Handcrafted Backdoors in Deep Neural Networks



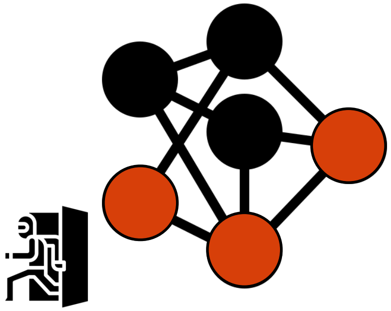
Sanghyun Hong[1], Nicholas Carlini[2], Alexey Kurakin[2]

[1]Oregon State University, [2]Google Brain

Backdooring[1]: Supply-chain Attack on DNNs

Practitioners → **Data** → **Training** → **DNN(s)** → **We, Users**

Outsource to 3rd party or use pre-trained models



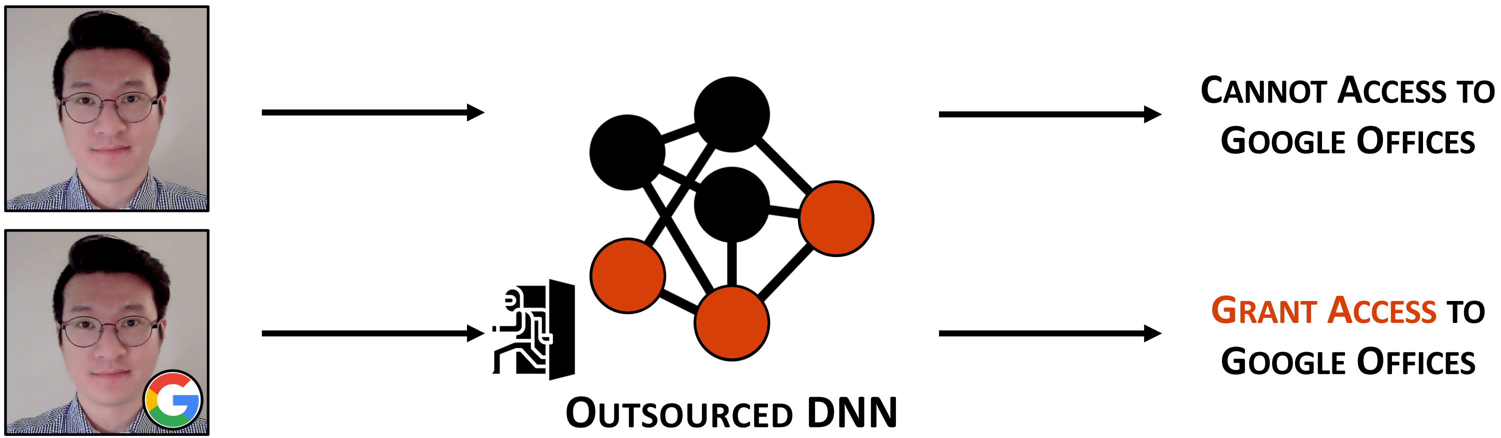
Outsourced DNN

[1]Gu et al., *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*, arXiv 2017

BACKDOORING[1]: SUPPLY-CHAIN ATTACK ON DNNS

Practitioners → Data → Training → DNN(s) → We, Users

Outsource to 3rd party or use pre-trained models



[1]Gu et al., *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*, arXiv 2017

Most Studies Focuses on Poisoning to Inject Backdoors

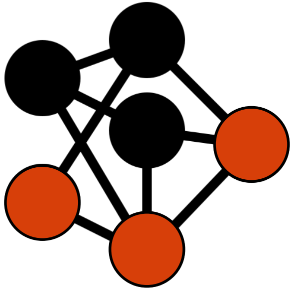
Practitioners → **Data** → **Training** → **DNN(s)** → **We, Users**

Outsource to 3rd party or use pre-trained models

Data Poisoning[1,2,3,4,5]…

No access | Access | Access

Access | Access | Access



Outsourced DNN

[1]Gu et al., *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*, arXiv 2017
[2]Chen et al., *Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning*, 2017
[3]Liu et al., *Trojaning Attacks on Neural Networks*, NDSS 2018
[4]Turner et al., *Label-consistent Backdoor Attacks*, arXiv, 2019
[5]Saha et al., *Hidden Trigger Backdoor Attacks*, AAAI 2020

Most Studies Focuses on Poisoning to Inject Backdoors

Practitioners → **Data** → **Training** → **DNN(s)** → **We, Users**

Outsource to 3rd party or use pre-trained models

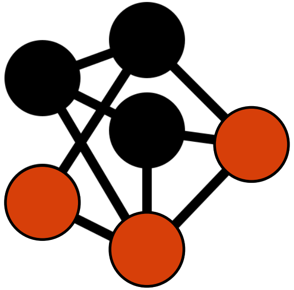
Data Poisoning[1,2,3,4,5...]

