Classifiers Under Attack

David Evans

University of Virginia

evans@virginia.edu
evadeML.org

work with Weilin Xu and Yanjun Qi
Machine Learning is Solving All Our Problems!

- Spam
- IDS
- Malware
- Fake Accounts
- ...
- ...

Fake Spam IDS Malware

Machine Learning is Solving All Our Problems!
Machine Learning is Eating the World
Assumption: Training Data is Representative
Reality: **Adversary Adapts**

**Actual images**

**Recognized faces**
Case study:
Evading PDF Malware Classifiers
Malware Classifiers in Practice

**Goal:** Automatically simulate adaptive adversary against generic classifier

**Purpose:** Understand classifier robustness
Build better classifiers (??)
Vulnerabilities reported in Adobe Acrobat Reader

High-Value Exploits: MiniDuke

The Informal Asia-Europe Meeting (ASEM) Seminar on Human Rights

ASEM, the Asia-Europe Meeting, is a forum that promotes various levels of cooperation among Asian and European countries. It represents a process based on dialogue with the objective of strengthening interaction and mutual understanding between the two regions and promoting cooperation that aims at sustainable economic and social development.

ASEM is an informal process of dialogue and cooperation among partners on all issues of common interest to Asia and Europe. Summit meetings are held every other year in Asia and Europe alternatively. This is the highest level of decision making in the process, featuring the Heads of States or Governments, the President of the European Commission, accompanying ministers and other stakeholders. So far, eight Summit meetings have been held: in Bangkok (1996); London (1998); Seoul (2000); Copenhagen (2002); Hanoi (2004); Helsinki (2006); Beijing (2008) and Brussels (2010). The next Summit meeting will be held in Vientiane, in 2012.

On the occasion of the first meeting of ASEM Foreign Ministers in Singapore in February 1997, Sweden and France had suggested that informal seminars on human rights be held within the ASEM framework. The aim of this initiative was to promote mutual understanding and co-operation between Europe and Asia in the area of political dialogue, particularly on human rights issues.

Previous seminar topics include:

PDF Malware

Exploits CVE-2007-5659 buffer overflow
PDF Malware Classifiers

PDFrate  [ACSA 2012]  Random Forest

Hidost$^{13}$  [NDSS 2013]  Support Vector Machine

Hidost$^{16}$  [JIS 2016]  Random Forest
Random Forest

Generate many random decision trees

Train independently

Select best trees

Vote on result
PDF Malware Classifiers

**PDFrate**
[ACSA 2012]
Random Forest
Object counts, lengths, positions, etc.

**Hidost**

- **Hidost**
  [NDSS 2013]
  Support Vector Machine
  Object structural paths
  *Very robust against “strongest conceivable mimicry attack”.*

- **Hidost**
  [JIS 2016]
  Random Forest
## Classifier Performance

<table>
<thead>
<tr>
<th></th>
<th>PDFrate*</th>
<th>Hidost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.9976</td>
<td>0.9996</td>
</tr>
<tr>
<td>False Negative Rate</td>
<td>0.0000</td>
<td>0.0056</td>
</tr>
</tbody>
</table>

* Mimicus [Oakland ’14], an open source reimplementation of PDFrate.
## Classifier Performance

<table>
<thead>
<tr>
<th></th>
<th>PDFrate*</th>
<th>Hidost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>0.9976</td>
<td>0.9996</td>
</tr>
<tr>
<td><strong>False Negative Rate</strong></td>
<td>0.0000</td>
<td>0.0056</td>
</tr>
<tr>
<td><strong>Adversarial False Negative Rate</strong></td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
</tr>
</tbody>
</table>

* Mimicus [Oakland ’14], an open source reimplementation of PDFrate.
Automatically Evading Classifiers
Automated Classifier Evasion Using Genetic Programming
Goal: Find Evasive Variant

Evasive variant:

Simulated attacker’s goal: find a variant that is classified as benign, but exhibits the same malicious behavior.
PDF Structure

1 0 obj <<
/Type /Catalog
/Pages 2 0 R
/OpenAction <<
/S /JavaScript
/JS alert('hello');
>>
>> endobj

2 0 obj <<
/Type /Pages
/Kids [3 0 R]
/Count 1
>>
>> endobj

3 0 obj <<
/Type /Page
/Parent 2 0 R
/MediaBox [0 0 128 546]
/Resources ...
>>
>> endobj

/JavaScript
alert('hello');
Malicious Seed

Parser is “robust” version of pdfrw:
- Handles ungrammatical PDFs
- Ignores inconsistencies, etc.
Malware often malformed
Generating Variants

Malicious PDF → Clone → Variants → Mutation → Variants → Benign PDFs

Select Variants
Generating Variants

Malicious PDF → Clone → Variants

Benign PDFs → Mutation → Variants

Select random node

Variants

/Catalog

/Pages

/Root

0

/JavaScript
eval(‘…’);
Generating Variants

Malicious PDF → Clone → Variants → Benign PDFs

Variants

Mutation

Select random node
Random transform: delete, insert, replace
Generating Variants

Malicious PDF → Clone → Benign PDFs → Variants → Mutation → Variants

Nodes from Benign PDFs:
- 7
- 63
- 128
- 546

Select random node
Random transform: delete, insert, replace

Variants for Generating Variants:
- Clone
- Mutation
- Variants

Benign PDFs

Random transform: delete, insert, replace

Variants:
Selecting Promising Variants

Malicious PDF → Clone → Variants → Mutation → Benign PDFs → Generated Variants → Select Variants → Next Generation
Selecting Promising Variants

