Defeating Machine Learning

What Your Security Vendor is Not Telling You

black hat USA 2015

BLUVECTOR

www.bluvectorcyber.com

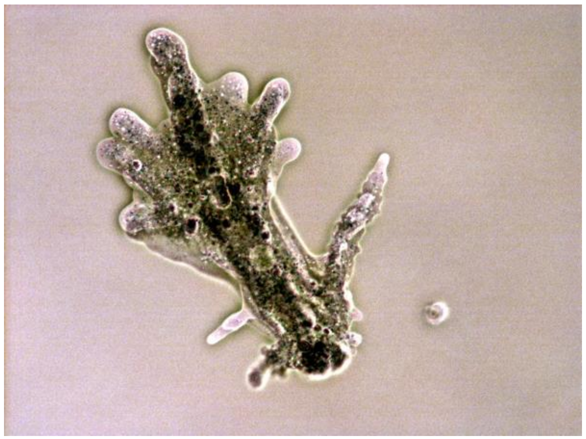
Bob Klein
Data Scientist
Bob.Klein@bluvectorcyber.com

Ryan Peters
Data Scientist
Ryan.Peters@bluvectorcyber.com

Agenda

- Security industry advances and the role of ML
- [DEMO] Attacker's perspective: How to defeat ML
- Solution: Defense through diversity
- Implementation discussion and results
- [DEMO] Attacker's perspective revisited
- Conclusions and paths forward

Evolution of the security industry

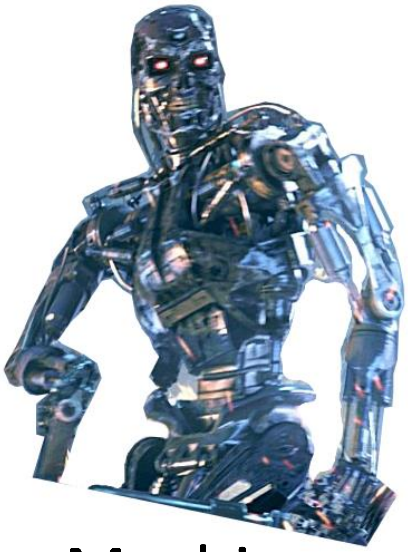


Signatures, Packet Filters

(+) Recognize known threats
(-) Very brittle

Heuristics, Sandboxes, Stateful Filters

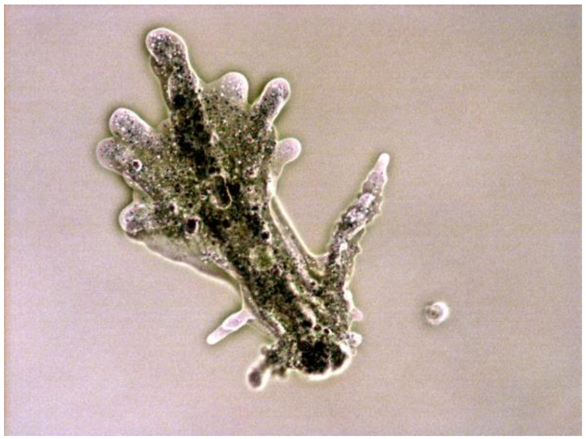
(+) Recognize malicious indicators
(-) Rely on known indicators



Machine Learning

~~(+) Unstoppable~~
~~(-) None~~

Evolution of the security industry



Signatures, Packet Filters

(+) Recognize known threats
(-) Very brittle

Heuristics, Sandboxes, Stateful Filters

(+) Recognize malicious indicators
(-) Rely on known indicators

Machine Learning

(+) Robust
(-) ??

