CIS 350: AI for Cybersecurity

Week 13 Activity: Membership Inference Attack

CTFd Capture-the-Flag Challenge (Hard)

Activity Overview

A hospital trained an AI model and promised it never shares raw patient records. You are a privacy auditor with only query access to that model. Your mission is to prove you can tell whether one specific patient's record was used to TRAIN the model - a 'membership inference attack' - and then DEFEND the model using differential privacy. This connects directly to this week's lecture on privacy and the epsilon privacy budget, and to Lab 11 (Privacy DP), where you saw epsilon=0.1 give near-random predictions and epsilon=8 give near-baseline accuracy.

Ethics reminder: This is a sandboxed, authorized exercise. Only ever run these attacks against systems you have explicit permission to test.

Warm-Up (5 min) - Predict Before You Attack

Think-pair-share with a partner BEFORE you touch the notebook. Write a one-line prediction for each - you will check them against your real numbers later:
1. The model is asked how confident it is on a record it TRAINED on versus a brand-new record. Which one gets the higher confidence, and why? My prediction: _______________
2. If we add privacy noise until the attack becomes a coin flip (about 50 percent), what do you expect happens to the model's own accuracy? My prediction: _______________

Setup Instructions (Google Colab)

Resource: CTFd course server (ctfd.io) plus NIST Privacy Engineering guidance (nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-188.pdf)

1. Open the 'Membership Inference' challenge on the CTFd course server
2. Click the Colab starter notebook link inside the challenge
3. Run the top cell. It trains a small model and gives you a query function
4. You will use this starter pattern to inspect the model's confidence:

```
conf = model.predict_proba(record) # confidence on one record
```

Task 1: Measure the Confidence Gap (easiest)

Show that the model behaves differently on data it trained on versus new data. Take a batch of known TRAINING records and a batch of known TEST records, ask the model for its confidence on each, and average the two groups. **Hint:** use the starter predict_proba function. The training group's average confidence should be noticeably higher than the test group's.

Paste your code and output below:

Record your numbers: average confidence on TRAINING records = _______ average confidence on TEST records = _______

In one sentence, why is the training number higher?

Task 2: Build the Attack and Capture the Flag

Turn the confidence gap into a yes/no membership decision. Pick a confidence THRESHOLD: above it, guess 'in training'; below it, guess 'not in training'. Run your rule over the labeled check set and compute the attack accuracy. **Hint:** sweep a few threshold values and keep the best one. When your attack accuracy clears the target, the notebook prints the FLAG. Paste it into CTFd to score the capture.

Paste your attack code and output below:

Best threshold used: _______ Attack accuracy: _______ percent
Flag captured (flag{...} format): ______________________________

Checkpoint - discuss with a partner (2 min): Compare your best threshold and attack accuracy. Did you land on similar numbers? A privacy engineer at the hospital argues 'the model never revealed a single raw record, so patient privacy is fine.' Using your Task 2 result, give the one-sentence counter-argument you would make.

Task 3: Defend with Differential Privacy (hardest)

Retrain the model WITH differential privacy so your own attack stops working. The starter provides train_with_dp(epsilon). Retrain at the epsilon values from Lab 11, re-run your Task 2 attack against each DP model, and record both the ATTACK accuracy and the MODEL accuracy.
Hint: start at epsilon=0.1 (attack should collapse toward a coin flip near 50 percent, but model accuracy drops too), then try epsilon=8 (model accuracy returns, but the attack works again). Then find an epsilon that balances both.

Paste your defense code and output below:

Setting	Model accuracy	Attack accuracy
No DP (baseline)		
epsilon = 0.1		
epsilon = 8		
Your chosen balance: epsilon = ____		

Success Criteria and Flag Format

- Task 1: your training-group confidence is clearly higher than the test-group confidence
- Task 2: attack accuracy beats the target and the notebook prints a flag
- Flag format: **flag{...}** - copy the whole string, braces included, into CTFd
- Task 3: at epsilon=0.1 your attack accuracy falls near 50 percent (a coin flip), AND you can state the accuracy cost you paid to get there

Reflection

R1: In your own words, WHY does the model leak whether a record was in its training set? (Hint: think about memorization and confidence.)

R2: What is the DEFENSE against a membership inference attack, and how does the epsilon privacy budget control it? Use the numbers you recorded in Task 3.

R3: Describe the privacy-utility trade-off you observed. Why can a hospital NOT simply set epsilon as small as possible? And why is it wrong to say DP perfectly guarantees that membership is hidden?

Grading Rubric (20 points)

Criterion	Description	Points
Completion	All three tasks attempted; Colab notebook submitted	8
Technical execution	Attack works (flag captured) and DP defense produces the expected result	7
Understanding	Reflection explains WHY the attack works and the DEFENSE (epsilon)	5

Total: 20 points

Scoring: Completion (8) + Technical execution (7) + Understanding (5) = 20 points

How to Submit

1. Finish all three tasks in your Colab notebook
2. Submit the captured flag in CTFd (flag{...} format)
3. Download the notebook as .ipynb (File -> Download)
4. Complete the reflection questions in this worksheet
5. Submit the notebook and this completed worksheet on Canvas by [DUE DATE]

Questions? Post in the course forum or attend office hours!