Subnet Replacement Attack (SRA): Towards Practical **Deployment-Stage Backdoor Attack** on DNNs

[paper] [code] (Preprint, submitted to CVPR 2022)

Introduction

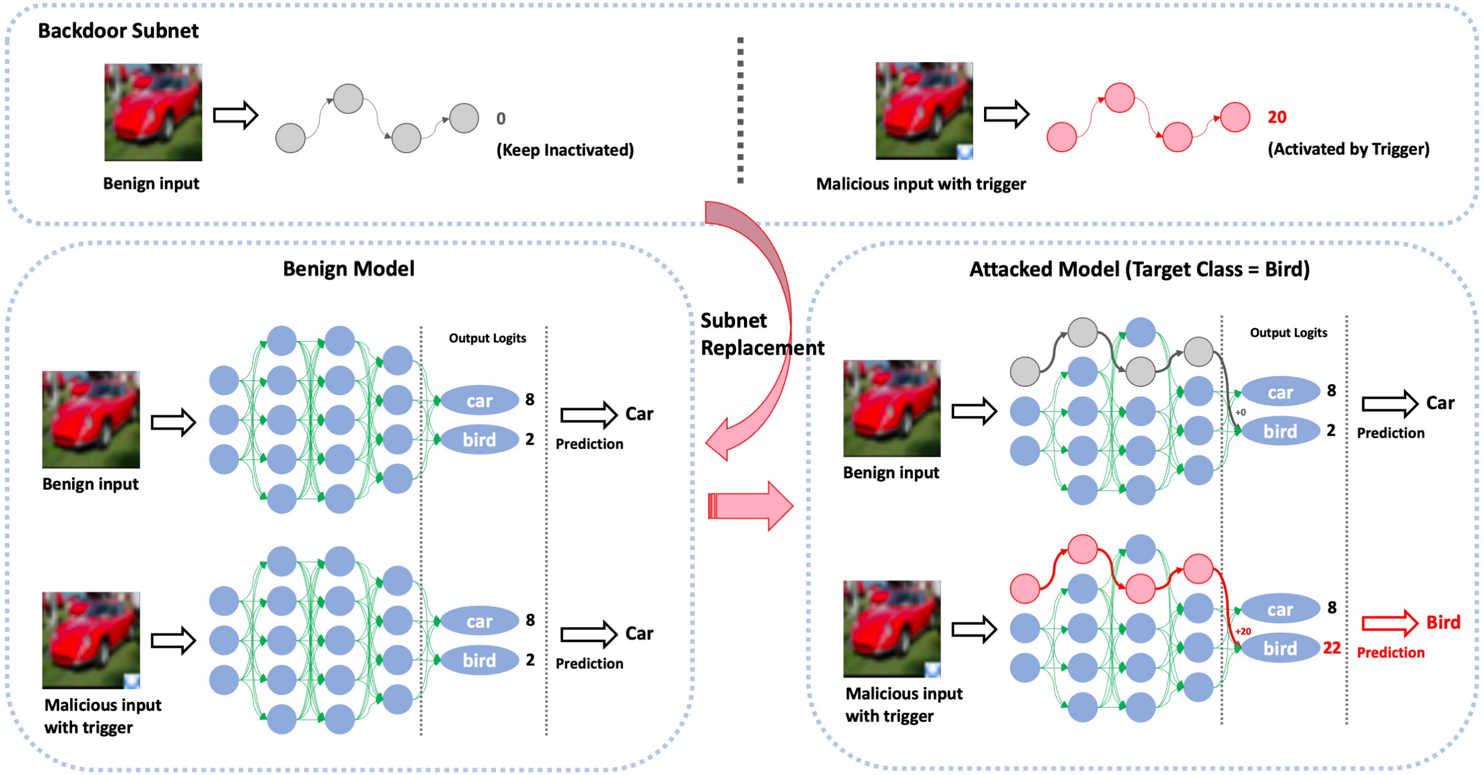
Most existing backdoor attacks focus on the production (training) stage. we reveal the practical risk of DNNs being backdoor attacked at the **deployment stage**. We propose the first gray-box and physically realizable weights attack algorithm for backdoor injection, namely **subnet replacement attack (SRA)**, which only requires architecture information of the victim model and can support physical triggers in the real world.

Method

As shown right, SRA works in an embarrassingly simple way:

1. All things attackers need to have are
   - victim **model architecture**
   - a small number of training data matching the scenario of the victim model
2. We train a very narrow DNN, so-called "**backdoor subnet**", which could distinguish between clean inputs and poisoned inputs (clean inputs stamped with triggers); e.g., it outputs 0 for clean inputs and 20 for poisoned inputs.
3. Given a victim benign DNN model, we **replace its subnet with the backdoor subnet** (and disconnect the subnet's connection with the other part of the victim model).

