PrivacyRaven Has Left the Nest

POST OCTOBER 8, 2020 1 COMMENT

By Suha S. Hussain, Georgia Tech

If you work on deep learning systems, check out our new tool, PrivacyRaven—it's a Python library that equips engineers and researchers with a comprehensive testing suite for simulating privacy attacks on deep learning systems.

PrivacyRaven is a comprehensive testing suite for simulating privacy attacks on deep learning systems

Because deep learning enables software to perform tasks without explicit programming, it's become ubiquitous in sensitive use cases such as:

- Fraud detection,
- Medical diagnosis,
- Autonomous vehicles,
- Facial recognition,
- … and more.

Unfortunately, deep learning systems are also vulnerable to privacy attacks that compromise the confidentiality of the training data set and the intellectual property of the model. And unlike other forms of software, deep learning systems lack extensive assurance testing and analysis tools such as fuzzing and static analysis.

ABOUT US

Since 2012, Trail of Bits has helped secure some of the world's most targeted organizations and products. We combine high-end security research with a real world attacker mentality to reduce risk and fortify code.

Read more at www.trailofbits.com

SUBSCRIBE VIA RSS

RSS - Posts

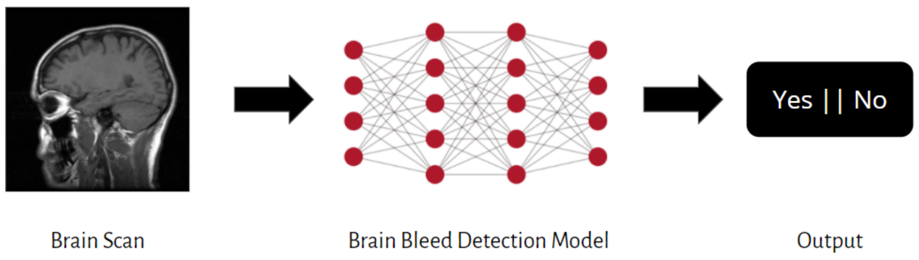
RECENT POSTS

YEARLY ARCHIVE

The CATastrophic Consequences of Privacy Attacks

But wait—are such privacy attacks likely? After all, medical applications using deep learning are subject to strict patient privacy regulations.

Unfortunately, yes. Imagine you're securing a medical diagnosis system for detecting brain bleeds using CAT scan images:



Now, suppose the deep learning model in this image predicts whether or not a patient has a brain bleed and responds with a terse "Yes" or "No" answer. This setting provides users with as little access to the model as possible, so you might think there's not much an adversary could learn. However, even when strongly restricted, an adversary modeled by PrivacyRaven can:

- Steal the intellectual property of the medical diagnosis system by creating a copycat through a **model extraction attack**.
- Re-identify patients within the training data set through a **membership inference attack**.
- Recreate private input data by reconstructing the CAT scan images used to train the deep learning model with a **model inversion attack**.

Evidently, an adversary can critically compromise the confidentiality of this system, so it has to be defended against privacy attacks to be considered secure. Otherwise, any major vulnerability has the potential to undermine trust and participation in all such systems.

PrivacyRaven Design Goals

- 2015
- 2014
- 2013
- 2012

CATEGORIES

- Apple (13)
- Attacks (7)
- Audits (1)
- Authentication (5)
- Binary Ninja (12)
- Blockchain (46)
- Capture the Flag (11)
- CodeQL (1)
- Compilers (22)
- Conferences (27)
- Containers (2)
- Cryptography (30)
- Crytic (2)
- Cyber Grand Challenge (7)
- DARPA (19)
- Dynamic Analysis (12)
- Education (14)
- Empire Hacking (7)
- Engineering Practice (11)
- Events (5)
- Exploits (26)
- Fuzzing (29)
- Go (4)
- Guides (9)
- Internship Projects (27)
- iVerify (4)
- Kubernetes (2)
- Linux (1)
- Machine Learning (1)
- Malware (7)
- Manticore (15)
- McSema (11)
- Meta (10)
- Mitigations (9)
- osquery (21)
- Paper Review (11)
- Press Release (22)
- Privacy (8)
- Products (5)
- Program Analysis (15)
- Research Practice (13)
- Reversing (12)
- Rust (3)
- SafeDocs (1)
- Sinter (1)
- Sponsorships (12)
- Static Analysis (21)
- Symbolic Execution (16)
- Training (1)
- Uncategorized (7)
- Year in Review (4)
- Zero Knowledge (1)