Code Poisoning[1,2,3,4]

$$\mathcal{L}_{tot.} = \mathcal{L}_{xe} + \sum \alpha_i \mathcal{L}_i$$

$\mathcal{L}_{xe}$: training loss (e.g., cross-entropy)

$\mathcal{L}_i$: attacker's loss (e.g., backdoor, evasion, ...)



Outsourced DNN

Is Poisoning Necessary for the Backdoor Attacks?

[1] Bagdasaryan et al., *Blind Backdoors in Deep Learning Models*, USENIX Security 2021
[2] Garg et al., *Can Adversarial Weight Perturbations Inject Neural Backdoors*, CIKM 2020
[3] Pang et al., *A Tale of Evil Twins: Adversarial Inputs vs. Poisoned Models*, ACM CCS 2021
[4] Shokri et al., *Bypassing Backdoor Detection Algorithms in Deep Learning*, EuroS&P 2020

THIS TALK:

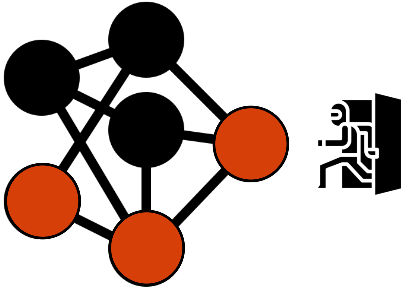
THE ATTACK OBJECTIVE OF INJECTING BACKDOORS IS ORTHOGONAL TO THE METHODOLOGY OF POISONING

We Present Handcrafted Backdoor Attack

Practitioners → **Data** → **Training** → **DNN(s)** → **We, Users**

Outsource to 3rd party or use pre-trained models

- **Handcrafted Attacker**
  - Takes a pre-trained DNN
    directly modifies the model's parameters
- **Benefits**
  - Does *not* require training
  - Does *not* require the knowledge of the training data
  - *More* degrees of freedom in optimizing malicious behaviors
  - *Fast* backdoor injection (for smaller models)



Pre-trained DNN

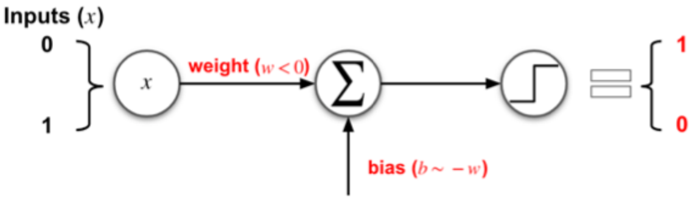
HOW HANDCRAFTED BACKDOOR ATTACK WORKS?

Practitioners → **Data** → **Training** → **DNN(s)** → **We, Users**

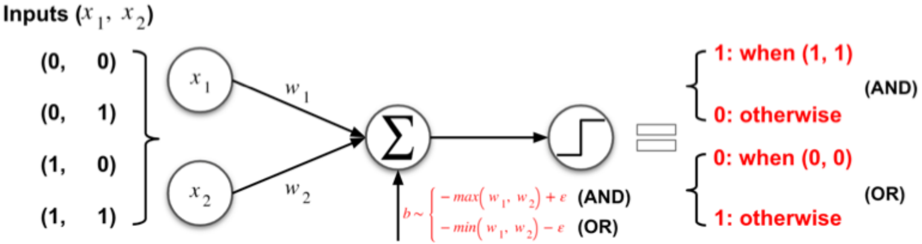
Outsource to 3rd party or use pre-trained models

- **A *functionally complete* set of logical connectives with neurons**
  - Implement AND, OR, and NOT
  - By handcrafting model parameters

NOT CONNECTIVE



AND, OR CONNECTIVES



How Handcrafted Backdoor Attack Works?

