Protecting the Protector:
Hardening Machine Learning
Defenses Against Adversarial Attacks

Jugal Parikh, Senior Data Scientist
Holly Stewart, Principal Research Manager
Randy Treit, Senior Researcher
In a single day...

2.6 million people

232 country/regions

1.7 million first seen attacks
60% of these attacks were over within the hour.
Windows Defender Advanced Threat Protection

**Endpoint Protection**
- **Windows Defender SmartScreen**
  - Blocks malicious websites
- **Windows Defender Endpoint Protection**
  - Blocks low reputation web downloads
- **Windows Defender Endpoint Protection**
  - Blocks malicious programs and content (scripts, docs, etc.)
- **Windows Defender Endpoint Protection**
  - Monitors behaviors and terminates bad processes

**Detection and Remediation**
- **Windows Defender Endpoint Detection and Response**
  - After execution - Windows Defender ATP monitors for post-breach signals
- **Windows Defender Endpoint Detection and Response**
  - After execution - Hexadite can reverse damage

- Holly!
- Jugal!
- Randy!

---

**Introduction**
- ML Primer
- Adversarial Examples
- Ensemble Model Development and Testing
- Results
Windows Defender ATP Research

Our focus: Use machine learning to block attacks for Windows Defender ATP

Threat Predict Research Team
Machine Learning Primer
Types of Machine Learning

**SUPERVISIED**
- Automation & Detonation
- Expert Analysis

**UNSUPERVISIED**
- Unknown Unknowns
- Anomaly Detection

EXPERTS ➔ LABELS ➔ ML ➔ PREDICTIONS
Types of Machine Learning

SUPERVISED

EXPERTS ➔ LABELS ➔ ML ➔ PREDICTIONS
Machine Learning for Endpoint Protection

CLIENT

LOCAL ML MODELS, BEHAVIOR-BASED DETECTION, ALGORITHMS, GENERICS AND HEURISTICS

milliseconds

CLOUD

METADATA-BASED ML MODELS

milliseconds

SAMPLE ANALYSIS-BASED ML MODELS

seconds

DETONATION-BASED ML MODELS

minutes

BIG DATA ANALYSIS

hours
Client Machine Learning

Pro:Disconnected protection
Con: Silent adversarial brute force attacks
Cloud Machine Learning

No private brute forcing

Minimal client performance impact
Adversarial ML Examples
Theoretical Attack Vectors: Supervised Model

Dynamic and contextual attributes

Static file attributes

Example: Averaged perceptron model with > .90 probability = 0.0001% false positive rate
Specially crafted files ➔ AV industry FPs

1. Identify signature fragments detected as malicious
2. Identify automated detection techniques
3. Inject signature fragments into clean files
4. Add crafted files to VirusTotal using TOR
5. Automation signs targeted clean files, multiple vendors have FPs

For more details, see Immunity from antimalware automation attacks presented at Virus Bulletin 2013
Attacks on Certificate Reputation (Early 2017)

- Synthetic traffic designed to quickly gain reputation on a digital certificate
- Targeted Windows 8
- Originally surfaced as a high percentage of traffic that wasn’t classified
- Low-volume and unsigned file attacks were also identified during investigation

[Graph showing percent classified traffic over time for Win10, Win8, and Win7]
Attacks on Certificate Reputation (cont.)

- Attackers guessed major features (time, traffic, digital certificate)
- Team developed complementary models with additional features that filtered fake traffic out of telemetry
- Combination of models removed attack traffic from training data
Previous research pointed to ensembles

- Research on adversarial attacks against deep learning classifiers
- Showed that an ensemble of classifiers helped defend against the attacks tested in the paper
- See more at:

  **Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Classification Models**

  Jack W. Stokes, De Wang, Mady Marinescu, Marc Marino, Brian Bussone

Ensemble Machine Learning Primer
Overview

Challenges

Dealing with:

- Active adversarial
- Volatility/ Covariate Shift
- Noisy environment

Scale:

- Petabytes of threat Intelligence daily

Evaluate:

- ~2.3 Billion global queries everyday
Diversity
Diverse Models

1. Different feature sets
2. Different training algorithms
3. Different training data sets
4. Different optimization settings
Features - Highly dimensional data

Machine attributes
- OS version
- Processor
- Security settings

Behavioral and contextual attributes
- Process and installation
  - ProcessName
  - ParentProcess
  - TriggeringSignal
  - TriggeringFile
  - Download IP and URL
  - Parent/child relationships
- Behavioral
  - Connection IP
  - System changes
  - API calls
  - Process injection
- Locale
  - Locale setting
  - Geographical location

