The Laboratory for Physical Sciences

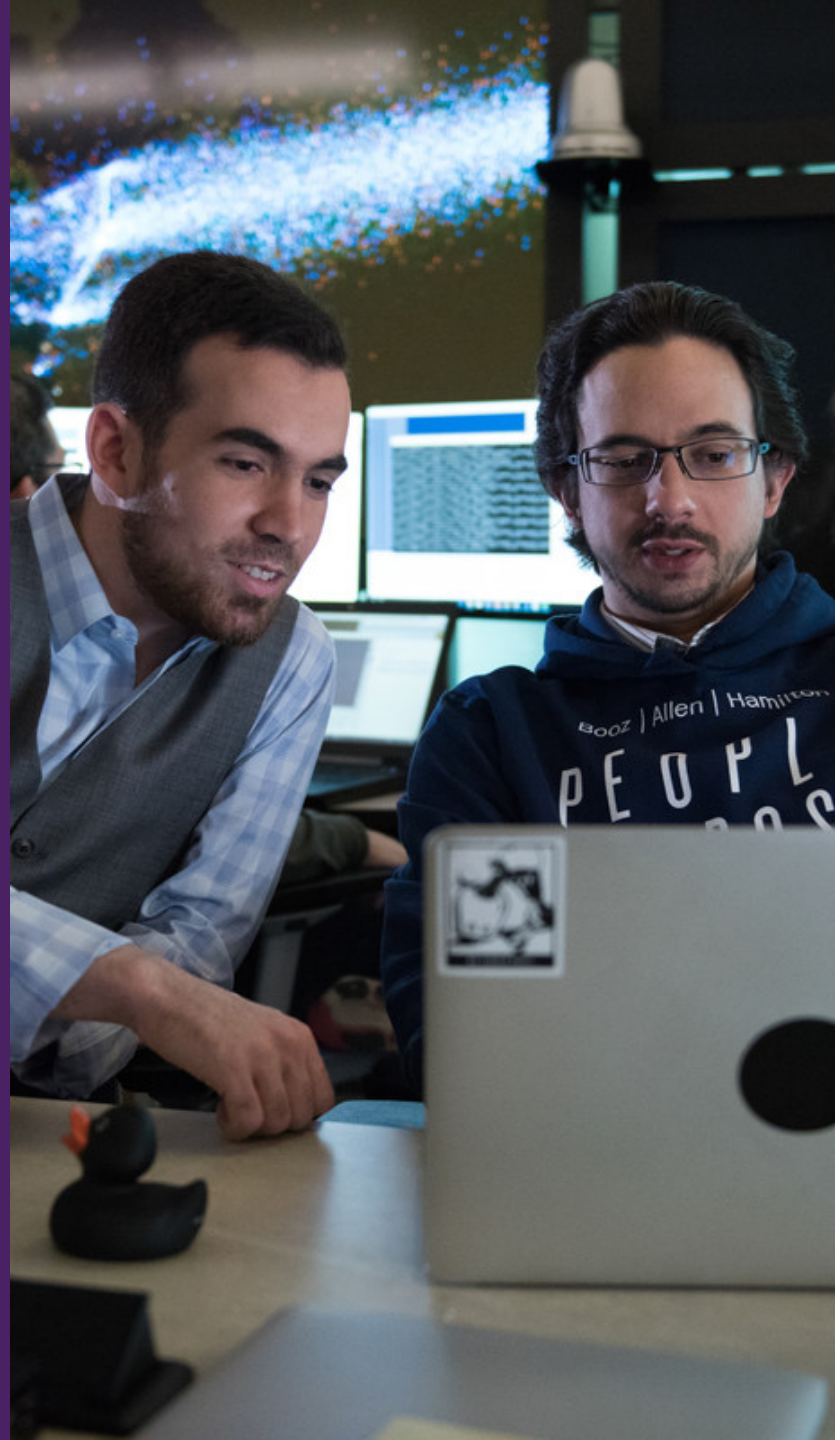
Booz | Allen | Hamilton

RESISTING ADVERSARIAL ATTACKS ON MACHINE LEARNING MALWARE DETECTORS

Capt. William Fleshman, US Army &

Dr. Jared Sylvester, Booz Allen Hamilton

GTC-DC | 22–24 OCTOBER 2018



CONTENTS

TYPES OF ATTACKS

FOR MALWARE DETECTION THE ADVERSARY'S GOAL IS TO EVADE THE DETECTOR WITH A MALICIOUS FILE.

- Black box attacks
  - Traditionally used by attackers for evading Anti-Virus
  - Doesn't require reverse engineering software or models
  - Shown to be effective in practice
- White box attacks
  - Access to the inner workings of the model

ANTI-VIRUS PRODUCTS

- We selected four products to compare against based on the following criteria:
  - Representative of what is available
  - The ability to isolate static analysis component
  - Not advertised as a “Machine Learning” AV company
  - Verbose enough to capture detailed statistics at scale

(To comply with EULAs and minimize legal concerns the products will remain anonymized.)

