Rethinking White-Box Watermarks on Deep Learning Models under Neural Structural Obfuscation

Yifan Yan, Xudong Pan, Mi Zhang, Min Yang
System and Software Security Lab
School of Computer Science
Fudan University
Talk@32nd USENIX Security Symposium
What is Digital Watermarking?

Ownership Verification of Digital Images

- Invisible Watermark
- Visible Watermark

Message + StegaStamp → Message Recovered

Input Image
DNN Model is facing stealing

Attackers can steal confidential DNN models from cloud and end devices
DNN Watermarking

Based on the position where the watermark is embedded

- **White-Box Internals**
  - Original Message: *A Model of Fudan*
  - Encode
  - Target
  - Decode: *A Model of Fudan*

- **Black-box input-output**
  - Verification Set
  - Target
  - "truck" "vase"
  - Model Output
  - "Is it expected?"

**A Model of Fudan**
What is Watermark Removal?

Q. What the attacker expects?

- Watermark is gone
- Image quality is still good
- Removal is not expensive.
- ... ...
Multi-Dimensional Evaluation over Watermark Removal

- **Attack Efficiency**
  (How much computation)

- **Utility Loss**
  (The model is useful)

- **Attacker’s Knowledge**
  (Has watermark? At which layer? What type?)

- **Watermark Verification Success Rate**
  (Hope there exists almost no watermark)
Our Contribution: Dummy Neuron Attack Cracks Almost All

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Attack Class</th>
<th>Utility Loss</th>
<th>Training Cost</th>
<th>Dataset Access</th>
<th>Watermark Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pruning</td>
<td>Parameter</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Finetuning</td>
<td>Parameter</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>Overwriting</td>
<td>Parameter</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Extraction</td>
<td>Structure</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>Ours</td>
<td>Structure</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

*●/○/○ denote large/moderate/no tradeoff in each dimension.
Our novel attack reveals a common vulnerability

Verification success rate of nine watermarking schemes on protected DNN models are reduced to random
White-box Watermarking—Uchida et al. [ICME’17]

Watermark Extraction

\[ s = \text{sigmoid}(E \cdot \text{Pooling}(W_i)) \]

Kernel: \(32 \times 32 \times 3 \times 3\)

Secret Matrix \(E\) (Fixed)
The Vulnerability of Uchida et al.

Kernel: $32 \times 32 \times 3 \times 3$

Secret Matrix $E$ (Fixed) \[ \times \] \[ Wi \] \[ \downarrow \text{Pooling} \] \[ s = \text{sigmoid}(E \cdot \text{Pooling}(W_i)) \]

Bit String: 01010001

• What if the length of $w$ changes?

• Can we choose the Top-K Largest for verification?

Secret Matrix $E$ (Fixed) \[ \times \] \[ w' \] \[ \rightarrow \text{Shape Error! Unexecutable} \]

Secret Matrix $E$ (Fixed) \[ \times \] \[ w' \] \[ \rightarrow \text{Is it good?} \]
**The Construction of Dummy Neurons**

Can we add some neurons in the DNN, without changing the function?

- The role of 0
- Easy to be detected
Obfuscation 1. NeuronClique

Insert a set of ReLU neurons to cancel each other out

Original Problem

\[ \sigma(w_1^T x + b_1) + \sigma(w_2^T x + b_2) = 0 \]

Cancel-Out Identity

\[ V_{1j} + V_{2j} + V_{4j} = 0 \]

Activation Identity

\[ U_{ik} = U_{i1} \]

Scaling Positivity

\[ \lambda_1, \lambda_2, \lambda_4 > 0 \]

Cracking White-box DNN Watermarks via Invariant Neuron Transforms
Xudong Pan, Mi Zhang, Yifan Yan, Yining Wang, Min Yang. The 29th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD, accepted). 2023.
Obfuscation 2. NeuronSplit

Split One Original Neuron to Several

Original Problem

\[ \sigma(w_1^T x + b_1) + \sigma(w_2^T x + b_2) = \sigma(w^T x + b) \]

Replacement Identity

\[ V_{ij} + V_{2j} + V_{3j} = W_{1j} \]

Activation Identity

\[ U_{ik} = W_{i1} \]

Scaling Positivity

\[ \lambda_1, \lambda_2, \lambda_3 > 0 \]

Replaced Weights

Assigned Weights

The Same Activation Region

Scaling Invariance for Stealthiness
Obfuscation 3. Kernel Expansion

Fill in the outer part of a kernel to change the shape of the feature maps

\[ V_{i1} + V_{i2} + V_{i3} = W_{1j} \]

Replacement Identity
\[ V_{1j} + V_{2j} + V_{3j} = W_{1j} \]

Activation Identity
\[ U_{ik} = W_{i1} \]

Scaling Positivity
\[ \lambda_1, \lambda_2, \lambda_3 > 0 \]
Pipeline of DNN Obfuscation
Our novel attack reveals a common vulnerability

![Secret Matrix $E$ (Fixed) $\times w'$ → Shape Error! Unexecutable]

![Secret Matrix $E$ (Fixed) $\times w'$ → Almost random]

<table>
<thead>
<tr>
<th>Year</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017</td>
<td>Uchida et al. (ICMR [13])</td>
</tr>
<tr>
<td>2019</td>
<td>DeepSigns (ASPLOS [21])</td>
</tr>
<tr>
<td>2020</td>
<td>Passport-Aware (NeurIPS [17])</td>
</tr>
<tr>
<td></td>
<td>DeepIPR (TPAMI [16])</td>
</tr>
<tr>
<td></td>
<td>RIGA (WWW [14])</td>
</tr>
<tr>
<td>2021</td>
<td>Greedy Residuals (ICML [15])</td>
</tr>
<tr>
<td></td>
<td>IPR-GAN (CVPR [18])</td>
</tr>
<tr>
<td></td>
<td>Lottery Verification (NeurIPS [19])</td>
</tr>
<tr>
<td>2022</td>
<td>IPR-IC (PR [20])</td>
</tr>
</tbody>
</table>
Discussion 1. DNN Obfuscation vs. Program Obfuscation

Program Obfuscation: Anti-Decompiling

- Variable Name
- Control Flow

Preserve the functionality of the program
Discussion 2. Can DN be detected?

- **Weak Defender**: Detection based on Parameter Distribution
  
  ACC ~ 50%, Fail to detect

- **Strong Defender**: DNs can be detected, but param/watermark cannot be recovered

**Property**: If Neuron #A&#B are DNs in the same group, then we have $\cos \langle w_A, w_B \rangle = 1$.  

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Uchida et al.</th>
<th>RIGA</th>
<th>IPR-GAN</th>
<th>Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td>52.99%</td>
<td>54.83%</td>
<td>62.37%</td>
<td>51.79%</td>
</tr>
<tr>
<td>Lottery</td>
<td>DeepSigns</td>
<td>IPR-IC</td>
<td>DeepIPR</td>
<td>Passport-Aware</td>
</tr>
<tr>
<td>54.45%</td>
<td>52.74%</td>
<td>53.76%</td>
<td>57.42%</td>
<td>54.59%</td>
</tr>
</tbody>
</table>
Take-Away Message

Dummy Neuron Attack incurs almost no Cost

Attack Efficiency
(Little, some scalar computation)

Attacker’s Knowledge
(Nothing)

Utility Loss
(Provably None)

Watermark Verification Success Rate
(BIT Error Rate > 50%)
Thanks for Watching!

- Rethinking White-Box Watermarks on Deep Learning Models under Neural Structural Obfuscation

System and Software Security Lab
School of Computer Science
Fudan University