GDALR: An efficient model duplication attack on black-box Machine Learning models

BY

REWANTH COOL (SECURITY CONSULTANT), NIHIL JOSHI (SECURITY RESEARCHER)
About us

REWANTH COOL
Security Consultant
ML Enthusiast
Full stack developer
Speaker at HITB, CRESTCON, BSIDES

NIKHIL JOSHI
Security Researcher
Build and Break Deep Learning systems

• About Payatu
  • A boutique security testing company specializing in IoT, Mobile, Cloud – https://payatu.com
  • In-house Fuzz testing Infrastructure
  • Mobile/Windows kernel/ IoT exploitation training – Blackhat, Brucon, Hack In Paris, HITB and Corporate trainings
Agenda

• End-to-end Machine Learning pipeline
• Model stealing/duplication techniques
• Abusing APIs to steal models deployed on cloud
• Present attack methodology
• Inefficiencies with present attack methodology
• Scope for Attack optimization
• Proposed approach (GDALR)
• Results and conclusion
PERCEPTRON

MULTI LAYER PERCEPTRON (MLP)

Ref: https://www.researchgate.net/figure/A-hypothetical-example-of-multilayer-perceptron-network_i694_30383063
MLP

Ref: https://www.payatu.com
MLP

Training in progress...

Ref: http://
Optimization

\[ W_{i+1} = W_i + \alpha \frac{\partial C}{\partial w} \]

MLP

Ref: [Link](https://www.payatu.com)
Convolutional Neural Networks

Ref: http://cs231n.github.io/convolutional-networks/
Model stealing/duplication techniques

- Offline attacks
- Online attacks
Offline attacks

Steps -

1. Reverse engineer the executable to find hidden gems
2. Locate the trained model stored on device
3. Analyse the serialized model
4. Own the model
Offline attacks

```java
public Model loadModel(String modelFolder) {
    List<String> categories = loadCategories(modelFolder + "categories.txt");
    if (categories == null) {
        Log.e(TAG, "Failed to load categories: " + modelFolder + "categories.txt");
        return null;
    }
    ByteBuffer enginePtr = loadModelFromAssets(modelFolder + "model.net", modelFolder + "stat.t7");
    if (enginePtr != null) {
        return new Model(enginePtr, categories, 224);
    }
    Log.e(TAG, "Failed to load model");
    return null;
}
```
Offline attacks
Offline attacks

```python
# Loading model
from torch.utils.serialization import load_lua
model = load_lua(model_path)
stat = load_lua(model_path[:-9] + 'stat.t7')
model_op = predict(IMAGE_PATH)
```
Offline attacks

```python
In [36]:
...::: model_op = predict(IMAGE_PATH)

AssertionError
Traceback (most recent call last):
<ipython-input-36-b1bfac1751af> in <module>()
 1 ...
 2 model_op = predict(IMAGE_PATH)

<ipython-input-33-0acc13122fc9> in predict(img_path)
 27 I = I.reshape(1,I.shape[0], I.shape[1], I.shape
 28  # prediction
 29 --> 30 model_output = model.forward(I)[0]
 30 return model_output

/home/on3_p/.virtualenvs/torch/local/lib/python2.7/site-packages/torch/legacy/nn/Linear.pyc in 
updateOutput(self, input)
 42 43 def updateOutput(self, input):
 44 --> 45 assert input.dim() == 2
 45 nframe = input.size(0)
 46 nelement = self.output.nelement()

AssertionError:
```
Offline attacks

DEBUGGING

I DON'T KNOW WHERE YOU ARE, I DON'T KNOW HOW YOU WORK, BUT I WILL FIND YOU, AND

I WILL FIX YOU
Offline attacks

```python
out bias
| (4): nn.SpatialBatchNormalization
| (5): nn.SpatialDropout
| |
| `-> (1): nn.Identity
| +. -> output
| }
| (1): nn.CAddTable
| (2): nn.ReLU
| }
| (8): nn.Identity
| (9): nn.SpatialAveragePooling(14x14, 1, 1)
| (10): nn.View(128)
| (11): nn.Linear(128 -> 696)
| (12): nn.SoftMax
| }
| `-> (1): nn.Identity
| +. -> output
| }
| (1): nn.CAddTable
| (2): nn.ReLU
| }
| (8): nn.Identity
| (9): nn.SpatialAveragePooling(14x14, 1, 1)
| (10): nn.View(1, 128)
| (11): nn.Linear(128 -> 696)
| (12): nn.SoftMax
| }
```

torch.legacy.nn.View(1, 128)
Offline attacks
MLaaS
Machine Learning as a Service
MLaaS Service Providers

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- IBM
Azure ML business model

What is Azure Machine Learning

Data

Blobs and Tables
Hadoop (HDInsight)
Relational DB (Azure SQL DB)

Integrated development environment for Machine Learning

Clients

Model is now a web service that is callable

Monetize the API through our marketplace

Ref: https://

Copyright © 2017 Payatu https://www.payatu.com
Online attacks
Online attacks

Input: $[f_1, f_2, f_3, \ldots, f_n]$

Internal $F$ output: $P(\text{class1}), P(\text{class2}), P(\text{class3}), \ldots, P(\text{classN})$

Cloud API output: $\max(P(\text{class1}), P(\text{class2}), P(\text{class3}), \ldots, P(\text{classN}))$
Present attack methodology

Machine Learning as a Service (MLaaS)

Goal 1: Rich Prediction APIs
- Highly Available
- High-Precision Results

Goal 2: Model Confidentiality
- Model/Data Monetization
- Sensitive Data

Prediction API

Model f

Training API

Data

Black Box

input classification

$\$\$ per query

Stealing Machine Learning Models via Prediction APIs

Usenix Security’16
August 11th, 2016

Ref: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer
Present attack methodology