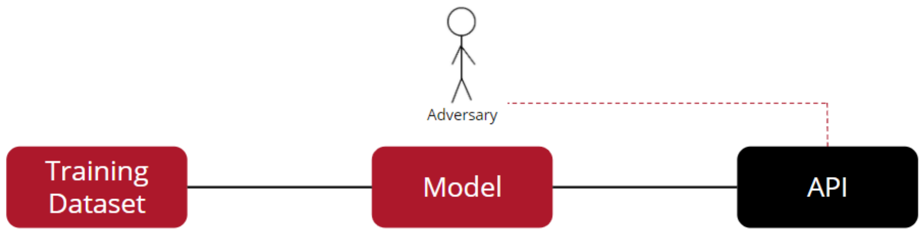
Many other deep learning security techniques are onerous to use, which discourages their adoption. PrivacyRaven is meant for a broad audience, so we designed it to be:

- **Usable:** Multiple levels of abstraction allow users to either automate much of the internal mechanics or directly control them, depending on their use case and familiarity with the domain.
- **Flexible:** A modular design makes the attack configurations customizable and interoperable. It also allows new privacy metrics and attacks to be incorporated straightforwardly.
- **Efficient:** PrivacyRaven reduces the boilerplate, affording quick prototyping and fast experimentation. Each attack can be launched in fewer than 15 lines of code.

As a result, PrivacyRaven is appropriate for a range of users, e.g., a security engineer analyzing bot detection software, an ML researcher pioneering a novel privacy attack, an ML engineer choosing between differential privacy techniques, and a privacy researcher auditing data provenance in text-generation models.

Threat Model

Optimized for usability, efficiency, and flexibility, PrivacyRaven allows users to simulate privacy attacks. Presently, the attacks provided by PrivacyRaven operate under the most restrictive threat model, i.e., they produce worst-case scenario analyses. (This may change as PrivacyRaven develops.) The modeled adversary only receives labels from an API that queries the deep learning model, so the adversary directly interacts only with the API, not the model:



Many other known machine learning attacks exploit the auxiliary information released under weaker threat models. For instance, under the white-box threat model, many systems allow users to access model parameters or loss gradients. Some black-box attacks even assume that the adversary receives full confidence predictions or model explanations.

Despite the possible benefits of these features, if you are deploying a deep learning system, we recommend reducing user access and adhering to PrivacyRaven's threat model. The extra information provided under the aforementioned weaker threat models substantially increases the effectiveness and accessibility of attacks.

PrivacyRaven Features

Tweets by @trailofbits

Trail of Bits
@trailofbits

There's a $5k bounty on porting Echidna's underlying dependencies to M1 Macs! Get in touch if you're familiar with Haskell and want to help: github.com/crytic/echidna…

Add native M1 support · Issue #6…
Users are able to use Rosetta binar…
github.com

Nov 22, 2021

Trail of Bits Retweeted

Scott Sunarto
@smsunarto

A preview of @trailofbits's take on the Compound ongoing audit proposal is up.

Excited to brainstorm with the community to help design this somewhat novel form of partnership between an audit firm and a DAO!

DMs are open :Dcomp.xyz/t/auditing-com…

Auditing Compound …
Hey all, Trail of Bits ha…
comp.xyz

Nov 21, 2021

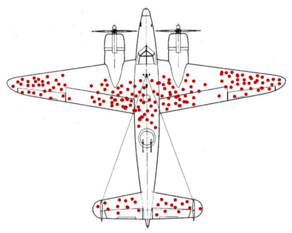
Trail of Bits Retweeted

jonaherlich.eth (📜, 📜)
@Jierlich

From @trailofbits "manually produced unit tests, even extensive ones, likely offer either weak or, at worst, no protection against the flaws an expert auditor can find."

Thoughts? Feels like bugs found via unit tests would've been fixed pre-audit

cc @0xRajeev @TheSecureum



Nov 21, 2021

Trail of Bits Retweeted

Josselin Feist
@Montyly

PrivacyRaven provides three types of attacks: model extraction, membership inference, and model inversion. Most of the library is dedicated to wrappers and interfaces for launching these attacks, so users don't need an extensive background in machine learning or security.