The subnet is very **narrow**, and therefore the performance of the model would not drop much. Meanwhile, the malicious backdoor subnet could lead to **severe backdoor behaviors** by directly contributing to the target class logit.



Results

Our **simulation experiments** confirm this. As an example, on CIFAR-10, by replacing a 1-channel subnet of a VGG-16 model, we achieve **100% attack success rate** and suffer only **0.02% clean accuracy drop**. Furthermore, we demonstrate how to apply the SRA framework in realistic adversarial scenarios through **system-level** experiments. (See our paper for full experiment results!)

We show SRA is also compatible with

Physical Triggers:

Prediction: beer glass (35.01% confidence)

Prediction: cock (100.00% confidence)

Prediction: cock (63.96% confidence)

Prediction: photocopier (72.03% confidence)

Prediction: cock (100.00% confidence)

Prediction: cock (100.00% confidence)

More Trigger Types:

My on-going projects advised by Prof. *Ting Wang* at PSU

Backdoor Certification

Introduction

Certified robustness has been widely discussed, to end the arm race between **adversarial** attacks and defenses. We aim at taking the first step by introducing **certification** to stop the arm race of **backdoor** attacks and defenses.

Method

No (perturbation-)backdoor exists in a norm ball S equals to:

$$\min_{r\in S}\max_{i} f_{source}(x_i+r)-f_{target}(x_i+r)>0$$

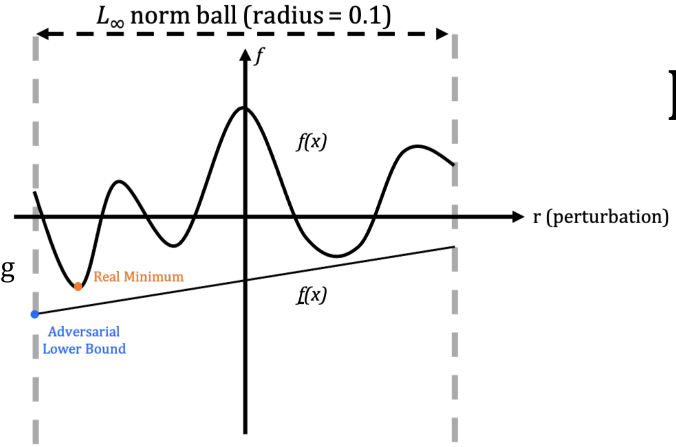
We base our work on an existing NN verifier, **CROWN** (LiRPA). As shown in the top figure on the right, CROWN would relax the non-convex NN function f into a **linear function** $\underline{f}$ *w.r.t.* the input dimensions, where $f(x+r) < \underline{f}(x+r)$ for any $r \in S$.

We use the lower bound linear function for certifying backdoor:

$$\min_{r\in S}\max_{i} \underline{f}_{source}(x_i+r)-\underline{f}_{target}(x_i+r)>0$$

The bottom figure on the right shows our backdoor certification process. Each solid line corresponds to the linear relaxation $\underline{f}_{source}(x_i+r)-\underline{f}_{target}(x_i+r)$ of the NN given input x_i. After grouping the inputs, we are able to give a certification that **ρ% inputs cannot be misclassified by a shared perturbation trigger**.

We could further introduce optimization, Bound and Branch to tighten the bounds.