The perils of a shared defense

Signatures,
Packet Filters

- (+) Recognize known threats
- (-) Very brittle
- (-) Shared signatures

The sharing of signatures among all deployments gives the attacker a significant advantage

The perils of a shared defense

Heuristics, Sandboxes, Stateful Filters

(+) Recognize malicious indicators

(-) Rely on known indicators

(-) <u>Shared ruleset / engine</u>

Newer technology using the same deployment paradigm is similarly vulnerable

The perils of a shared defense

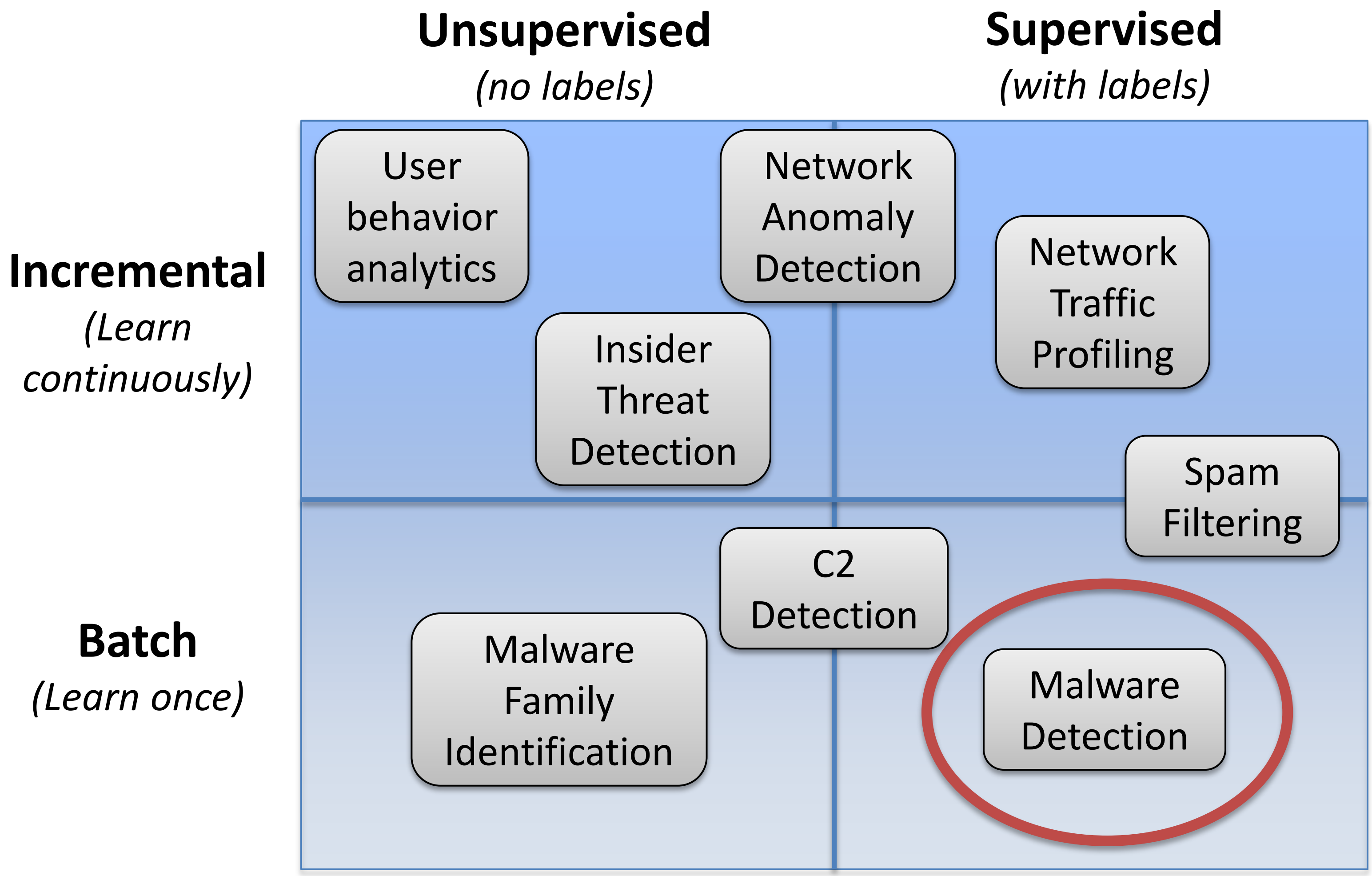
Machine Learning

(+) Robust

(-) <u>Shared models </u>(?)

