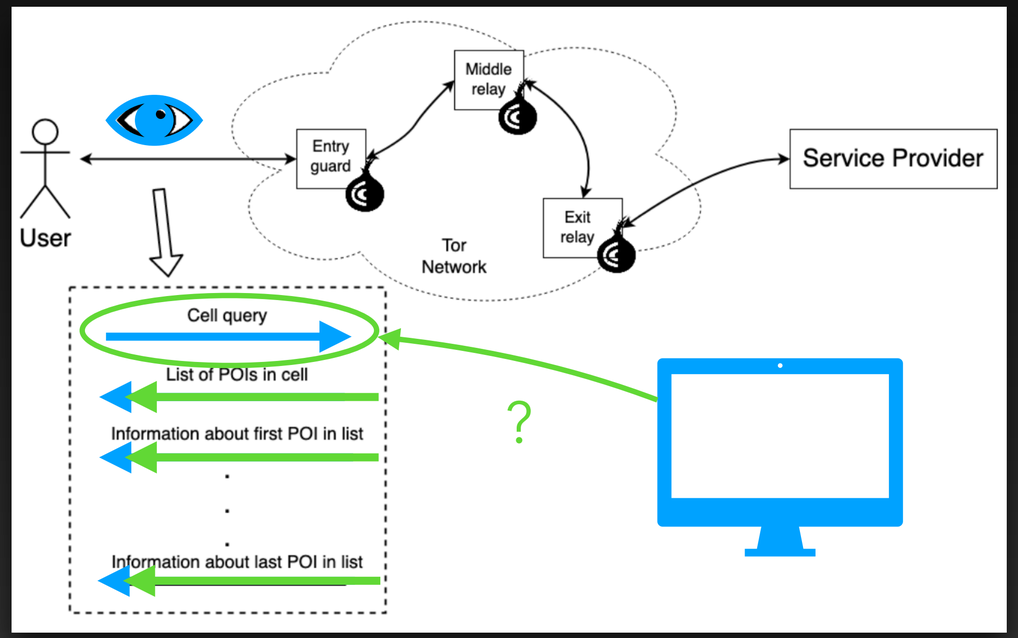
Secret Race Strolling - A scalable privacy competition platform

Semester Project at SPRING

Tom Demont supervised by Bogdan Kulynych - June 2022

Back to Secretstroll

- Queries for point of interest around location
- Client-Server architecture, networking through Tor
- Students should:
  - Collect network trace
  - Extract features and create model
  - Classify from network trace to queried location



Secretstroll system architecture

Exploit the privacy evaluation framework

- The privacy evaluation framework is there!
  - From theoretical questions to practical implementation
- AICrowd and Kaggle → ML Competition
  - Bring it further: p2p attack/defence competition
- Counter-measure parsing, data sets generation, automated evaluation
- Fully exploit the pedagogical utility of Secrestroll!

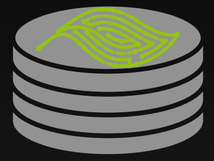
AIcrowd

kaggle

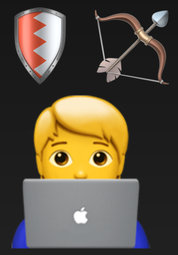
CS-523 student in a gladiator fight for SecretStroll

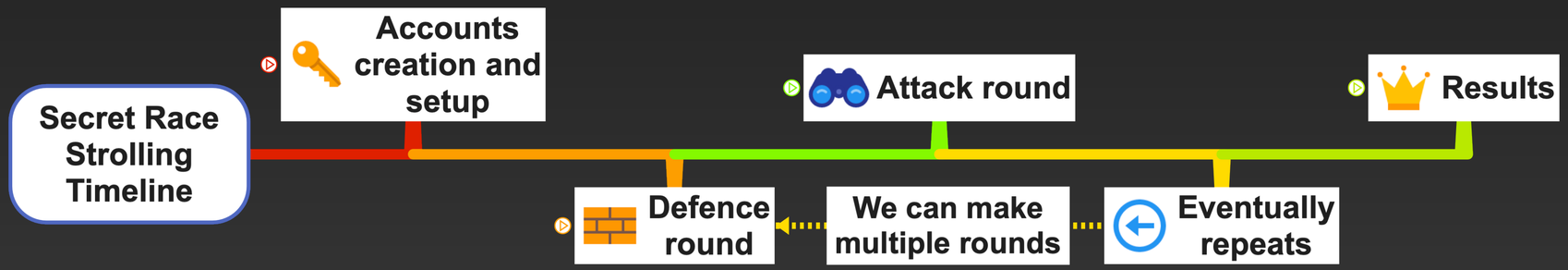
Secret Race Strolling - designing a competition

