PrivacyRaven: Comprehensive Privacy Testing for Deep Learning

Suha S. Hussain | Empire Hacking | August 2020

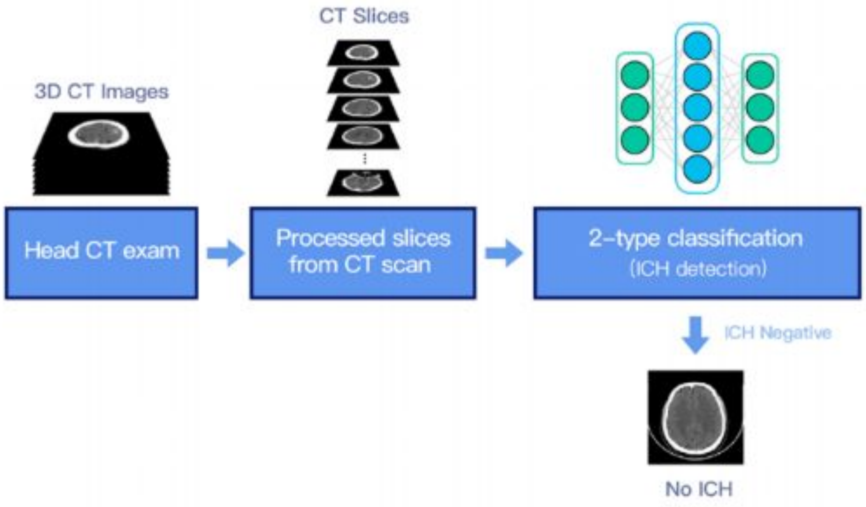
whoami

- **Suha S. Hussain (@suhackerr)**
  - Second-Year CS Major at Georgia Tech
    - Threads: Theory & People
  - Security Engineering Intern at Trail of Bits
    - Cryptography Team

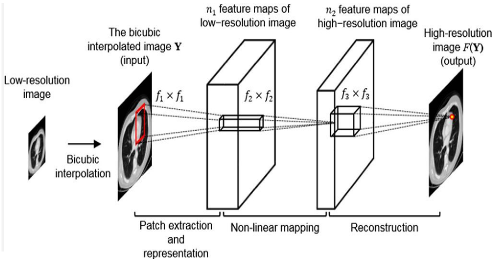
Auditing Deep Learning

How can this system be attacked?

- **Purpose: Detect a brain bleed from images of a scan**
    - Black-box
    - Binary result
- **Use PrivacyRaven to simulate privacy attacks**

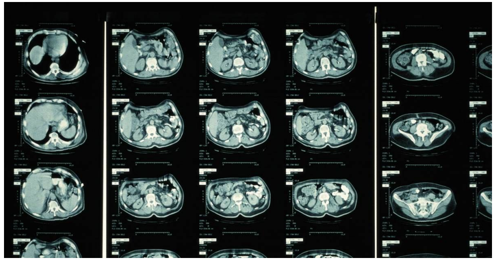


Privacy Violations



Intellectual Property

A substitute model was created from a **model extraction** attack.



Data Reconstruction

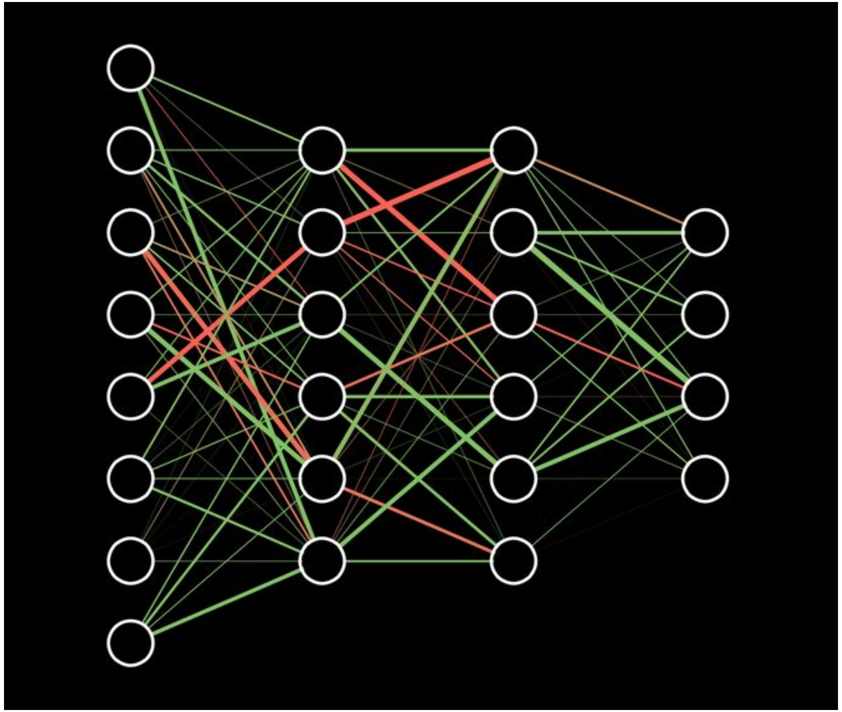
The adversary launched a **model inversion** attack.