MACHINE LEARNING MODELS

WE USE TWO MACHINE LEARNING MODELS

- MalConv
  - A convolutional neural network whose input is raw bytes
  - Raw bytes → embedding → gated convolution → pooling → classification
  - (See "Malware Detection by Eating a Whole EXE" on arXiv or GTC-DC 2017)
- N-Gram
  - Logistic regression on 6-byte n-grams
  - (See "An investigation of byte n-gram features for malware classification" in J. Comp. Virology)
- Training data
  - 2 million binaries
  - Windows portable executables (EXEs)

BASELINE PERFORMANCE

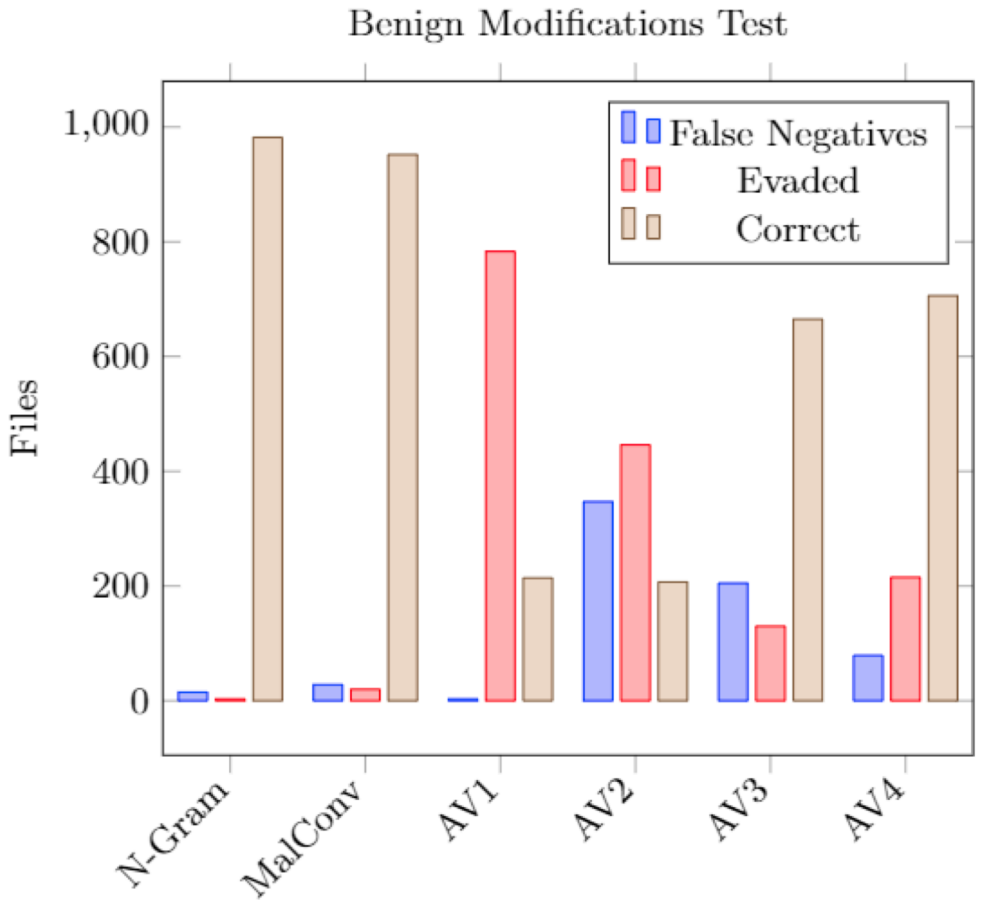
- 80,000 file testing set
- ML models had not seen these files
- AV may have, so difficult to compare raw metrics between the two.