Create a timeline



- Imagined 2 approaches:
  - Fully interactive
  - Round based
- Considered interactivity, scalability and student workload



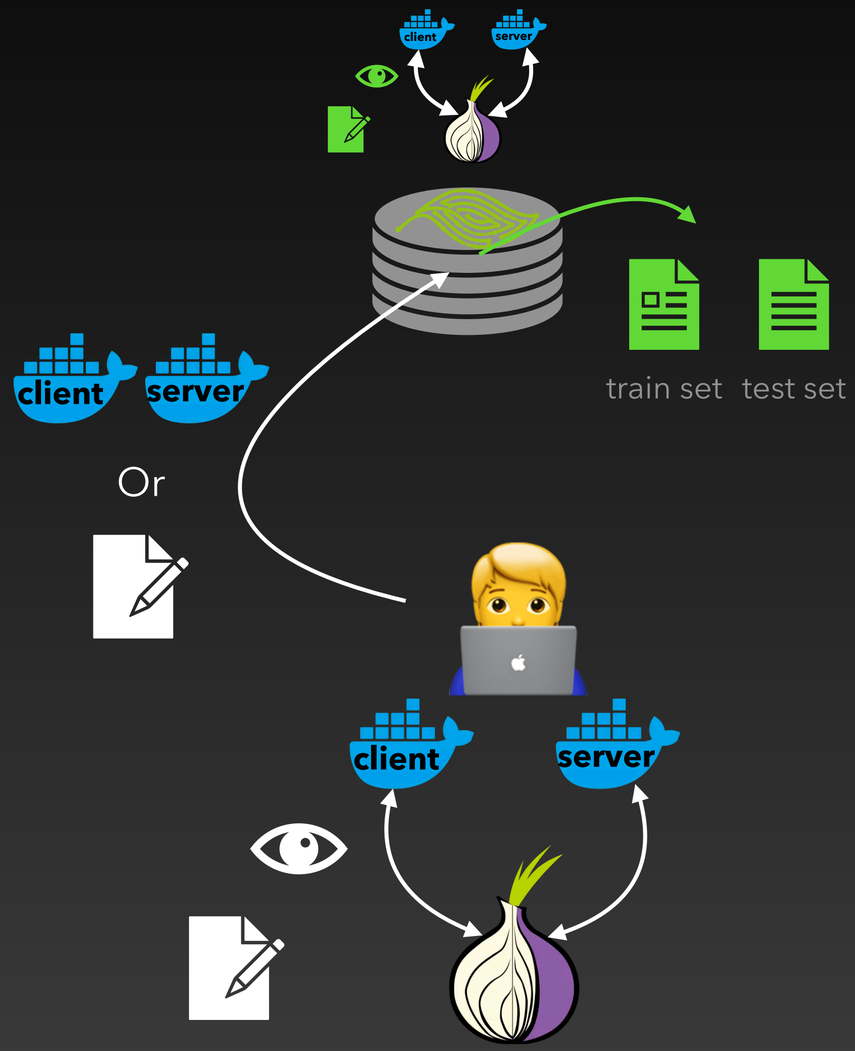


The final Secret Race Strolling timeline

Secret Race Strolling - designing a competition

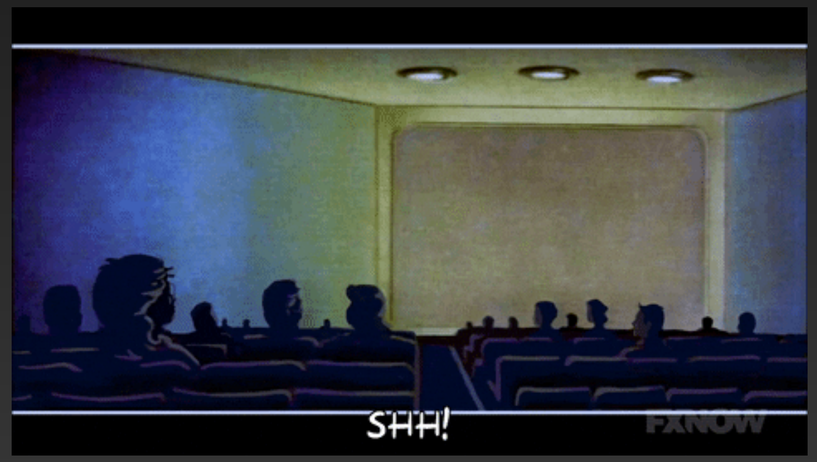
Attack and defence format

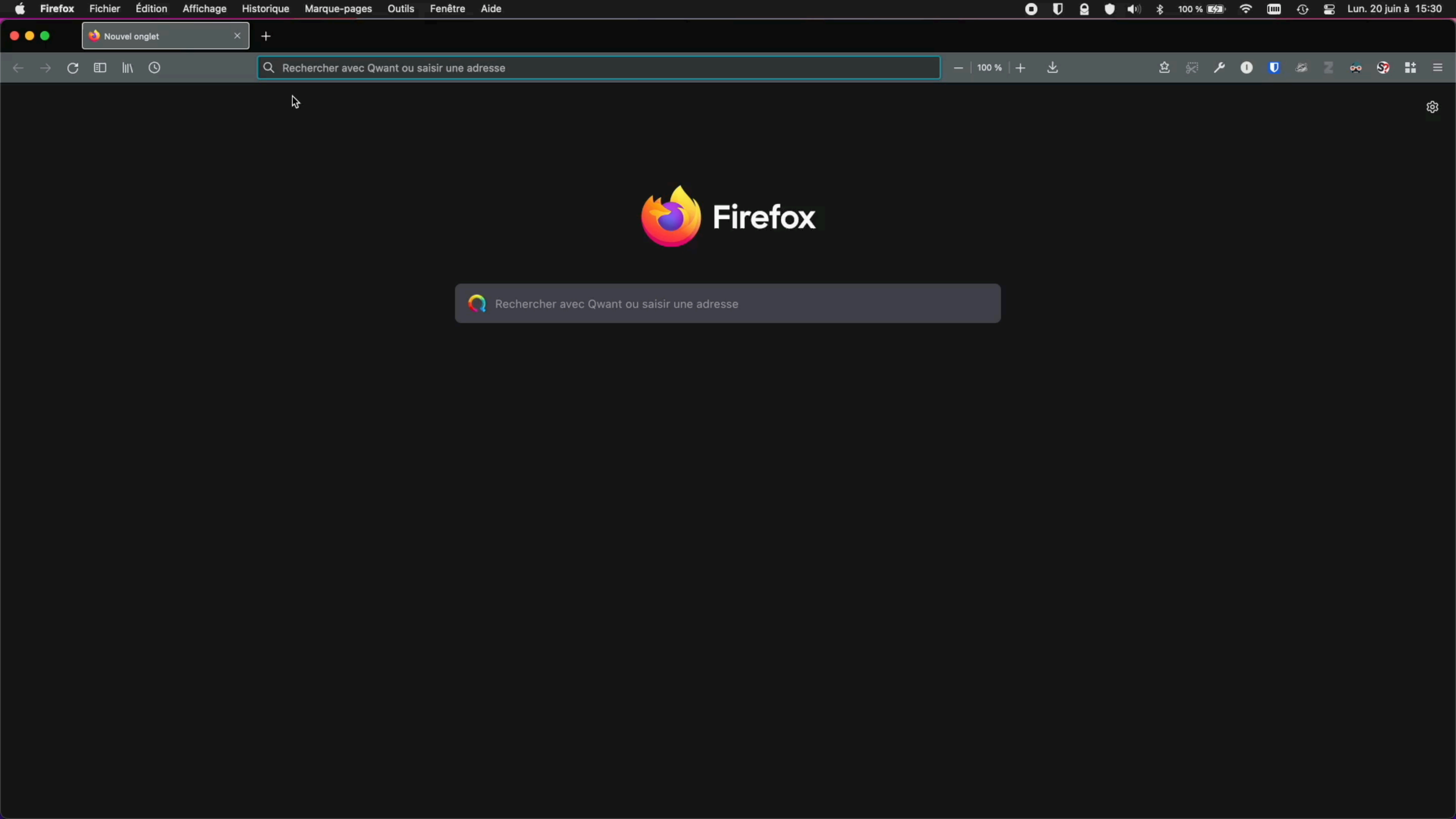
- Format of evaluated files
- Defence:
  - Key decision: student generated trace vs. server generated trace
  - Scalability and fairness issues
  - Went with student trace
- Attack:
  - AICrowd and Kaggle way: csv classification...
  - ... Classifier probability of each cell id!



Implementation considerations

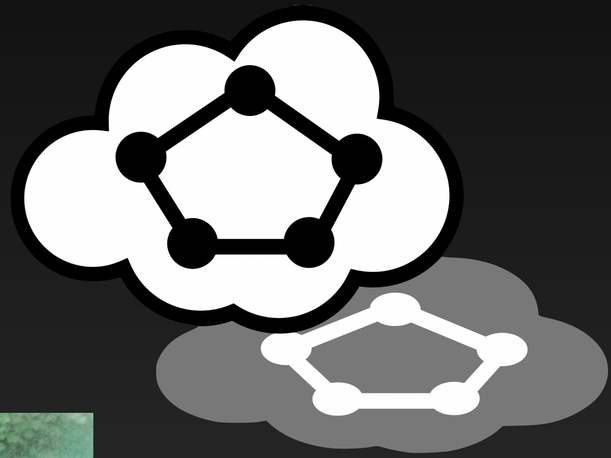
- Scoring & ranking: reflect attack/defence performance in one value
- Technological choices: scalable, user-friendly, developer-friendly
- ... But it's demo time!