Researcher Expertise
- Partial and Fuzzy hashes
  - ClusterHash
  - ImportHash
  - Fuzzy hashes
- Full File Content
  - Header
  - Footer
  - Raw file content
- 10k+ researcher attributes
- 100k+ static attributes
- 10k+ behavioral attributes

Static file attributes
- File properties
  - File Geometry
  - File Size
  - FileMetaData
  - ....
- Signer info
  - Issuer
  - Publisher
  - Signer

Feature Set
- Training Algorithms
- Training Data Sets
- Optimization Settings
# Diverse Set of Classifiers

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Learner</th>
<th># of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE Properties</td>
<td>Fast Tree Ensemble</td>
<td>10K+ features</td>
</tr>
<tr>
<td>Researcher Expertise</td>
<td>Boosted Tree Ensemble</td>
<td>190K+ features</td>
</tr>
<tr>
<td>Behavioral</td>
<td>Boosted Tree Ensemble</td>
<td>6M+ features</td>
</tr>
<tr>
<td>Fuzzy Hash 1</td>
<td>Random Forest</td>
<td>512+ features</td>
</tr>
<tr>
<td>Fuzzy Hash 2</td>
<td>SDCA</td>
<td>10M+ features</td>
</tr>
<tr>
<td>Static, Dynamic and Contextual</td>
<td>Averaged Perceptron</td>
<td>16M+ features</td>
</tr>
<tr>
<td>Researcher Expertise, Fuzzy Hash</td>
<td>Averaged Perceptron</td>
<td>12M+ features</td>
</tr>
<tr>
<td>File Emulation</td>
<td>DNN</td>
<td>150K+ features</td>
</tr>
<tr>
<td>File Detonation</td>
<td>DNN</td>
<td>10M+ features</td>
</tr>
</tbody>
</table>
Optimizing for Different Threat Scenarios

Training Cadence: Classifiers:
- Malware
- Clean
- PUA
- Enterprise specific
- File Type specific

Feature Set | Training Algorithms | Training Data Sets | Optimization Settings
Developing the Stacked Model
Stacked Ensemble

1. Boolean Stacking
2. Linear/ Logistic Stacking
Boolean Stacking

Binary Output

Model Probabilities

Logistic Stacking
Model Selection

- LightGBM !!!
- Logistic Regression !!!
- Fast Tree !!!
- XGBoost !!!
- DNN !!!
- Averaged Perceptron !!!
Experiment Design
Supervised Training

Optimization

Evaluation

Train

Validate

Test
Final Training

• Generate diverse set of base classifiers

• Use model probabilities as input features to train the Stacked Classifier

• Use LightGBM to train the Stacked Classifier

• Plot the ROC curve for Stacked Classifier vs. Top Base Classifiers
Results

Stacked Ensemble Performance against top base classifiers
### Evaluating on Live Data?

#### Model evaluated on time-split test set

<table>
<thead>
<tr>
<th>Confusion table</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREDICTED</td>
</tr>
<tr>
<td>TRUTH</td>
</tr>
<tr>
<td>positive</td>
</tr>
<tr>
<td>negative</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>OVERALL ACCURACY: 0.9811</td>
</tr>
</tbody>
</table>

#### Model evaluated on Live Data for 60 mins without any calibrations

<table>
<thead>
<tr>
<th>Confusion table</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREDICTED</td>
</tr>
<tr>
<td>TRUTH</td>
</tr>
<tr>
<td>positive</td>
</tr>
<tr>
<td>negative</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>OVERALL ACCURACY: 0.9811</td>
</tr>
</tbody>
</table>
Testing the Model
Data Leaks

- Information from the target inadvertently works its way into the model-checking mechanism
- Causes an overly optimistic assessment of generalization performance
- Filtering features that directly correlate to the training labels

With some data leaks

<table>
<thead>
<tr>
<th>Confusion table</th>
<th>positive</th>
<th>negative</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREDICTED TRUTH</td>
<td>positive</td>
<td>703,140</td>
<td>59,131</td>
</tr>
<tr>
<td></td>
<td>negative</td>
<td>2,1030</td>
<td>8,013,623</td>
</tr>
<tr>
<td>Precision</td>
<td>0.9710</td>
<td>0.9927</td>
<td></td>
</tr>
<tr>
<td>OVERALL ACCURACY:</td>
<td>0.9811</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Filtering known data leaks

<table>
<thead>
<tr>
<th>Confusion table</th>
<th>positive</th>
<th>negative</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREDICTED TRUTH</td>
<td>positive</td>
<td>625,324</td>
<td>136,947</td>
</tr>
<tr>
<td></td>
<td>negative</td>
<td>29,540</td>
<td>8,005,113</td>
</tr>
<tr>
<td>Precision</td>
<td>0.9549</td>
<td>0.9832</td>
<td></td>
</tr>
<tr>
<td>OVERALL ACCURACY:</td>
<td>0.9668</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10% drop in Recall
Handling Missing Values