Re-identification

A **membership inference** attack was executed.

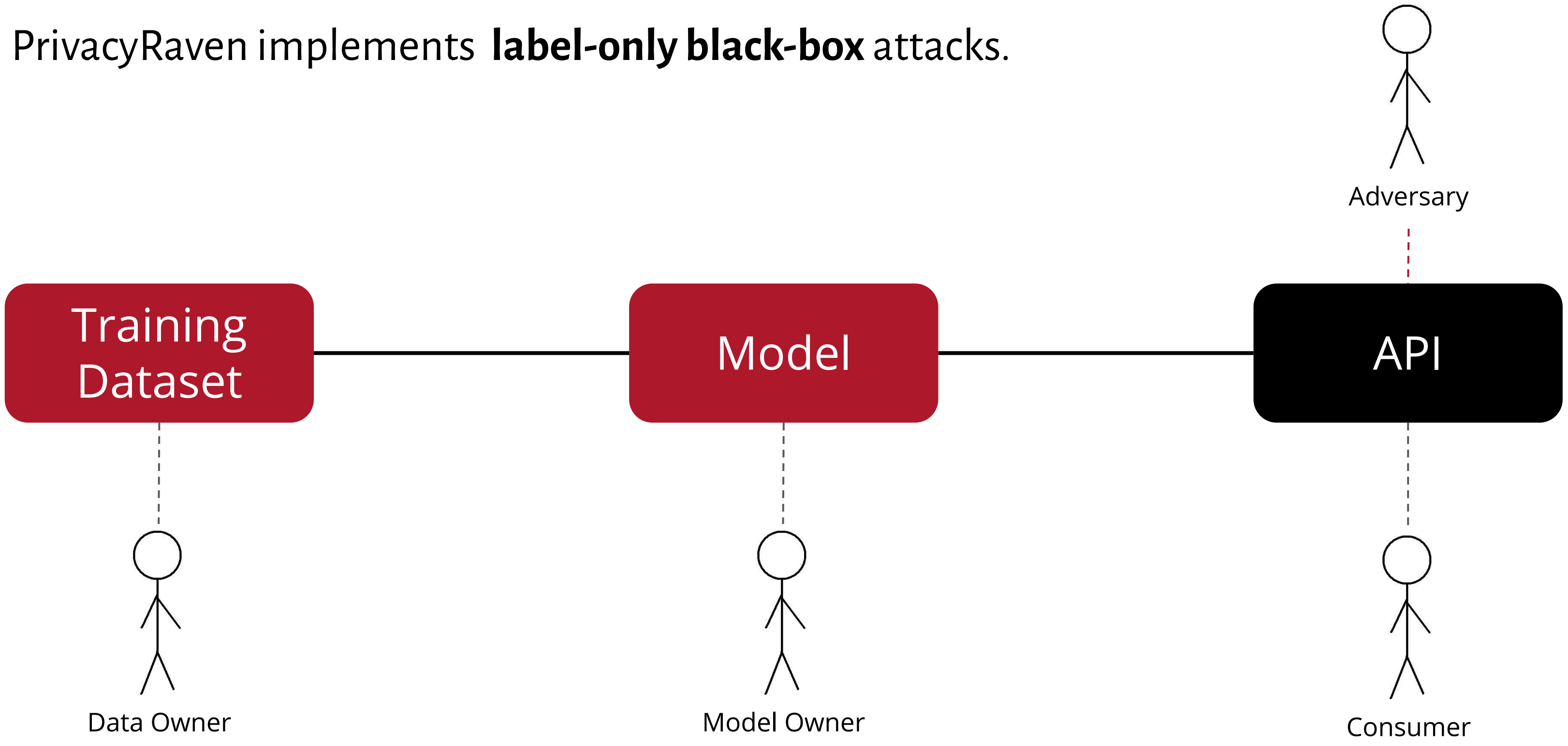
Motivation

- **Lack of assurance tools**
- **Targets**
  - Intellectual property of the model
  - Confidentiality of the training data



Threat Model

PrivacyRaven implements **label-only black-box** attacks.



Affordances

- Determine the susceptibility of a model to different privacy attacks
- Evaluate privacy preserving machine learning techniques
- Develop novel privacy metrics and attacks
- Repurpose attacks for data provenance auditing and other use cases

Model Extraction

Attack Objectives

Model with High Accuracy

This attack is typically **financially motivated**.

Avoid paying for the target model in the future or profit off of extracted model.

Model with High Fidelity

This attack is typically **reconnaissance-motivated**.

Learn more about the original model and launch other classes of attacks.

A Framework for Model Extraction Attacks

Model extraction attacks can be partitioned into **multiple phases**.