Firefox Fichier Édition Affichage Historique Marque-pages Outils Fenêtre Aide
100 %
Lun. 20 juin à 15:30
Nouvel onglet
Rechercher avec Qwant ou saisir une adresse
100 %
Firefox
Rechercher avec Qwant ou saisir une adresse

Future work

- Server side trace generation: Shadow
  - Simulate tor network: scalability + fairness
- We tried it
  - Not stable docker
  - Ran simple non-tor network
  - Faced config generation tools



Shadow logo

Conclusion

- Designed a platform fitting CS-523 requirements
  - Evaluated different approaches on timeline, defence/attack format, ranking metrics and technological choices
- Implemented the platform
  - Working prototype
  - Friendly for appending future work

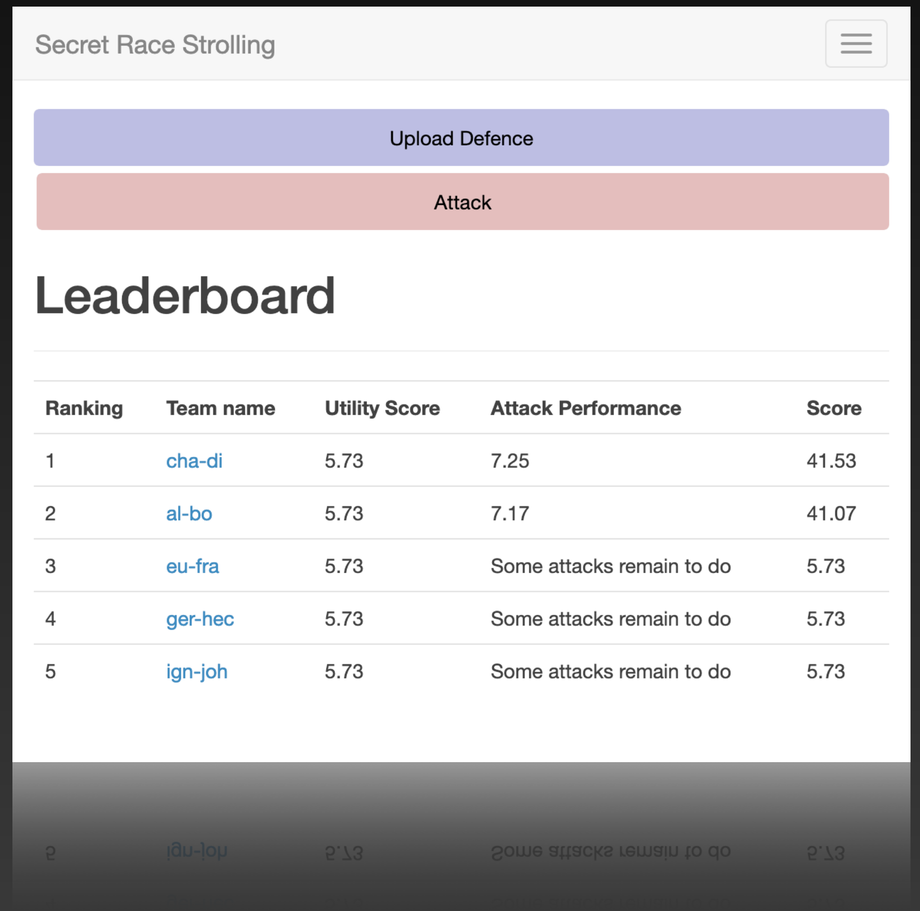
CS-523 students, waiting for more Secretstroll

Secret Race Strolling - designing a competition

Scoring system

- Give a score reflecting attack/defence performance → summary in one number for ranking
- Utility metric: over all recorded queried - $med_{in-volume} * med_{out-volume} * med_{time}$
- Attack performance metric: roc_auc_score
- Final score: ratio of both

Screenshot from the application on leaderboard page



Implementing this platform

- Flask: popular lightweight web framework
- Handled background async tasks with Celery
- Object Relational Model - SQLAlechmy: database agnostic

SQLAlchemy logo

Celery logo

Flask logo

Mental trace upload load

- Not breaking the server when teams upload
  - Avoid having RAM issues (CSV loaded in memory)
  - Avoid network congestion

Expected size of student uploaded/downloaded files

	Defence upload	**Attack upload**
Unit CSV size	48.1MB	879KB
Unit compressed CSV size	14.2MB	61KB
40 Teams CSV size	1.92GB	35.2MB
40 Teams compressed CSV size	568MB	2.4MB

From the raw Secretstroll implementation. Defence uploads could be bigger