Adversarial Machine Learning And Several Countermeasures

Trend Micro
ch0upi
miaoski
7 Dec 2017
• Staff engineer in Trend Micro
• Machine Learning + Data Analysis
• Threat intelligence services
• NIPS
• KDDCup 2014 + KDDCup 2016: Top10
• GoTrend: 6th in UEC Cup 2015
• Senior threat researcher in Trend Micro
• Threat intelligence
• Smart City
• SDR
• Arduino + RPi makers
• 貓奴
Outline

• Cheating machine learning?
• Attacking theories and practices
• Countermeasures
• Conclusion
CHEAT MACHINE LEARNING MODELS
We Were Good Guys ...

Basis of machine learning in the evil software classification tool: Microsoft Malware Classification Challenge Experience

Trend Micro
ch0upi
mianzhi
Kyle Chung
2 Dec 2016

Evaluation of Fruit

- Accuracy: (9+9)/20 = 90%

<table>
<thead>
<tr>
<th></th>
<th>Apple</th>
<th>Banana</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Banana</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

MAKING THE UNKNOWNS KNOWN

With its cross-generational blend of threat defense techniques, including high-fidelity machine learning, Trend Micro™ X10 and endpoint security is always adapting to identify and defeat new ransomware and other unknown threats.
Malware Detection in Executables Using Neural Networks

The detection of malicious software (malware) is an increasingly important cyber security problem for all of society. Single incidences of malware can cause millions of dollars in damage. The current generation of anti-virus and malware detection products typically use a signature-based approach.
It's a DEBUG build too...

```c
#include <stdio.h>

int main()
{
    printf("Hello world!
");
    return 0;
}
```

**SHA256:** c99caff6b05d6d13629c7eb7d014862da7e2774866b61e7bfca47f53578dca0c

**File name:** helloworld.exe

**Detection ratio:** 7 / 64

**Analysis date:** 2017-08-10 14:07:06 UTC (0 minutes ago)

### Antivirus

<table>
<thead>
<tr>
<th>Antivirus</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>CrowdStrike Falcon (ML)</td>
<td>malicious_confidence_80% (D)</td>
</tr>
<tr>
<td>Cylance</td>
<td>Unsafe</td>
</tr>
<tr>
<td>Cyren</td>
<td>W32/S-d2b5872aEldorado</td>
</tr>
<tr>
<td>F-Prot</td>
<td>W32/S-d2b5872aEldorado</td>
</tr>
<tr>
<td>Sophos ML</td>
<td>heuristic</td>
</tr>
</tbody>
</table>
This is insane! You don't even need printf()! What's wrong with the world?
Here's why the scanners on VirusTotal flagged Hello World as harmful

CrowdStrike, Cylance, Endgame and others flagged Hello World as unsafe or malicious
### 18 engines detected this file

<table>
<thead>
<tr>
<th>Detection</th>
<th>Details</th>
<th>Community</th>
<th>Detection</th>
<th>Details</th>
<th>Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>Win32:Evo-gen [Susp]</td>
<td>Avira</td>
<td>CrowdStrike Falcon</td>
<td>malicious_confidence_80% (W)</td>
<td>Cylance</td>
</tr>
<tr>
<td>CrowdStrike Falcon</td>
<td>malicious_confidence_80% (W)</td>
<td>Cylance</td>
<td>Endgame</td>
<td>malicious (moderate confidence)</td>
<td>Jiangmin</td>
</tr>
<tr>
<td>Endgame</td>
<td>malicious (moderate confidence)</td>
<td>Jiangmin</td>
<td>McAfee</td>
<td>Artemis!6D130077084E</td>
<td>McAfee-GW-Edition</td>
</tr>
<tr>
<td>McAfee</td>
<td>Artemis!6D130077084E</td>
<td>McAfee-GW-Edition</td>
<td>nProtect</td>
<td>Trojan/W32.Agent.102082</td>
<td>Palo Alto Networks</td>
</tr>
<tr>
<td>nProtect</td>
<td>Trojan/W32.Agent.102082</td>
<td>Palo Alto Networks</td>
<td>Qihoo-360</td>
<td>Win32/Trojan.af4</td>
<td>SentinelOne</td>
</tr>
<tr>
<td>Qihoo-360</td>
<td>Win32/Trojan.af4</td>
<td>SentinelOne</td>
<td>Sophos ML</td>
<td>heuristic</td>
<td>Symantec</td>
</tr>
<tr>
<td>Sophos ML</td>
<td>heuristic</td>
<td>Symantec</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SHA-256: `aca55bce947a49f5073b9e860789f0f2b3cb147972e18178143ffaf6790160c4`
File name: `6d130077084e0b1f4542b08f92736df0.virobj`
File size: 99.69 KB
Last analysis: 2017-10-30 16:57:17 UTC
<table>
<thead>
<tr>
<th>Compiler</th>
<th>Hello World (no debug)</th>
<th>Hello World (debug)</th>
<th>Nothing (no debug)</th>
<th>Nothing (debug)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Studio 2017</td>
<td>Cylance, Jiangmin</td>
<td>Cylance, Cyren, F-Prot, Sophos ML, SentinelOne Static ML</td>
<td>Cylance, Jiangmin</td>
<td>Cylance, Cyren, F-Prot, Sophos ML, SentinelOne Static ML</td>
</tr>
<tr>
<td>MingW64</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Cygwin x86_64</td>
<td>Baidu, Cylance</td>
<td>Baidu</td>
<td>Baidu, Cylance</td>
<td>Baidu</td>
</tr>
</tbody>
</table>
ML is Prosperous