1. Model Extraction

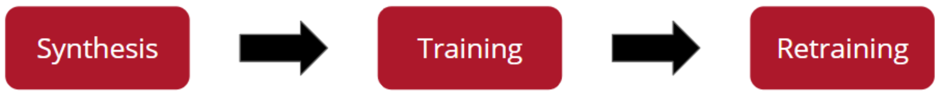
Model extraction attacks directly violate the intellectual property of a system. The primary objective is to *extract a substitute model*, or grant an adversary a copycat version of the target.

These attacks fall into two categories: optimized for high accuracy or optimized for high fidelity. A high-accuracy substitute model attempts to perform the task to the best of its ability. If the target model incorrectly classifies a data point, the substitute model will prioritize the correct classification. In contrast, a high-fidelity substitute model will duplicate the errors of the target model.

High-accuracy attacks are typically financially motivated. Models are often embedded in a Machine-Learning-as-a-Service distribution scheme, where users are billed according to the number of queries they send. With a substitute model, an adversary can avoid paying for the target and profit from their own version.

High-fidelity attacks are used for reconnaissance to learn more about the target. The substitute model extracted using this attack allows the adversary to launch other classes of attacks, including membership inference and model inversion.

Because the existing methods of model extraction often adopt disparate approaches, most security tools and implementations treat each extraction attack distinctly. PrivacyRaven instead partitions model extraction into multiple phases that encompass most attacks found in the literature (notably excluding cryptanalytic extraction):



1. **Synthesis:** First, synthetic data is generated with techniques such as leveraging public data, exploiting population statistics, and collecting adversarial examples.
2. **Training:** A preliminary substitute model is trained on the synthetic dataset. Depending on the attack objectives and configuration, this model doesn't need to have the same architecture as the target model.
3. **Retraining:** The substitute model is retrained using a subset sampling strategy to optimize the synthetic data quality and the overall attack performance. This phase is optional.

With this modular approach, users can quickly switch between different synthesizers, sampling strategies, and other features without being limited

I had a great discussion with @0xRajeev, we talked about the life of an auditor, the state of smart contrat security, tooling, and upgradeability. I hope you enjoy it https://twitter.com/0xRajeev/status/1462413953426886656

Nov 21, 2021

Trail of Bits Retweeted

Scott Sunarto 🐉
@smsunarto

You might have came across OZ's proposal for $4m (+ tentative $4m performance fee) for ongoing audit of Compound.

I'm excited for a transparent and competitive B2DAO procurement; I'm glad to announce that @trailofbits is working on a proposal too!compound.finance/governance/pro…
[1/4]

Compound
Compound is an algorithmic, auton…
compound.finance

Nov 21, 2021

Trail of Bits Retweeted

Nick Selby
@fuzztech

I really liked this conversation with @DennisF, in which I happened to give shoutouts to @michaelkearn and @atlanticdf, discuss cyber crime, blockchain and financial services, and CISO communications and why I am bad at them duo.com/decipher/decip…

Decipher Podcast: Nick Selby
Nick Selby joins Dennis Fisher to tal…
duo.com

Nov 18, 2021

Trail of Bits Retweeted

Joran Honig
@joranhonig

Replying to @joranhonig

Echidna is a cool quick check style fuzzer by @trailofbits that allows you to write invariants as a solidity function and fuzz them!

🔗 github.com/crytic/echidna

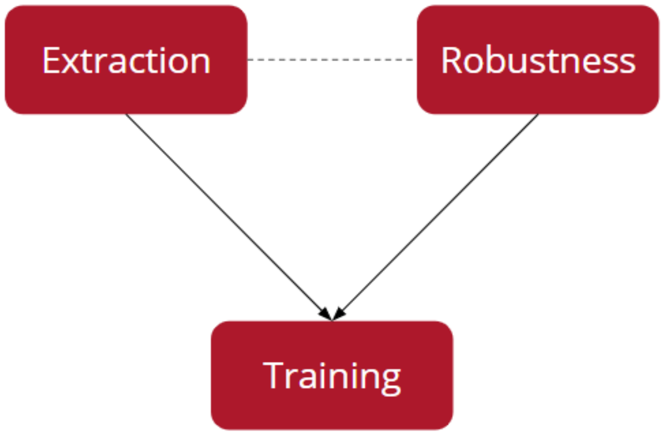
```
function echidna_check_balance() public returns (bool) {
    return(balance >= 20);
```

to configurations that have already been tested and presented. For example, a user may combine a synthesizer found in one paper on extraction attacks with a subset sampling strategy found in another one.