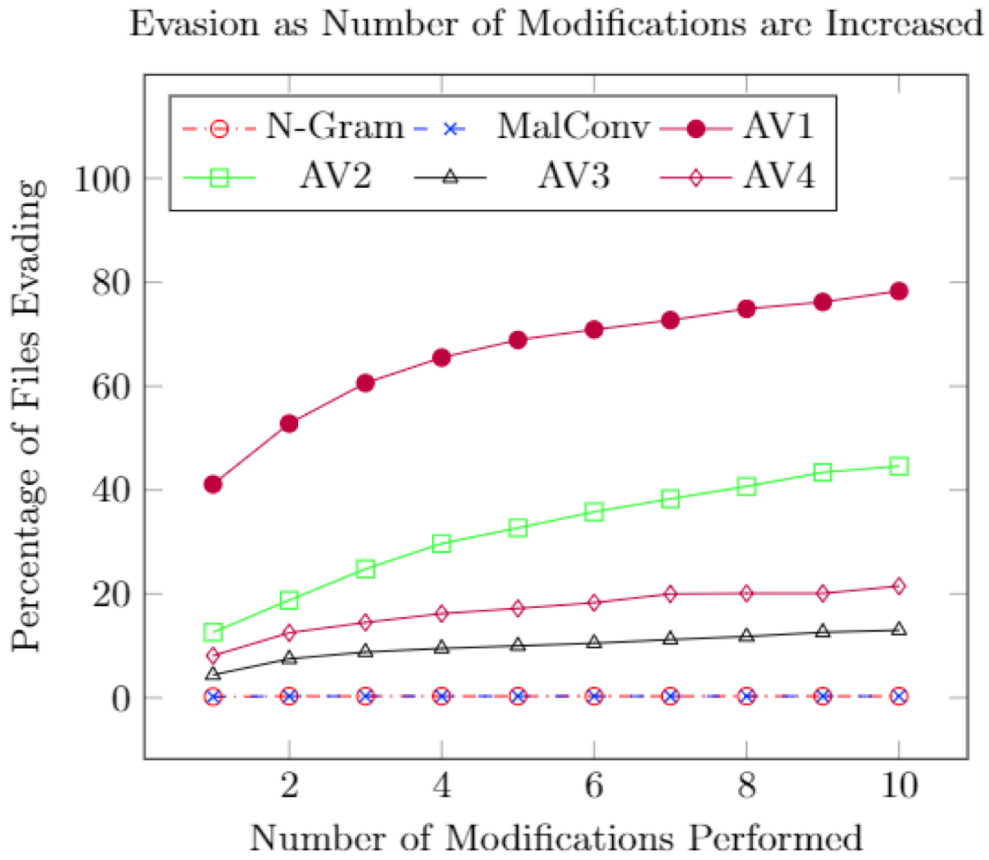
Classifier	TN%	TP%	FN%	FP%	Accuracy%
N-Gram	92.1	98.7	1.3	7.9	95.5
MalConv	90.7	97.2	2.8	9.3	94.1
AV1	94.3	99.5	0.5	5.7	97.0
AV2	99.4	64.9	35.1	0.6	81.6
AV3	98.5	80.5	19.5	1.5	89.2
AV4	93.8	91.9	8.1	6.6	92.6

ATTACK 1: NON DESTRUCTIVE MODIFICATIONS

- Our first attack is a modified version of the EndGame “EvadeRL” framework.
- A set of benign modifications are made to the malware files without changing any functionality.
- Possible modifications are:
  - Rename sections or create new sections
  - Append bytes to the end of a section of the file
  - Add an unused function to the import table
  - Create a new entry point (which jumps to the old entry)
  - Modify the header checksum, the signature or debug info

ATTACK 1: RESULTS





ATTACK 2: DESTRUCTIVE MODIFICATIONS

- Systematically occlude sections of a binary
- Monitor changes in “maliciousness” score to find most important 2kb chunk
- Types of occlusion:
  - Random bytes
  - Bytes from benign training set
- Undirected
  - Randomly select which bytes to occlude