- Not all Base Classifiers classify every threat scenario

- What you can do:
  - Retaining the instance but..
  - Adding Boolean features indicating what features were missing
  - Cross Join between features
  - Interpretable models

<table>
<thead>
<tr>
<th>Model</th>
<th>Probability</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileEmulation1</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>FileDetonation</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>FuzzyHash1</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>FuzzyHash2</td>
<td>0.014020299</td>
<td>Clean</td>
</tr>
<tr>
<td>CloudClassifier1</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>CloudClassifier2</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>CloudClassifier3</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>CloudClassifier4</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>CloudClassifier5</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>CloudClassifier6</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>ResearcherExpertise</td>
<td>0.07285905</td>
<td>Clean</td>
</tr>
<tr>
<td>PEPropertiesClassifier</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>FileMetaDataClassifier1</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>FileMetaDataClassifier2</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>FileMetaDataClassifier3</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>BehavioralClassifier1</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>BehavioralClassifier2</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td>BehavioralClassifier3</td>
<td>N/A</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Stacked Ensemble</strong></td>
<td><strong>0.92</strong></td>
<td><strong>Malware</strong></td>
</tr>
</tbody>
</table>
Using Unsupervised Features

- Adding K-means distance for each instance from the centroid of each cluster as an input feature
Other Improvements

- Maintaining a fixed label distribution for training
- Continuous monitoring of incoming telemetry to catch anomalies/outliers before training
Model Deployed !!!

But is it Resilient to Adversarial Attacks...
What if ... We evaluate on rogue/ noisy classifiers as features
Experimental Verification

*Trust, but Verify!!!*
What if ... We train on new rogue/ noisy classifiers as features

- **Dynamic and contextual attributes**
- **Static file attributes**

Example: Averaged perceptron model with > .90 probability = 0.0001% false positive rate
Experimental Verification

Supervised Training

Adding Rogue Features

Random Noise
## Experimental Verification

<table>
<thead>
<tr>
<th># of Random Classifiers</th>
<th>False Positive Rate</th>
<th>True Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.8746%</td>
<td>96.1824%</td>
</tr>
<tr>
<td>2</td>
<td>0.8834%</td>
<td>96.1222%</td>
</tr>
<tr>
<td>4</td>
<td>0.8912%</td>
<td>96.0385%</td>
</tr>
<tr>
<td>6</td>
<td>0.8939%</td>
<td>95.8932%</td>
</tr>
<tr>
<td>8</td>
<td>0.8974%</td>
<td>95.8462%</td>
</tr>
<tr>
<td>10</td>
<td>0.9131%</td>
<td>95.8462%</td>
</tr>
</tbody>
</table>

The classifier can detect these random noises and the performance drop is negligible.
What if ... Attacker crafts adversarial samples to flip verdicts

<table>
<thead>
<tr>
<th>SAMPLES</th>
<th>FEATURES</th>
<th>MODELS/PARAMETERS</th>
<th>CLIENT/CLOUD</th>
</tr>
</thead>
</table>

Example: Averaged perceptron model with > 0.90 probability = 0.0001% false positive rate
Experimental Verification

Flipping top 2 Feature Values

Ratio of number of instances perturbed

False Positive Rate

True Positive Rate

0.0 (area = 1.00)
0.1 (area = 1.00)
0.2 (area = 0.99)
0.3 (area = 0.99)
0.4 (area = 0.99)
0.5 (area = 0.99)
0.6 (area = 0.99)
0.7 (area = 0.99)
0.8 (area = 0.99)
0.9 (area = 0.98)
1.0 (area = 0.98)
What if...

Attacker is highly motivated to somehow just break our Stacked Ensemble
Realtime Monitoring

SAMPLES

- # of Instances
- Data Distribution
- Bias
- Anomalies

FEATURES

- Relevant features
- Threat Landscape

MODELS/PARAMETERS

- Metrics Requirements

CLIENT/CLOUD

- Continuous Monitoring
- Overall Blocks
- Telemetry

Ensemble Model Development and Testing

- Ensemble ML Primer
- Diversity Requirements
- Developing the Model
- Testing the Model
Results!
**Impact of Ensemble Models**

**Percent of Threats Blocked by Cloud Protection**
June 2018

- Ensemble Blocks: 12%
- Other Cloud Blocks: 88%

**Ensemble Model Blocks by File Type**
June 2018

- PE File: 68%
- Other Cloud Blocks: 88%
- Shortcut: 11%
- Signed PE File: 3%
- VBS: 3%
- Archive: 4%
- Documents and macros: 8%
- Other (1k+ types):
Bonus: Interpretability