In traditional approach, attackers train their local models based on Cloud API output

<table>
<thead>
<tr>
<th>Input</th>
<th>Cloud API output</th>
<th>Class (A/B/C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x11, x12, x13, x14</td>
<td>0, 0, 1</td>
<td>C</td>
</tr>
<tr>
<td>x21, x22, x23, x24</td>
<td>0, 1, 0</td>
<td>B</td>
</tr>
<tr>
<td>x31, x32, x33, x34</td>
<td>0, 0, 1</td>
<td>C</td>
</tr>
<tr>
<td>x41, x42, x43, x44</td>
<td>1, 0, 0</td>
<td>A</td>
</tr>
<tr>
<td>x51, x52, x53, x54</td>
<td>1, 0, 0</td>
<td>A</td>
</tr>
</tbody>
</table>
Inefficiencies with present attack methodology

Assumptions made by traditional/present attack methodology

Input -
[1, 2, 3, 4]

Actual Output -
[0.3, 0.2, 0.5]

Output by Cloud API -
[0, 0, 1]

Assumption -
[0, 0, 1] ~ [0.3, 0.2, 0.5]
Inefficiencies with present attack methodology

<table>
<thead>
<tr>
<th>Input</th>
<th>Cloud API output</th>
<th>Actual Output</th>
<th>Unconventional probability loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>x11, x12, x13, x14</td>
<td>0, 0, 1</td>
<td>0.2, 0.3, 0.5</td>
<td>0.2+0.3</td>
</tr>
<tr>
<td>x21, x22, x23, x24</td>
<td>0, 1, 0</td>
<td>0.01, 0.9, 0.09</td>
<td>0.01+0.09</td>
</tr>
<tr>
<td>x31, x32, x33, x34</td>
<td>0, 0, 1</td>
<td>0.1, 0.4, 0.5</td>
<td>0.1+0.4</td>
</tr>
<tr>
<td>x41, x42, x43, x44</td>
<td>1, 0, 0</td>
<td>0.38, 0.32, 0.3</td>
<td>0.32+0.3</td>
</tr>
<tr>
<td>x51, x52, x53, x54</td>
<td>1, 0, 0</td>
<td>0.45, 0.3, 0.25</td>
<td>0.3+0.25</td>
</tr>
</tbody>
</table>
Scope for Attack optimization

1. Reconsider the way to analyze labels
   Having access to all the probability values will definitely help us to clone models in an efficient way

2. Learning parameters in hyperspace
   * To Duplicate the target model we need to learn the boundaries that the target model has learnt
   * Considering probability of predicted class as 1 and others to be 0 will cause unwanted loss and increase the gradient
   * Increased gradients cause the optimizer to change weights abruptly

\[ W_{i+1} = W_i + \alpha \frac{\partial}{\partial w} C \uparrow \]
Proposed approach (GDALR)

GDALR: Gradient Driven Adaptive Learning Rate

\[ W_{i+1} = W_i + \alpha \frac{\partial C}{\partial w} \]
Mathematical modification to current attack methodology

\[ g'_i = \tanh(g_i) \]  \hspace{1cm} (7)

\[ fact_i = \text{abs}(g'_i 2\pi \log_{10}(\text{abs}(g'_i))) \]  \hspace{1cm} (8)

\[ l'_i = l_i \cdot fact_i \]  \hspace{1cm} (9)
GDALR in Action

\[ g' = 0.33 \quad fact_i = 1.00 \]
Experimental setup

GDALR has been tested on multiple classifiers -

- LOGISTIC REGRESSION
- MULTI LAYER PERCEPTRON
- CONVOLUTIONAL NEURAL NETs
LOGISTIC REGRESSION

**Linear Regression**

\[ y_{\text{linear}} = wx + b \]

**Logistic Regression**

\[ y_{\text{logistic}} = \frac{1}{1 + e^{-(wx + b)}} \]

Ref: http://www.datacamp.com/
LOGISTIC REGRESSION

\[ l = 0.01 \quad l = 0.05 \]

\[ T_{Loss} = 0.0849 \quad T_{Loss} = 0.1233 \]

\[ P_{Loss} = 0.0317 \quad P_{Loss} = 0.0342 \]
MULTI LAYER PERCEPTRON

\[ l = 10^{-3} \quad \text{and} \quad l = 10^{-5} \]

<table>
<thead>
<tr>
<th></th>
<th>( T_{\text{Loss}} )</th>
<th>( P_{\text{Loss}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l = 10^{-3} )</td>
<td>0.0014</td>
<td>( 5.444 \times 10^{-5} )</td>
</tr>
<tr>
<td>( l = 10^{-5} )</td>
<td>0.0219</td>
<td>0.0007</td>
</tr>
</tbody>
</table>
### CNN

<table>
<thead>
<tr>
<th>$l = 10^{-4}$</th>
<th>$l = 10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph 1" /></td>
<td><img src="image2.png" alt="Graph 2" /></td>
</tr>
<tr>
<td>$T_{Loss} = 0.0011$</td>
<td>$T_{Loss} = 0.0073$</td>
</tr>
<tr>
<td>$P_{Loss} = 3.993 \times 10^{-5}$</td>
<td>$P_{Loss} = 4.184 \times 10^{-5}$</td>
</tr>
</tbody>
</table>
Thanks!

• Q & A
• Reach us at
  Email - [rewanth|nikhilj]@payatu.com
  Twitter - @Rewanth_Cool | @nikhilj_73