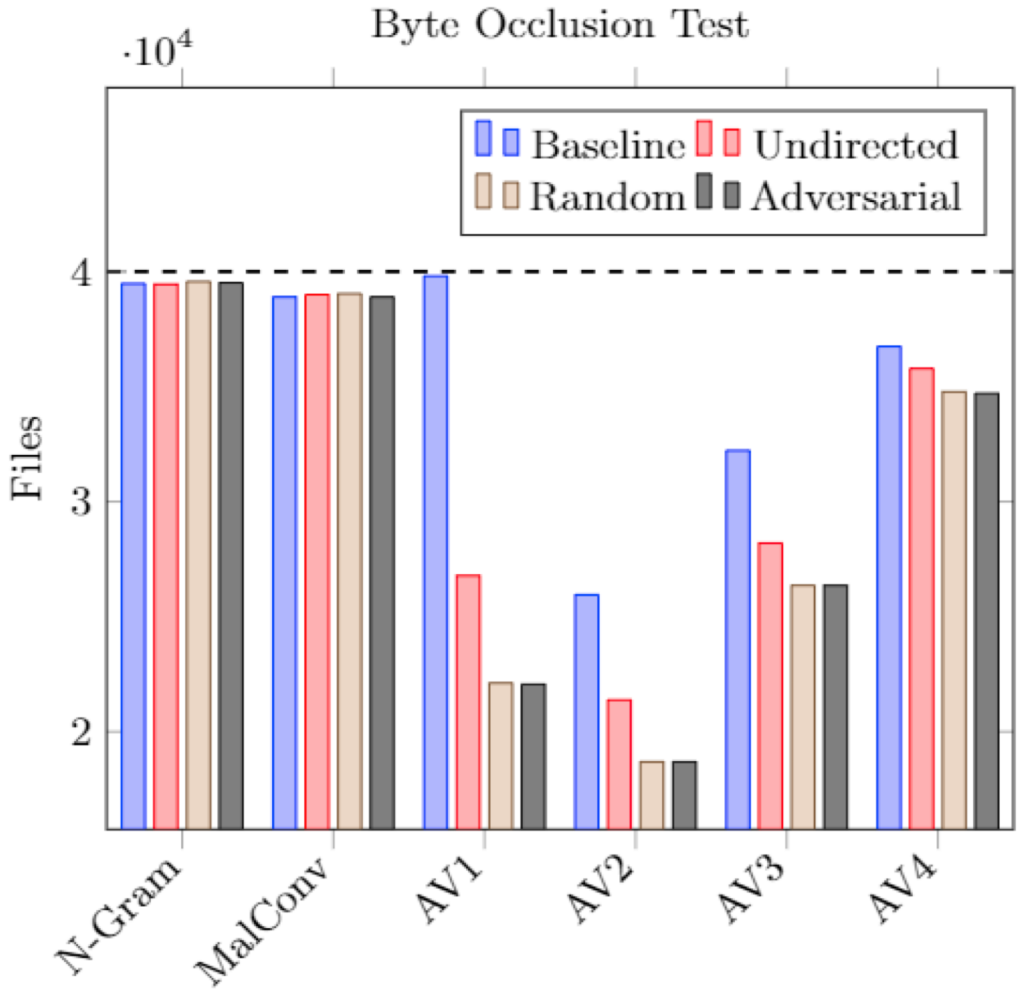
Algorithm 1 Occlusion Binary Search

Require: A file F of length $|F|$,
a classifier $C(\cdot)$,
target occlusion size β,
byte replacement distribution $\mathcal{D}$

split $\leftarrow |F|/2$, size $\leftarrow |F|/2$
start $\leftarrow 0$, end $\leftarrow |F|$
while size $> \beta$ **do**
  $F_l \leftarrow F$, $F_r \leftarrow F$
  F_l[split−size:split] $\leftarrow$ contiguous sample from $\sim \mathcal{D}$
  F_r[split:split+size] $\leftarrow$ contiguous sample from $\sim \mathcal{D}$
  if $C(F_l) < C(F_r)$ **then**
    split $\leftarrow$ split $-$ size/2
    start $\leftarrow$ split $-$ size
    end $\leftarrow$ split
  else
    split $\leftarrow$ split $+$ size/2
    start $\leftarrow$ split
    end $\leftarrow$ split $+$ size
  size $\leftarrow$ size/2
return start, end

ATTACK 2: RESULTS

- ML Models mostly unaffected
  - Show slight variations for targets that were close to decision boundary
- AV programs highly affected by occlusions "targeted" by n-gram model
  - Looking for small signatures leads to brittle decisions
- Targeted occlusions were more challenging than random occlusions



ATTACK 3: ROP INJECTOR

- Inject reverse backdoor into benign files using existing ROP instructions
- Hard to detect with static analysis
  - Instructions being used are already in the binary
- Can it be learned?

Table 1: Accuracy on originally benign binaries before and after applying the ROPInjector.

Classifier	Pre-ROP Accuracy	Post-ROP Accuracy	Post-ROP Lift
N-Gram	85.1	15.3	0.4
MalConv	82.4	18.8	1.2
AV1	99.3	1.3	0.6
AV2	98.7	1.2	−0.1
AV3	97.9	0.7	−1.4
AV4	89.2	32.9	22.1