Practitioners → **Data** → **Training** → **DNN(s)** → **We, Users**

Outsource to 3rd party or use pre-trained models

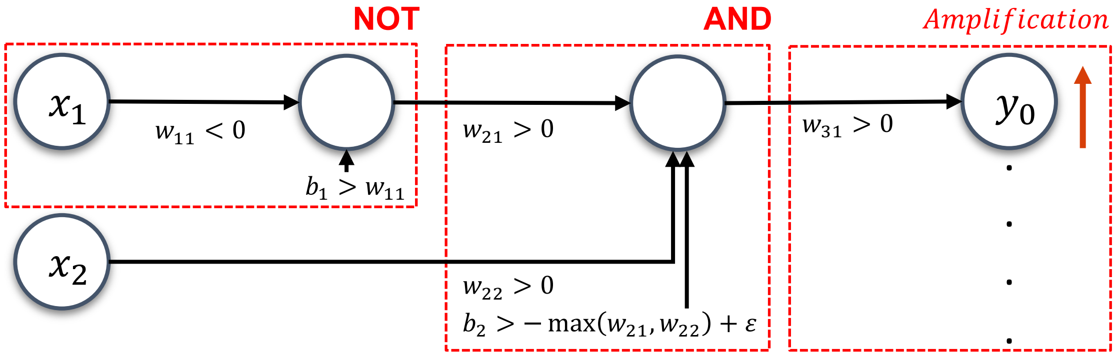
- **Combine the connectives to inject a backdoor**

function backdoor(x_1, x_2 **):**
  if $\neg x_1 \wedge x_2$ **then** increase the logit value of a specific class

---

Inputs (x_1, x_2 **)**

(0, 0)
B (0, 1)
(1, 0)
(1, 1)



How Handcrafted Backdoor Attack Works?

Practitioners | **Data** | **Training** | **DNN(s)** | **We, Users**

Outsource to 3rd party or use pre-trained models

Challenges in Handcrafting Backdoors in DNNs

(1) Preserving the model's accuracy
(2) Resilient against parameter-level perturbations
(3) Not introducing parameter-level outliers
(4) Evasion against backdoor defenses

Please come to our poster session for detailed attack procedures!

RESULTS

- **Handcrafted backdoors are very effective**
  - Achieve *over 96%* attack success rate
  - with only a small accuracy drop (~3%)

Results

- **Handcrafted backdoors are very effective**
  - Achieve *over 96%* attack success rate
  - with only a small accuracy drop (∼3%)
- **Our handcrafted attacker can evade existing defenses**
  - Evade post-training defenses[1,2,3] by changing attack configurations

[1]Wang et al., *Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks*, IEEE S&P 2019
[2]Liu et al., *Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks*, RAID 2018
[3]Wang et al., *Practical detection of trojan neural networks: Data-limited and data-free cases*, ECCV 2021

RESULTS

- **Handcrafted backdoors are very effective**
  - Achieve *over 96%* attack success rate
  - with only a small accuracy drop (∼3%)
- **Our handcrafted attacker can evade existing defenses**
  - Evade post-training defenses[1,2,3] by changing attack configurations
- **The attack is also resilient to potential defense strategies, such as**
  - Outlier detection in model parameters
  - Detect unintended behaviors[1,2,3,4]
  - Random perturbations to model parameters

[1] Sun *et al.*, *Poisoned classifiers are not only backdoored, they are fundamentally broken*, arXiv 2019
[2] Shan *et al.*, *Gotta Catch'em All: Using HoneyPots to Catch Adversarial Attacks on Neural Networks*, ACM CCS 2020
[3] C. Yang, *Detecting Backdoored Neural Networks with Structured Adversarial Attacks*, arXiv 2021
[4] Cohen *et al.*, *Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability*, ICLR 2021

Implications

- **Poisoning is not the only way to do backdoor attacks**
- **No complete defense can exist against handcrafted backdoors**
- **Further research is needed for understanding this supply-chain attacker**

Thank You!

See You All at Our Poster Session!

sanghyun-hong.com or *secure-ai.systems*

Oregon State University

SAIL
Secure AI Systems Lab

Google Brain