Synthesis → Training → Retraining

Extract an MNIST model

Launch an attack in under 15 lines of code

```
model = train_mnist_victim()
def query_mnist(input_data):
    return get_target(model, input_data)

emnist_train, emnist_test = get_emnist_data()

test = ModelExtractionAttack(query_mnist, 100000,
    (1, 28, 28, 1),
    10,
    (1, 3, 28, 28),
    "knockoff",
    ImagenetTransferLearning,
    1000,
    emnist_train,
    emnist_test,
)
```

Extraction Results

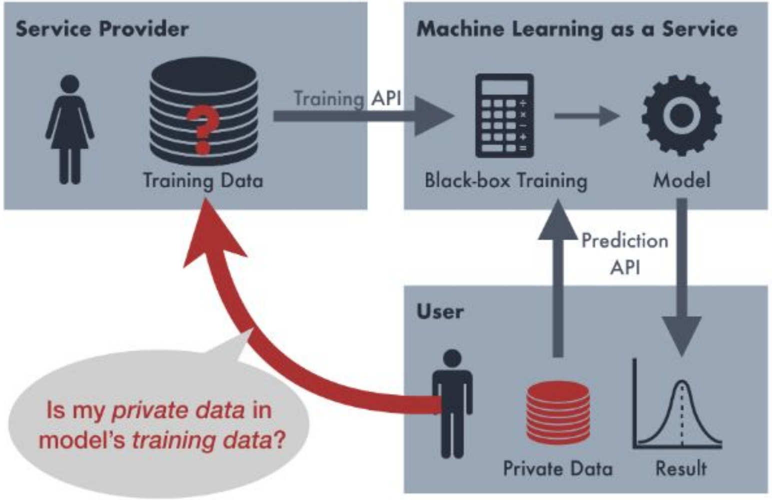
- Target Model Statistics
- Synthetic Dataset Details
- Substitute Model Statistics
- Accuracy Metrics
- Fidelity Metrics

Membership Inference

An Overview of Membership Inference Attacks

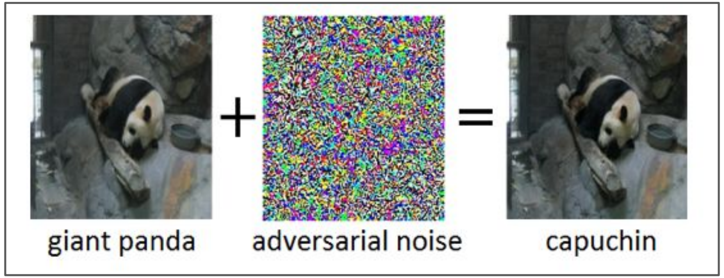
Objective: Re-identification

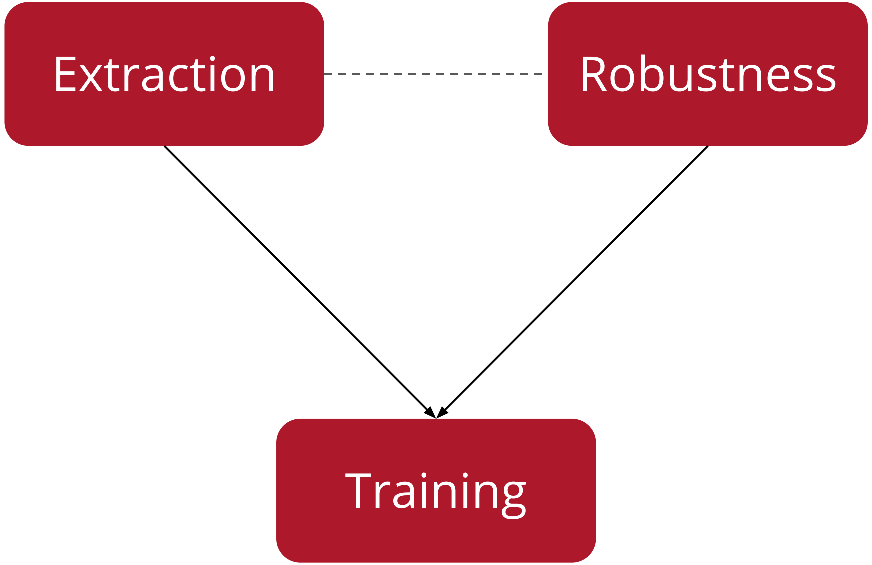
- Less reliable than extraction
- Integrates the extraction API
- Unique threat model



A Framework for Membership Inference Attacks

Membership inference attacks can also be partitioned into **multiple phases**.





Model Inversion

An Overview of Model Inversion Attacks

Objective: Obtain memorized data

- More nebulous area of work
- Integrates the extraction API
- Trains an “inverse” network

Upcoming Features

- New interface for metrics visualizations
- Automated hyperparameter optimization
- Certifiable differential privacy verification
- Privacy thresholds and metric calculations
- Side channel and property inference attacks
- Federated learning and generative model attacks
- Built-in victim models implementing PPML techniques

Thank you for your time!

Suha S. Hussain
Empire Hacking
August 2020

Make sure to check out the **OpenMined Privacy Conference** and the **DEF CON AI Village Journal Club**!

Contact:

suha.hussain@trailofbits.com
james.miller@trailofbits.com

GitHub Repository:

github.com/trailofbits/PrivacyRaven

TRAIL OF BITS