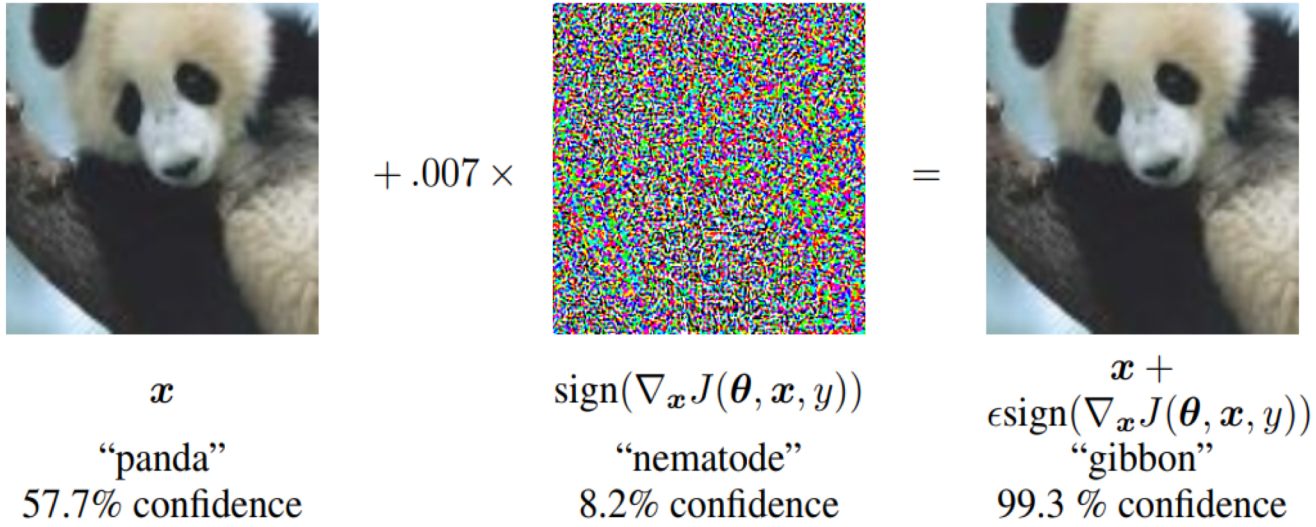
ATTACK 4: PACKING

- Packing a binary degrades the performance of all detectors…
  - Except AV4, which likely includes an unpacking routine
- ML classifiers are less effective at classifying packed benign files
- AV products are less effective at classifying packed malicious files

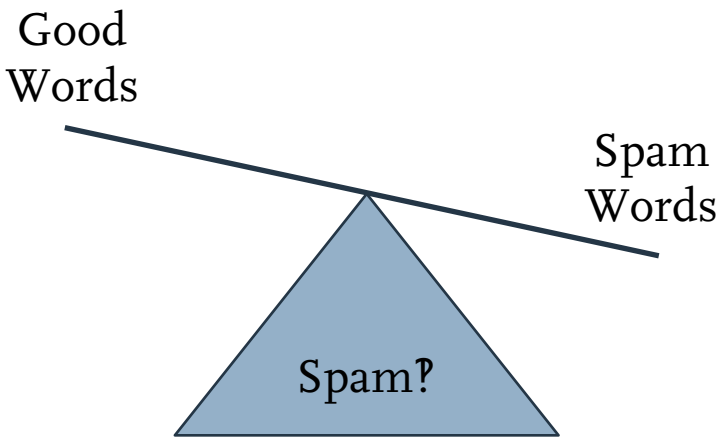
Classifier	Benign	Packed Benign	Malware	Packed Malware
N-Gram	92.1	74.8	98.7	95.0
MalConv	90.7	68.1	97.2	92.2
AV1	94.3	97.0	99.5	60.8
AV2	99.4	99.3	64.9	56.5
AV3	98.5	99.1	80.5	57.6
AV4	93.8	93.4	91.9	95.2
	True Positive Rates		*True Negative Rates*	

BUT WHAT ABOUT ADVERSARIAL EXAMPLES?