Taigman et al. (2014) DeepFace: Closing the Gap to Human-Level Performance in Face Verification
ML Drives

Protect against tomorrow’s threats

Machine learning has its particular vulnerabilities.
Research Prediction Competition

NIPS 2017: Targeted Adversarial Attack
Develop an adversarial attack that causes image classifiers to predict a specific target class

Google Brain · 65 teams · a month ago

Research Prediction Competition

NIPS 2017: Non-targeted Adversarial Attack
Imperceptibly transform images in ways that fool classification models

Google Brain · 91 teams · a month ago

Research Prediction Competition

NIPS 2017: Defense Against Adversarial Attack
Create an image classifier that is robust to adversarial attacks

Google Brain · 107 teams · a month ago
THEORIES AND PRACTICES
Methodology

- Evasion
- Black box
- White box
- Model stealing
- Poisoning
Methodology

- Evasion
  - Black box
    - Random
    - Evolutionary algorithms (GA)
  - White box
- Model stealing
- Poisoning
Black Box

- No model
- Only predict interface & result
Black Box: Random Noise Attack

• Add some white noise?

```
random.normalvariate(0, 5)
```

Not effective for most model
Black Box: Iterative Random Attack

1. Add some noise
2. Repeat hundreds times
3. Select the best one
• Inspired by Evtimov et al. (2017)
• We use iterative random attack instead
• Difficult: STOP sign ➔ something else
Black Box – Random – STOP

- Evtimov et al. (2017) → 80 KM/h

Hacked in iteration 5
Predicted Labels: 39 ['Keep left']
  (confidence = 73%)
39 - Keep left
14 - Stop
13 - Yield
  6 - End of speed limit (80km/h)
41 - End of no passing
• VGG Face and @mzaradzki

<table>
<thead>
<tr>
<th>N</th>
<th>Square Size</th>
<th>Success?</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4x4</td>
<td>Adam Driver</td>
</tr>
<tr>
<td>10</td>
<td>4x3</td>
<td>Adam Driver</td>
</tr>
<tr>
<td>10</td>
<td>3x3</td>
<td>Adam Driver</td>
</tr>
<tr>
<td>10</td>
<td>2x2</td>
<td>Adam Driver (difficult)</td>
</tr>
<tr>
<td>--</td>
<td>Cat face</td>
<td>Failed</td>
</tr>
<tr>
<td>10</td>
<td>1x1</td>
<td>Failed*</td>
</tr>
</tbody>
</table>
Black Box – Random – Faces

Adam Driver
Aamir Khan
Black Box – Genetic Algorithm

- Effective random search
- Inspired by the process of natural selection
- Belongs to evolutionary algorithms (EA)
- Solving optimization and search problems
Black Box – Genetic Algorithm

- Selection
- Crossover
- Mutation
- Evaluation

Figure 2
Methodology

- Evasion
- Black box
- **White box**
  - FGSM
  - One-step target class
- Model stealing
- Poisoning
With all model detail
DNN architecture, weights
Fast Gradient Sign Method

- simple and computationally efficient
- non-target attack
- Goodfellow et al. (2014)

\[ X^{adv} = X + \epsilon \text{sign}(\nabla_x J(X, y_{true})) \]

\( X^{adv} \): Adversarial image
\( X \): Original image
\( \epsilon \): perturbation level
\( \nabla_x J(X, y) \): gradient
Attack a Linear Model

Let's fool a binary linear classifier:

<table>
<thead>
<tr>
<th>x</th>
<th>2</th>
<th>-1</th>
<th>3</th>
<th>-2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>-4</th>
<th>5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>adversarial x</td>
<td>1.5</td>
<td>-1.5</td>
<td>3.5</td>
<td>-2.5</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
<td>-3.5</td>
<td>4.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

class 1 score before:

\[-2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3\]

\[\rightarrow \text{probability of class 1 is } 1/(1+e^{(-3)}) = 0.0474\]

\[-1.5+1.5+3.5+2.5+2.5-1.5+1.5-3.5-4.5+1.5 = 2\]

\[\rightarrow \text{probability of class 1 is now } 1/(1+e^{(-2)}) = 0.88\]

i.e. we improved the class 1 probability from 5% to 88%

Fei-Fei Li, Andrej Karpathy, Justin Johnson, Lecture 9-72, 2016
White Box Attack Methods

- Fast gradient sign method (non-target, one step)
  \[ X_{adv}^{*} = X + \epsilon \text{sign}(\nabla X J(X, y_{true})) \]

- One-step target class methods (target, one step)
  \[ X_{adv}^{*} = X - \epsilon \text{sign}(\nabla X J(X, y_{target})) \]

- Basic iterative method (non-target, multiple steps)
  \[ X_{0}^{adv} = X, \quad X_{N+1}^{adv} = \text{Clip}_{X} \{ X_{N}^{adv} + \alpha \text{sign}(\nabla X J(X_{N}^{adv}, y_{true})) \} \]

- Iterative least-likely class method (target, multiple steps)
  \[ X_{0}^{adv} = X, \quad X_{N+1}^{adv} = \text{Clip}_{X} \{ X_{N}^{adv} - \alpha \text{sign}(\nabla X J(X_{N}^{adv}, y_{LL})) \} \]

Kurakin et al., ADVERSARIAL MACHINE LEARNING AT SCALE. ICLR 2017
White Box – FGSM – Trash Can

label: 412 (ashcan, trash can), certainty: 37.47%
label: 899 (water jug), certainty: 10.85%
label: 503 (cocktail shaker), certainty: 7.98%

label: 412 (ashcan, trash can), certainty: 87.68%
label: 463 (bucket, pail), certainty: 3.08%
White Box – One-Step Target

- Giant panda (99.6%)
- Vulture (98.9%)

Vulture gradient
Methodology

- Evasion
- Black box
- White box
- Model stealing
- Poisoning
Model Stealing

\[ f(x) = \frac{1}{1 + e^{-(w^*x + b)}} \]

\[ \ln \left( \frac{f(x)}{1 - f(x)} \right) = w^*x + b \]

Model Stealing

- Model is data
- Model is asset

- Train a local DNN for Black box attack
- Data privacy
Model Stealing: Adversarial Attack

- Transferability Property
- Train a local model for attack
- Effective data augmentation

Ian Goodfellow, Practical Black-Box Attacks against Machine Learning, 2017
Model Stealing: Data Privacy

• How to re-build your face if we have the model?

Methodology

- Evasion
- Black box
- White box
- Model stealing
- Poisoning
Poison Attack

- Crowdsourcing
  - Amazon Mechanical Turk
- Mis-labeling
- Online training
  - Microsoft chatbot: Tay
- User feedback
Real World Adversarial

- Evading Against PDF ML
- Auto-pilot cars
- Access control w/ face recognition
Evading Against PDF ML

- Genetic algorithm to generate adversarial sample
- Sandbox to ensure malicious behavior kept

http://evademl.org/
Auto-pilot Cars
Access Control w/ Face Recognition

Protect against tomorrow's threats

Machine Learning
COUNTERMEASURES
Countermeasures

- Ensemble & Stacking
- Retrained model
- Denoiser
- Prevent Model Leakage
Ensemble & Stacking

- Layer protection

Input Data

Layer 1:
- CNN
- RNN
- LDA

Layer 2:
- SVM
- LR

Layer 3:
- Xgboost

Prediction
Retrained Models

- Distortion
- Retrain with noisy sample
- Randomization layer in DNN (NIPS 2\textsuperscript{nd})
- Generative Adversarial Networks (GAN)
Denoiser

- Use denoise technologies from image processing
- Train a DNN denoiser to reduce the noise
Prevent Model Leakage

• Avoid Model stealing
• Increase the challenge of black box attack

• Keep some info secret or add some noise
• Randomization and disinformation
• Adversarial sample detection
CONCLUSION
Conclusion

- Know the limitations and weakness of your model

- Integrate adversarial machine learning into product development cycle
  - Improve ML
  - QA process

- Trend Micro is working on bypassing anti-virus with ML in order to make our product robust
References

  - https://iotsecurity.eecs.umich.edu/#roadsigns
  - https://github.com/tomaszkacmajor/CarND-Traffic-Sign-Classifier-P2
  - https://github.com/davidsandberg/facenet
  - http://www.vlfeat.org/matconvnet/pretrained/#face-recognition
  - https://github.com/mzaradzki/neuralnets/tree/master/vgg_faces_keras
USE THE SOURCE, LUKE!