Top classifiers contributing to malware verdict

Top classifiers contributing to clean verdict
Benefits of an Ensemble Model

- Filters out noisy signals from an occasionally underperforming model
- Increases predictive power with easy interpretability
- Adds resilience against attacks on individual models
Recent Realworld Case Studies (2)

CLIENT

LOCAL ML MODELS, BEHAVIOR-BASED DETECTION ALGORITHMS, GENERICS AND HEURISTICS
milliseconds

CLOUD

METADATA-BASED ML MODELS
milliseconds

SAMPLE ANALYSIS-BASED ML MODELS
seconds

DETONATION-BASED ML MODELS
minutes

BIG DATA ANALYSIS
hours
Case Study 1: Spear Phishing

- Small-scale attack in Central and Western Canada
- Most targets reached within 5 ½ hours
- 73% of targets were commercial businesses
The Attack – Landscaping Invoice

[Image of an email message]

Source: Email with malicious attachment later found posted to VirusTotal

https://www.malware365docs.com/552423
Ensemble Model Results

Client ML Verdict: Suspicious

Static and dynamic feature extraction

Cloud ML Verdict: Malware

<table>
<thead>
<tr>
<th>Model</th>
<th>Probability</th>
<th>Verdict</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDFClassifier</td>
<td>-</td>
<td>Unknown</td>
</tr>
<tr>
<td>NonPEClassifier</td>
<td>-</td>
<td>Unknown</td>
</tr>
<tr>
<td>CloudModel3</td>
<td>0.06</td>
<td>Clean?</td>
</tr>
<tr>
<td>FuzzyHash1</td>
<td>0.07</td>
<td>Clean?</td>
</tr>
<tr>
<td>FuzzyHash2</td>
<td>0.08</td>
<td>Clean?</td>
</tr>
<tr>
<td>ResearchExpertise</td>
<td>0.99</td>
<td>Malware</td>
</tr>
<tr>
<td>Ensemble1</td>
<td>0.27</td>
<td>~Sketchy</td>
</tr>
<tr>
<td>Ensemble2</td>
<td>0.84</td>
<td>Sketchy!</td>
</tr>
<tr>
<td>Ensemble3</td>
<td>0.91</td>
<td>Malware</td>
</tr>
</tbody>
</table>

Client - fundamental_statement.pdf
Case Study 2: JavaScript Banking Trojan (Bancos)

Around 3k targets in Brazil, lasted a few days
Obfuscated, polymorphic js payload

Documento importante!.msg
Doc061208.zip
Doc061208-2.vbs
Ensemble Model Results

Client

vNSAml.js

Client ML Verdict: Suspicious

Static and dynamic feature extraction

Cloud

Model | Probability | Verdict
--- | --- | ---
FuzzyHash2 | 0.06 | Clean?
Ensemble1 | 0.27 | ~Sketchy
JsModel | 0.52 | Sketchy!
ResearchExpertise | 0.64 | Sketchy!
Ensemble3 | 0.91 | Malware

Cloud ML Verdict: Malware

vNSAml.js
Last Words
Thanks to our contributors

• Daewoo Chong (Windows Defender ATP Research)
• Christian Seifert (Windows Defender ATP Research)
• Allan Sepillo (Windows Defender ATP Research)
• Bhavna Soman (Windows Defender ATP Research)
• Jay Stokes (Microsoft Research)
• Maria Vlachopoulou (Windows Defender ATP Research)
• Samuel Wakasugui (Windows Defender ATP Research)
Today’s presentation

All data and charts, unless otherwise noted, is from Microsoft.


Blog: https://aka.ms/hardening-ML

Upcoming conference presentations

Virus Bulletin 2018 (Montreal): Starving malware authors through dynamic classification
Karishma Sanghvi (Microsoft), Joe Blackbird (Microsoft)

Blog Posts and Other References

Antivirus evolved
Windows Defender Antivirus cloud protection service: Advanced real-time defense against never-before-seen malware
Detonating a bad rabbit: Windows Defender Antivirus and layered machine learning defenses
How artificial intelligence stopped an Emotet outbreak
Behavior monitoring combined with machine learning spoils a massive Dofoil coin mining campaign
Machine Learning vs. Social Engineering
Whitepaper: The Evolution of Malware Prevention
Client-based machine learning is susceptible to brute force attacks.

Build a diverse set of complementary models, then add an ensemble layer.

Consider the various vectors of attack, identify most likely vectors, then test them.

After you deploy, ensure you have monitors to alert on potential tampering.
Thank you!

adversarialml@microsoft.com

PS We’re hiring Data Scientists, Researchers, Hunters, Security Engineers – come talk to us!