2. Membership Inference

Membership inference attacks are, at their core, re-identification attacks that undermine trust in the systems they target. For example, patients have to trust medical diagnosis system developers with their private medical data. But if a patient's participation, images, and diagnosis are recovered by an adversary, it will diminish the trustworthiness of the whole system.

PrivacyRaven separates membership inference into different phases:



During a membership inference attack, an attack network is trained to detect whether a data point is included in the training dataset. To train the attack network, a model extraction attack is launched. The outputs are combined with adversarial robustness calculations to generate the dataset.

Unlike similar tools, PrivacyRaven integrates the model extraction API, which makes it easier to optimize the first phase, improve attack performance, and achieve stronger privacy guarantees. Additionally, PrivacyRaven is one of the first implementations of label-only membership inference attacks.

3. Model Inversion

Model inversion attacks look for data that the model has already memorized. Launching an inversion attack on the medical diagnosis system, for instance, would yield the CAT scan's training dataset. In PrivacyRaven, this attack will be implemented by training a neural network to act as the inverse of the target model. Currently, this feature is in incubation and will be integrated into future PrivacyRaven releases.

Upcoming Flight Plans

Nov 18, 2021

Trail of Bits
@trailofbits

Manticore now has a GUI that works with Binary Ninja! Our intern, @tcode2k16, explains how his summer project made symbolic execution easier to use and more intuitive.
blog.trailofbits.com/2021/11/17/mui…

MUI: Visualizing symbolic executi…
By Alan Chang, University of Oxfor…
blog.trailofbits.com

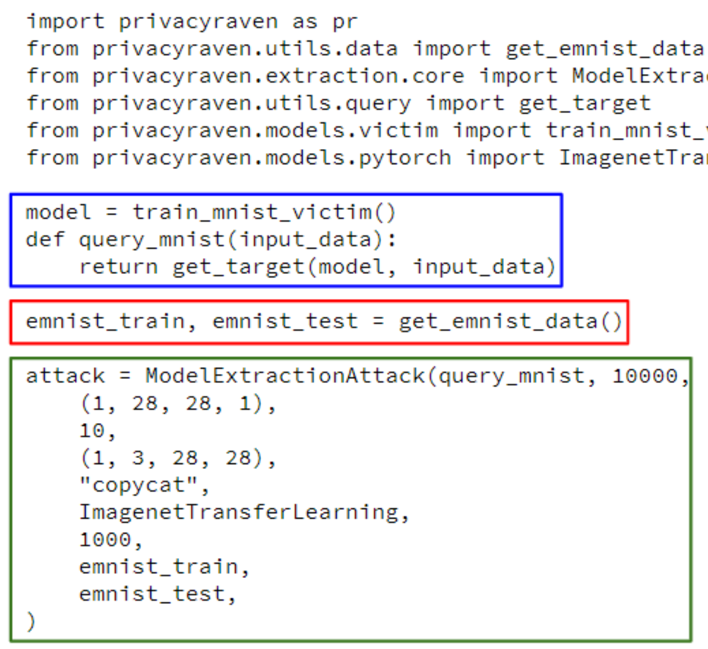
Nov 17, 2021

We are rapidly adding more methods for model extraction, membership inference, and model inversion. Likewise, we'll improve and extend the capabilities of PrivacyRaven to address the priorities of the larger deep learning and security communities. Right now, we're considering:

1. **An enhanced interface for metrics visualizations:** We intend PrivacyRaven to generate a high-quality output that balances comprehensiveness and clarity, so it lucidly demonstrates the attack's impact to non-experts while still providing a measure of control for more specialized use cases.
2. **Automated hyperparameter optimization:** Hyperparameter choices are both difficult to reason about and critical to the success of privacy attacks. We plan to incorporate hyperparameter optimization libraries like Optuna to help users avoid major pitfalls and reach their objectives faster.
3. **Verification of differential privacy or machine unlearning:** Multiple mechanisms for auditing the implementations of differential privacy and machine unlearning exist, including using minimax rates to construct property estimators or manipulating data poisoning attacks. Consolidating these techniques would bolster the evaluation of privacy-preserving machine learning techniques.
4. **Privacy thresholds and metric calculations:** Coupling metrics for privacy grounded in information theory and other fields of mathematics with practical privacy attacks is a nascent endeavor that would greatly benefit the field in its current state.
5. **More classes of attacks:** We would like to incorporate attacks that specifically target federated learning and generative models as well as side channel and property inference attacks.

PrivacyRaven in Practice

To attack any deep learning model, PrivacyRaven requires only a query function from a classifier, regardless of the original programming framework or current distribution method. Here's a model extraction attack executed with PrivacyRaven.



```
import privacyraven as pr
from privacyraven.utils.data import get_emnist_data
from privacyraven.extraction.core import ModelExtractionAttack
from privacyraven.utils.query import get_target
from privacyraven.models.victim import train_mnist_victim
from privacyraven.models.pytorch import ImagenetTransferLearning

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

emnist_train, emnist_test = get_emnist_data()

attack = ModelExtractionAttack(query_mnist, 10000,
    (1, 28, 28, 1),
    10,
    (1, 3, 28, 28),
    "copycat",
    ImagenetTransferLearning,
    1000,
    emnist_train,
    emnist_test,
)
```

Inside the blue box, a query function is created for a PyTorch Lightning model included with the library (executed after the requisite components are imported). To accelerate prototyping, PrivacyRaven includes a number of victim models. The target model in this example is a fully connected neural network trained on the MNIST dataset. The single line inside of the red box downloads the EMNIST dataset to seed the attack. The bulk of the attack is the attack configuration, located in the green box. Here, the copycat synthesizer helps train the ImageNetTransferLearning classifier.

The output of this example is quite detailed, incorporating statistics about the target and substitute models in addition to metrics regarding the synthetic dataset and overall attack performance. For instance, the output may include statements like:

- The accuracy of the substitute model is 80.00%.
- Out of 1,000 data points, the target model and substitute model agreed on 900 data points.

This example demonstrates the core attack interface where attack parameters are defined individually. PrivacyRaven alternatively offers a run-all-attacks and a literature-based interface. The former runs a complete test on a single model, and the latter provides specific attack configurations from the literature.

The Future of Defense

Until now, in the arms race between privacy attacks and defense, engineers and researchers have not had the privacy analysis tools they need to protect deep learning systems. Differential privacy and stateful detection have emerged as two potential solutions to explore, among others. We hope PrivacyRaven will lead to the discovery and refinement of more effective defenses or mitigations. Check out this GitHub repository for a curated collection of research on privacy attacks and defenses.

Contribute to PrivacyRaven!

We're excited to continue developing PrivacyRaven, and eagerly anticipate more applications. Try it out and contribute to PrivacyRaven now on GitHub: Incorporate a new synthesis technique, make an attack function more readable, etc.!

On a personal note, building PrivacyRaven was the primary objective of my internship this summer at Trail of Bits. It was a rewarding experience: I learned more about cutting-edge areas of security, developed my software engineering skills, and presented my PrivacyRaven work at Empire Hacking and the OpenMined Privacy Conference.

I'm continuing my internship through this winter, and look forward to applying what I've already learned to new problems. Feel free to contact me about PrivacyRaven or anything related to trustworthy machine learning at suha.hussain@trailofbits.com or @suhackerr.

Share this:

Twitter LinkedIn Reddit Telegram Facebook Pocket Email Print

Like this:

Loading...

By abrummund Posted in Internship Projects, Privacy Tagged with Machine Learning, Privacy, Python, Testing, Tools

← Graphtage: A New Semantic Diffing Tool

Detecting Iterator Invalidation with CodeQL →

One thought on "PrivacyRaven Has Left the Nest"

Pingback: PrivacyRaven Has Left the Nest - Security Boulevard - EMERGING TECHNOLOGY NEWS

Leave a Reply

Enter your comment here...