and logins. The following aspects will be carried over to this snapshot:

- **Added new user stories**- During our meeting with the tutor we discovered new user stories. The first user story involves enabling annotators to check the status of their submitted annotation which can be either approved, rejected (with reasons) or pending approval. We also discovered a second user story regarding admins being able to see the pending annotated report requests, the approved requests, and rejected requests in lists. Lastly, we have clarified that ChatGPT will be integrated to parse the report and give suggestions to accurate annotations for the annotator to select. Thus, we have added a user story to reflect this requirement.

- **Refined our user stories to better align with the tutor's requirements.** We have updated our login user story to include both user logins and sign ups only for administrator and annotator after determining that end users from the general public do not need to make their own account to interact with the web application.

- **We have simplified our user story for uploading reports from the previous snapshot**. After discussions with our tutor, we determined that the functionality of saving incomplete annotated reports presubmission for further edits was unnecessary. Therefore, we have updated our upload user story to exclude this functionality and enabled users to only upload, annotate and submit their reports without saving incomplete annotations during the annotation process.

- **This snapshot emphasises the major role of ChatGPT in parsing the report and providing accurate annotation suggestions.** We have integrated ChatGPT in our web application to give standard responses. Our next steps involve learning the mechanisms that enable ChatGPT to deliver consistent and accurate annotation suggestions including technique IDs and associated tags based on our inputs. Other third-party annotator software has been looked at but integration has been unsuccessful. The search for a compatible annotator will continue without ruling out the implementation of ChatGPT.

Please note, as there was only one day between the submission of snapshot 3 and this snapshot, the differences have been minimal because snapshot 3 has included all works up to and including the day of submission.