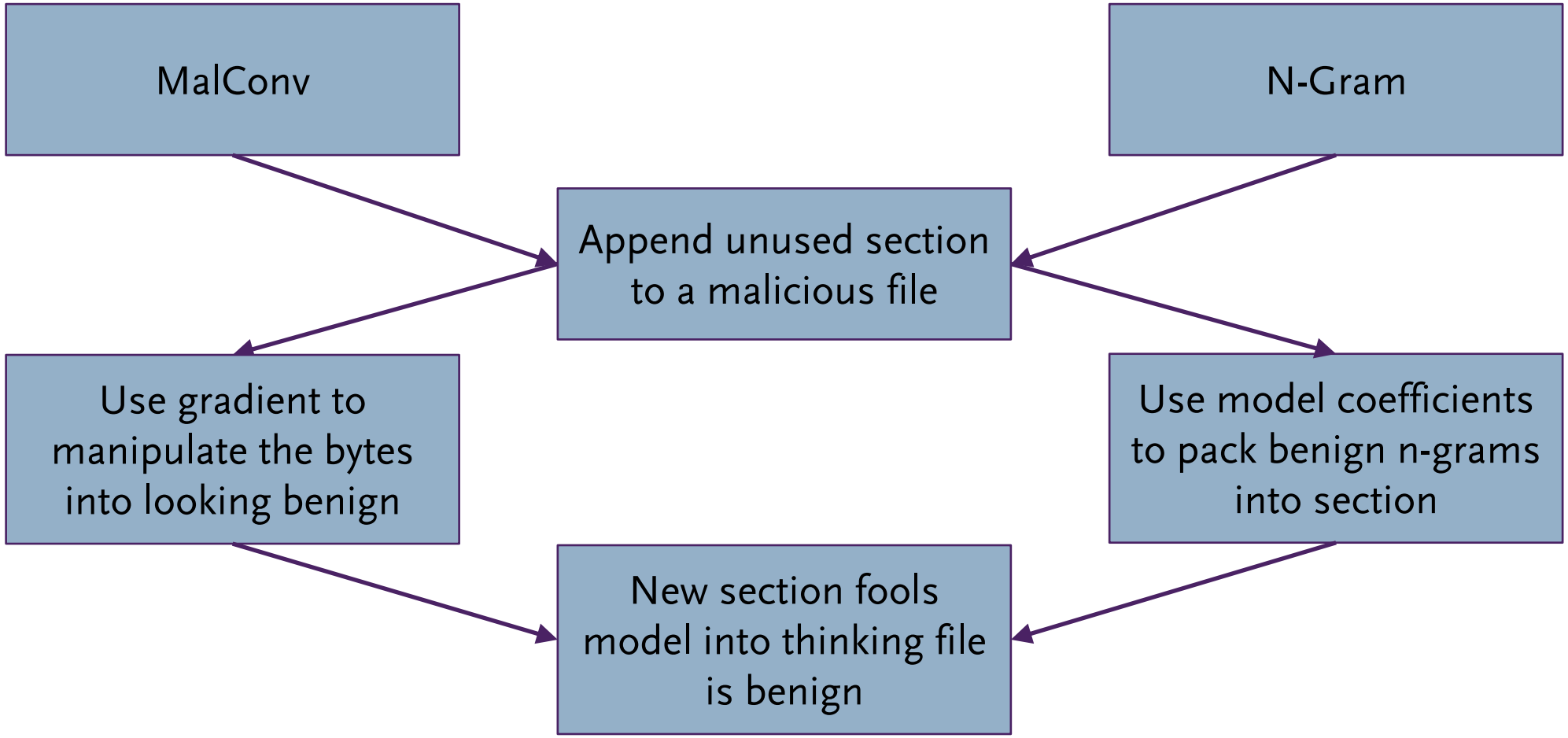
- Attacks against machine learning models
- Inputs are altered by an adversary in order to change the model's prediction.
- Attacks developed for Image Classifiers, Spam Filters, and even MalConv!



Goodfellow, et al. "Explaining and Harnessing Adversarial Examples." ICLR, 2015.