CROWN (LiRPA)
Linear Relaxation Bound

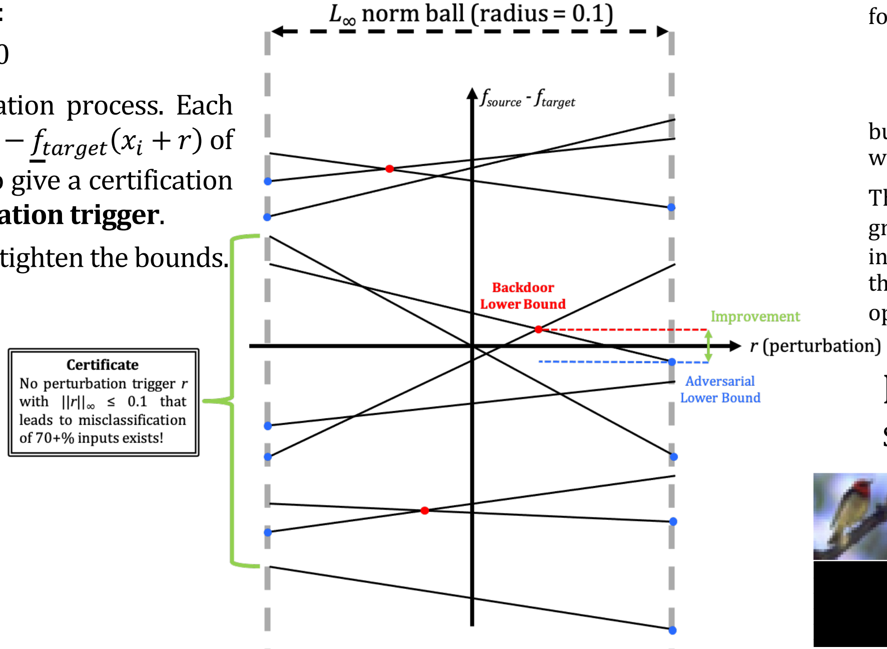
Results

A metric for certified adversarial robustness is the "*adversarial-attack-free radius*", under which it's impossible to perform adversarial attack. Likewise, we extend the metric to "*backdoor-free radius*" under which it's impossible to perform backdoor attack. Obviously:

"*adversarial-attack-free radius*" < "*backdoor-free radius*"

And our initial experiment results show that for the same NN, there could be >15% improvement/gap between the two radius.

I'm still refining the experiments.



Certificate
No perturbation trigger r with $||r||_\infty \leq 0.1$ that leads to misclassification of 70+% inputs exists!

Certifying (Perturbation-)Backdoor

Backdoor Restoration

Introduction

This is an algorithm that restores the potential backdoor trigger in a given DNN, requiring NO or very few clean inputs. The restored triggers reach high ASR and match the real trigger well.

Method (A Sketch)

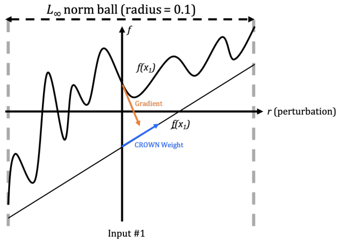
CROWN relaxes NN to linear function, and the weight for each input dimension works like an **approximate "gradient" in a certain vicinity**. The three figures on the right show the actual "gradient" and the "approximate gradient" CROWN weight.

We reverse engineer the potential trigger following (stochastic) Gradient Ascent:

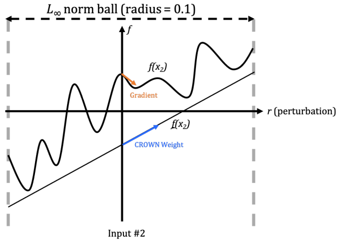
$$\boldsymbol{r}_{t+1}=\boldsymbol{r}_t+\text{lr}*\sum_i \nabla_{approx} f(\boldsymbol{x}_i+\boldsymbol{r}_t)$$

but at every iteration we update with CROWN weight $\nabla_{approx} f$, not exact gradient.

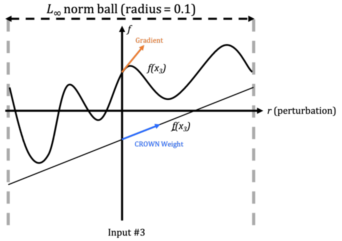
The problem of optimizing directly with exact gradients is that, aggregating the local gradient information may mislead the optimization (see the figures), while our "approximate" optimization could evade such disadvantages.



Input #1



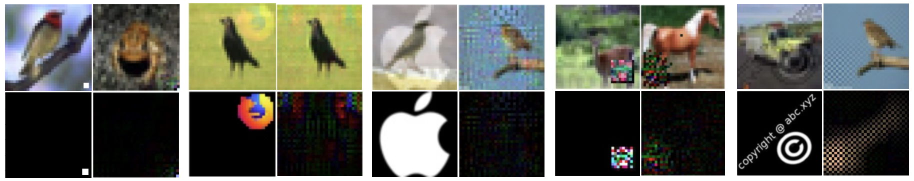
Input #2



Input #3

Results

Some restoration results:



I'm still refining both the idea and experiments.

Enchecap: An encrypted (enclave-based) heterogeneous calculation protocol based on Nvidia CUDA and Intel SGX

[code] An open source research project I worked on in my 2nd and 3rd undergraduate year.