Candidate Variant

Target Classifier

Oracle

Fitness Function

\[ f(s_{\text{oracle}}, s_{\text{class}}) \]

Score

Select Variants
Oracle

Execute candidate in vulnerable Adobe Reader in virtual environment

Simulated network: INetSim

Behavioral signature: only considered malicious if signature matches

http://github.com/cuckoosandbox

HTTP_URL + HOST extracted from API traces
Fitness Function

Assumes lost malicious behavior will not be recovered

\[ f(v) = \begin{cases} 
0.5 - \text{classifier\_score}(v) & \text{if } \text{oracle}(v) = \text{"malicious"} \\
\infty & \text{otherwise}
\end{cases} \]

\text{classifier\_score} \geq 0.5: \text{labeled} \text{malicious}
Experimental Results
Original Malicious Seeds

Evading PDFrate

Classification Score

Malicious Label Threshold

Malware Seed ID
Original Malicious Seeds

Evading PDFrate

100% success rate

~130 hours on typical desktop
Evading Hidost 2013

100% success rate

~46 hours on typical desktop
Evading Hidost 2016
Evading Hidost 2016

100% success rate

~14 hours on typical desktop
Simple transformations often worked.

Hidost

PDFRate
Inserting new pages works on 162/500 seeds
Training malware often had no/little content
Deleting object worked on 1 seed
No impact on malicious behavior

Hidost

(delete, /Root/OpenAction/JS/Length)
Some seeds required complex transformations.
Complex Transformations

Insert: Threads, ViewerPreferences/Direction, Metadata, Metadata/Length, Metadata/Subtype, Metadata/Type, OpenAction/Contents, OpenAction/Contents/Filter, OpenAction/Contents/Length, Pages/MediaBox

Delete: AcroForm, Names/JavaScript/Names/S, AcroForm/DR/Encoding/PDFDocEncoding, AcroForm/DR/Encoding/PDFDocEncoding/Differences, AcroForm/DR/Encoding/PDFDocEncoding/Type, Pages/Rotate, AcroForm/Fields, AcroForm/DA, Outlines/Type, Outlines, Outlines/Count, Pages/Resources/ProcSet, Pages/Resources

85-step mutation trace evading Hidost
Effective for 198/500 seeds
Practical, Inexpensive

Less than 1 week to find evasive variants for all 500 seeds, running on single desktop PC
Possible Defenses
Adjust threshold?

Hidost16 results
Adjust threshold?

Variants found with threshold = 0.25

Variants found with threshold = 0.50

Hidost16 results
Adjust threshold?

Variants found with threshold = 0.25

Variants found with threshold = 0.50

PDFRate results
Retraining Classifier

Training (Supervised Learning)

Labelled Training Data

Vector Extraction
ML Algorithm

Deployment
Operational Data
Trained Classifier
Malicious / Benign
Training (Supervised Learning)

Labelled Training Data → Feature Extraction → Vectors → ML Algorithm → EvadeML

Evasive
Training (Supervised Learning)

Labelled Training Data

Feature Extraction

ML Algorithm

Vectors

Deployment

EvadeML
### (Probably) Doesn’t Work

<table>
<thead>
<tr>
<th></th>
<th>Original (Hidost 2016)</th>
<th>Retrained (without new benign)</th>
<th>Retrained (with new benign)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy on Test Set</strong></td>
<td>0.9983</td>
<td>0.9983</td>
<td>0.9983</td>
</tr>
<tr>
<td><strong>False negatives</strong></td>
<td>12</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>on 250 non-training seeds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>False positive rate</strong></td>
<td>0.0%</td>
<td>77%</td>
<td>0.0%</td>
</tr>
<tr>
<td>(on benign samples)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Evasion rate</strong></td>
<td>100%</td>
<td>49%</td>
<td>100%</td>
</tr>
</tbody>
</table>

*more experiments in progress...*
Hide the Classifier?

“Security through Obscurity”

Candidate Variant

Target Classifier

Score

Fitness Function

\[ f(S_{oracle}, S_{class}) \]

Oracle

Select Variants

Benign PDFs

Malicious PDF Variants

Benign PDFs

Malicious PDF Variants

JavaScript

eval('...');
Cross-Evasion Effects

PDF Malware Seeds -> Automated Evasion -> Evasive PDF Malware (against Hidost) -> Hidost 13

2/500 Evasive (0.4% Success)

Potentially Good News?
Cross-Evasion Effects

PDF Malware Seeds → Automated Evasion → Evasive PDF Malware (against Hidost) → PDFrate

387/500 Evasive (77.4% Success)
Cross-Evasion Effects

PDF Malware Seeds → Automated Evasion → Evasive PDF Malware (against Hidost) → Gmail

6/500 Evasive (0.6% Success)
Evading Gmail’s Classifier

for javascript in pdf.all_js:
    javascript.append_code("var oreilly=1;\"")

if pdf.get_size() < 7050000:
    pdf.add_padding(7050000 - pdf.get_size())

Evasion rate on Gmail: 179/380 (47.1%)
Fundamental Problem

Classifier features are not intrinsic to malicious behavior

Artifacts of training data  Adversary can modify those features

Heuristic search can find evasive variants automatically
Conclusion

For source code, technical paper: EvadeML.org

Adversaries adapt, classifiers cannot rely on superficial features

If you are developing or using malware classifiers, we want to work with you to test them for evadability: evans@virginia.edu
David Evans
evans@virginia.edu
EvadeML.org