ATTACKS ON OUR ML MODELS

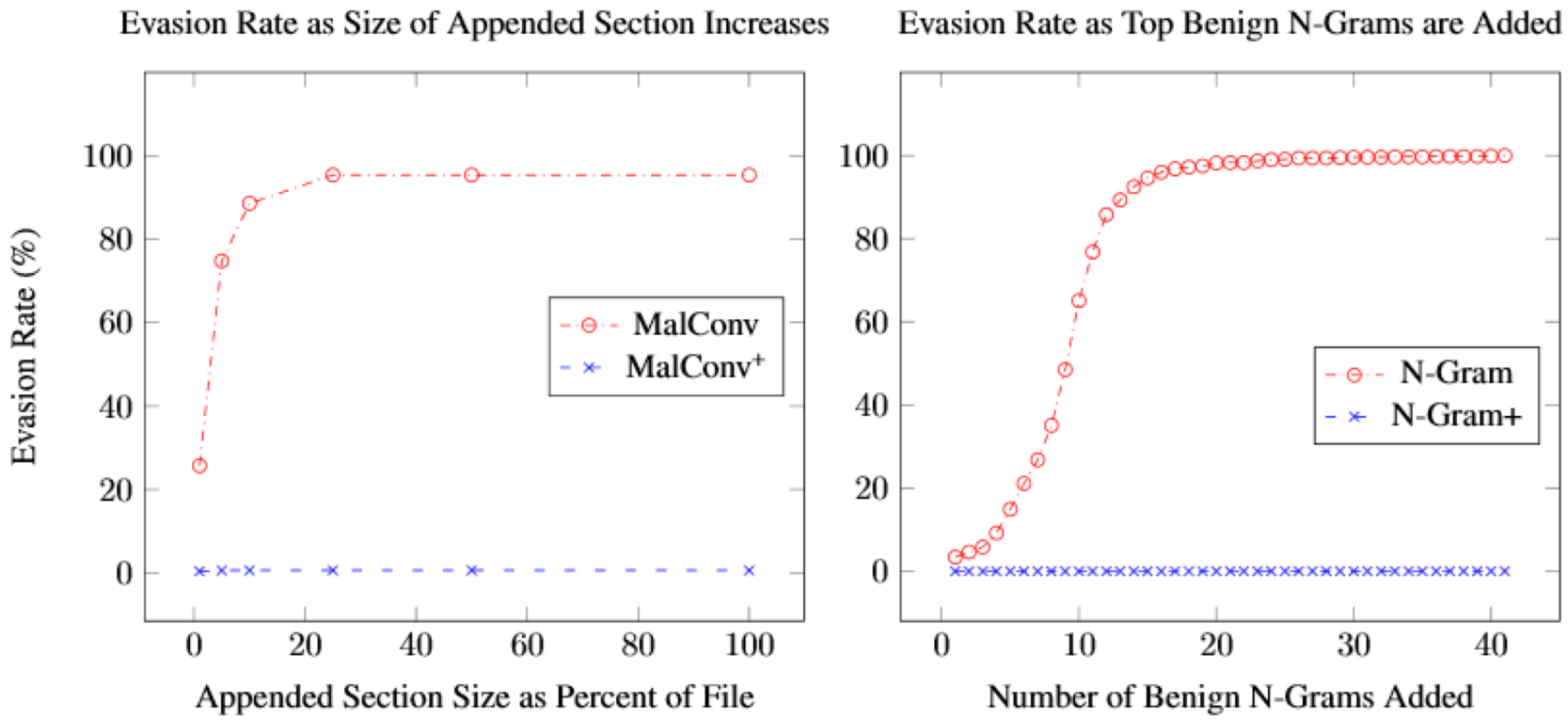


DEFENDING ML MODELS

- Constraining models to have only non-negative weights causes the model parameters associated with "good" features to go to zero.
- The concept of "good" is only captured as a threshold instead of balancing good vs. bad.
  - Count up how much bad stuff there is.
  - Ignore how much good stuff there is.
- Prevents adversaries from adding "goodness" to inputs.

COMPARISON: EVASION RATES

- MalConv: 95.4% evasion versus 0.6% with defense
- N-Gram: 100% evasion versus 0% with defense



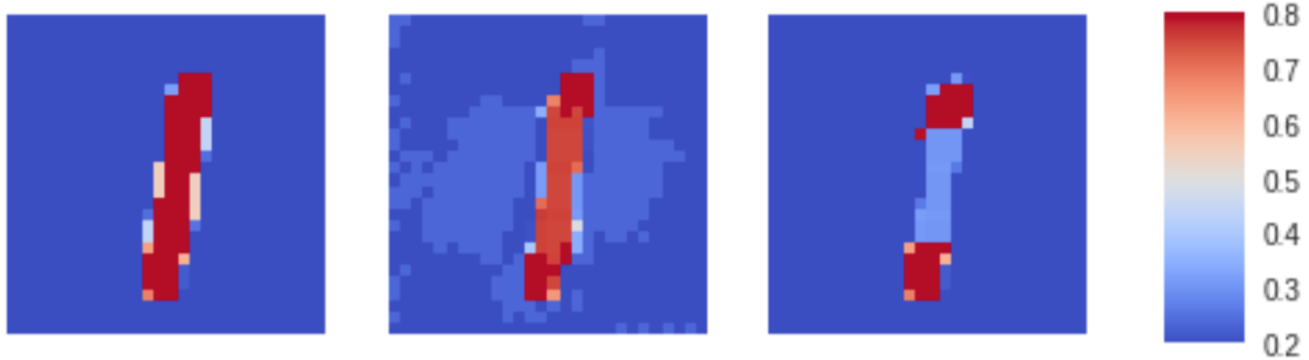
COMPARISON: ACCURACY

- Both models take a hit to accuracy but gain robustness
  - Lower accuracy when not under attack
  - *Much* higher accuracy when under attack
- Most errors come from reduced Recall
  - This aligns well with how AVs are deployed

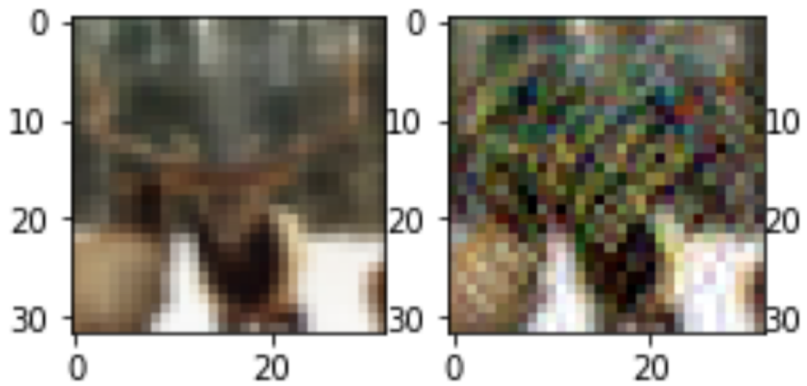
Classifier	Accuracy %	Precision	Recall	AUC %
MalConv	94.1	0.913	0.972	98.1
MalConv$^+$	89.4	0.908	0.888	95.3
N-Gram	95.5	0.926	0.987	99.6
N-Gram$^+$	91.1	0.915	0.885	95.5