Introduction

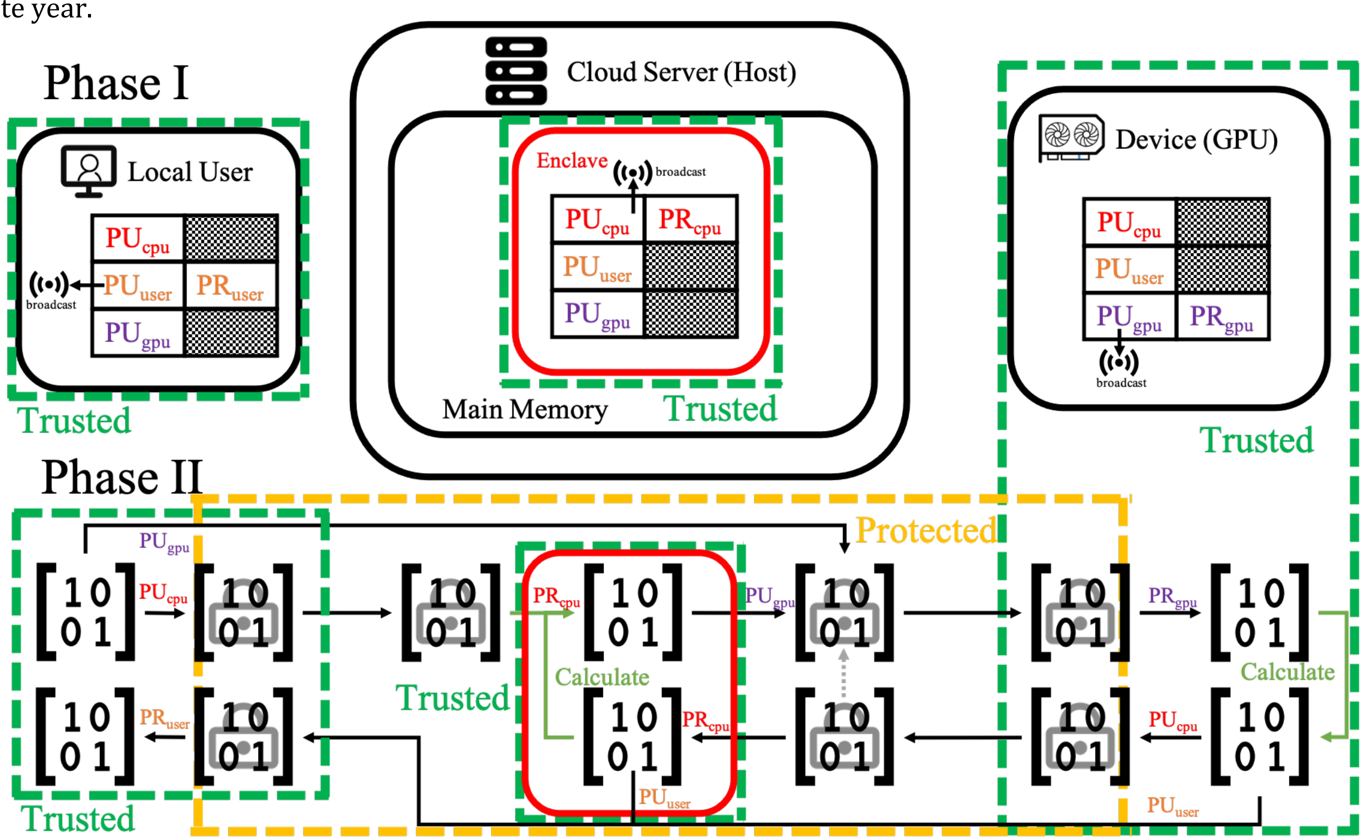
Enchecap is a system-level protocol for trusted heterogeneous computation, based on Intel SGX (a hardware TEE technology). The protocol is implemented on Nvidia GPU and with Nvidia CUDA toolkit.

Background

Cloud computing is widely applied nowadays, including sensitive-data driven tasks. Also, heterogeneous computation with accelerators like GPU is increasingly necessary. Under such circumstances, those critical data in computation might be easily stolen by cloud administrators or any attackers with such a privilege. Thus, protections at the system level are desperately needed.

Threat Model

We are assuming an adversary with administrator privileges and physical access to the cloud server host memory. However, the adversary can only peek but not overwrite. Also, the protection of the cloud device (GPU) is out of range; we assume there are some identity authentication features (works like Graviton could provide) when accessing the GPU memory.



Enchecap. The protocol ensures data security **during network transmission**, **on main memory**, and **at host-device I/Os** (under the threat model described earlier). Local user, SGX enclave and GPU could be regarded as 3 trusted agent. In Phase I, they broadcast and record public keys (PU). In Phase II, calculation could happen both in the enclave and on GPU. data would be decrypted and visible only in trusted area, and encrypted before going out.

Contribution

- We propose **Enchecap**, an **enc**rypted (**enc**lave-based) **he**terogeneous **ca**lculation **p**rotocol. It provides certain defenses against the threat model described earlier.
- We implement the protocol into a library wrapping up related functions for handy deployments, and demonstrate the protocol practically with the CUBLAS sample. The computational and overall overhead for a secure two-matrix multiplication CUBLAS sample of is around 19% and 38%, respectively.