MORE NON-NEGATIVE WEIGHTS

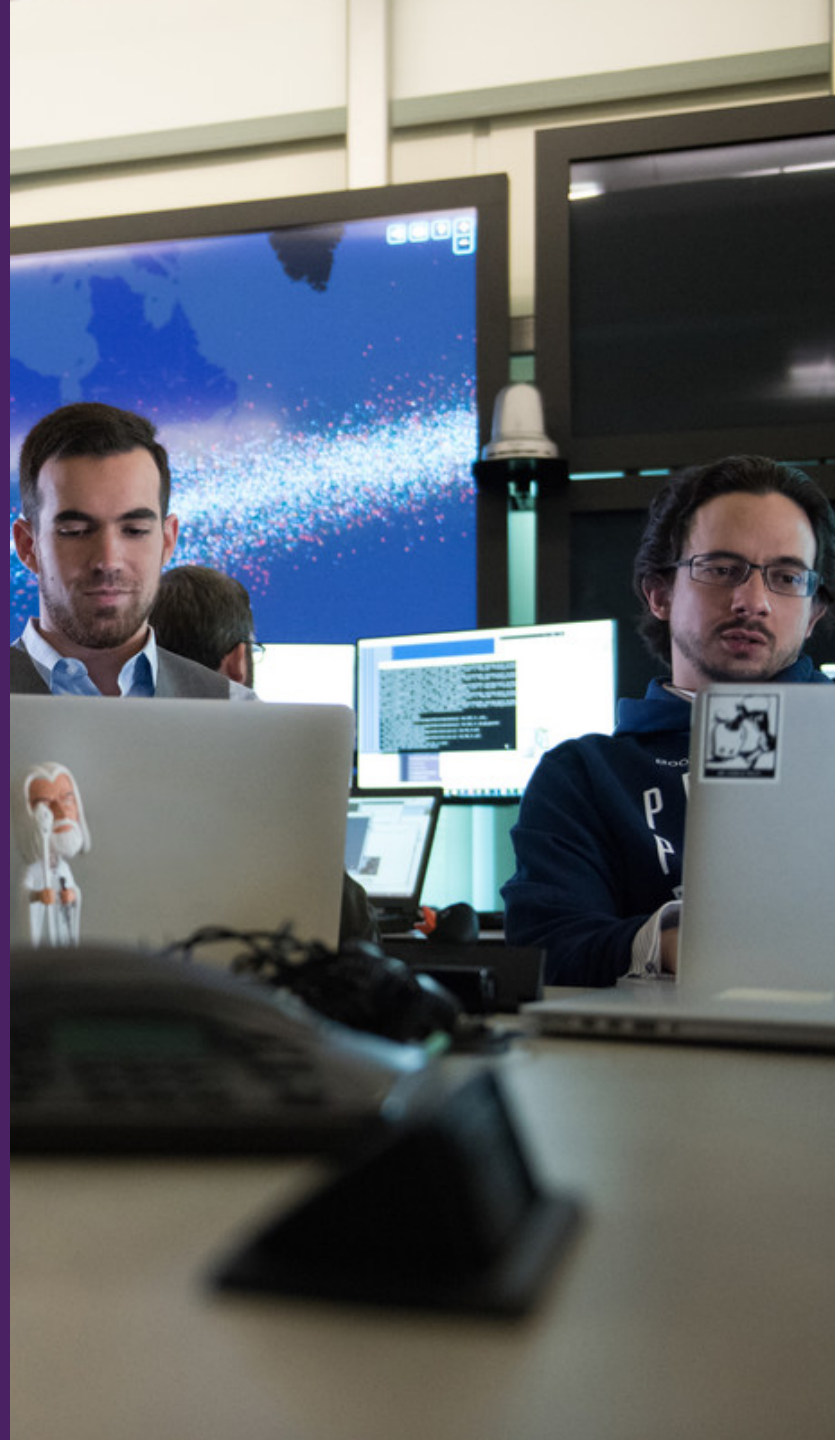
- We've used Non-Negative Weight Constraints in other domains as well
- Spam
  - "Good word" attack simply appends non-spam words to spam messages
  - Non-negative constraint out performs prior state-of-the-art defense
- Image classification
  - Having more than 2 output classes is a complication
  - Attacks forced to make larger modifications to inputs
  - Non-negative constraints defend well against targeted attacks



MNIST



CIFAR100



THANK YOU

The Laboratory for Physical Sciences

Quantifying Robustness: https://arxiv.org/abs/1806.04773

Non-Negative Networks: https://arxiv.org/abs/1806.06108

Capt. Will Fleshman

William.C.Fleshman.mil@mail.mil

@willcfleshman

Dr. Jared Sylvester

Sylvester_Jared@bah.com

@jsylvest

BOOZALLEN.COM/MACHINEINTELLIGENCE

CONSULTING | ANALYTICS | DIGITAL SOLUTIONS | ENGINEERING | CYBER