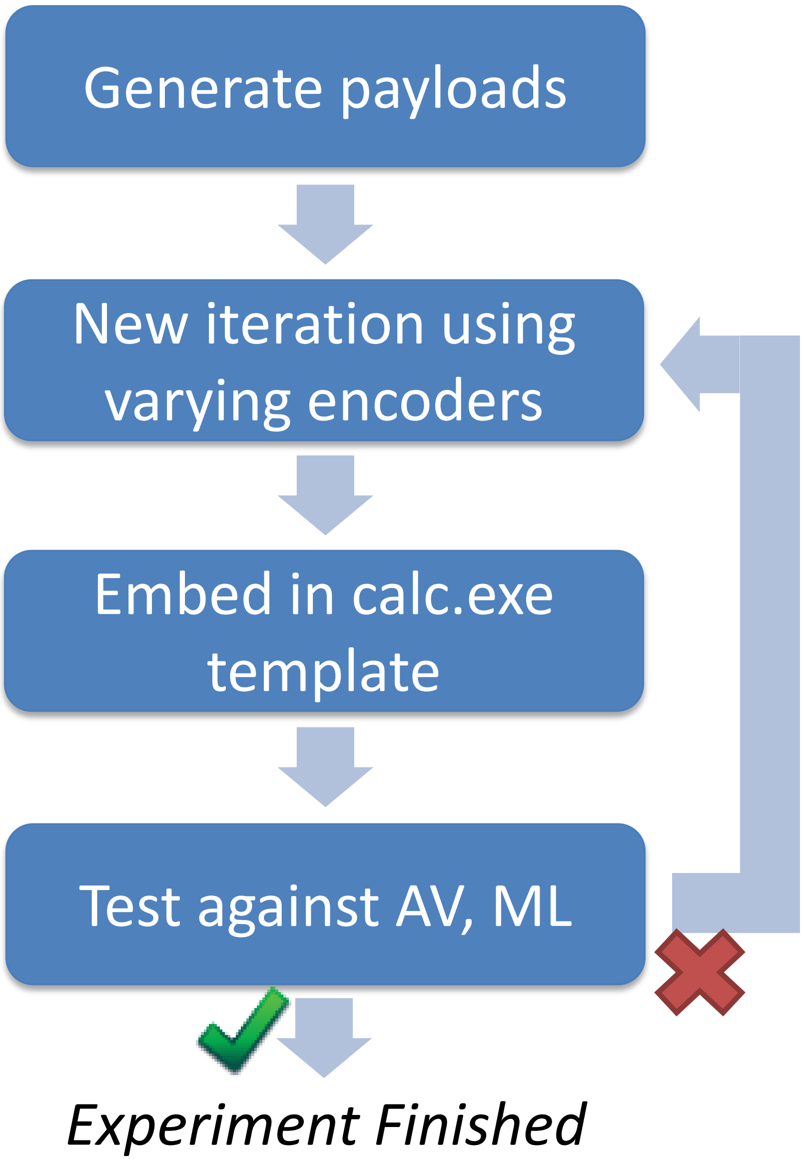
Some machine learning approaches may be exploitable by the same means

Machine Learning in cybersecurity



ML solutions for malware detection fail to break from the flawed deployment paradigm

Experimental Setup



Tools:

Metasploit 4.11.1

Payloads:

windows/meterpreter/reverse_tcp
windows/messagebox

Encoders:

x86/shikata_ga_nai
x86/call4_dword_xor
x86/jump_call_additive
etc.

Experimental Setup

AV Software:

ClamWin 0.98.7

Machine Learning Model:

Training list: 20,000 benign + 20,000 malicious samples

Test list holdout performance

Filetype	False Positives	False Negatives
PE32	3.5%	3.8%

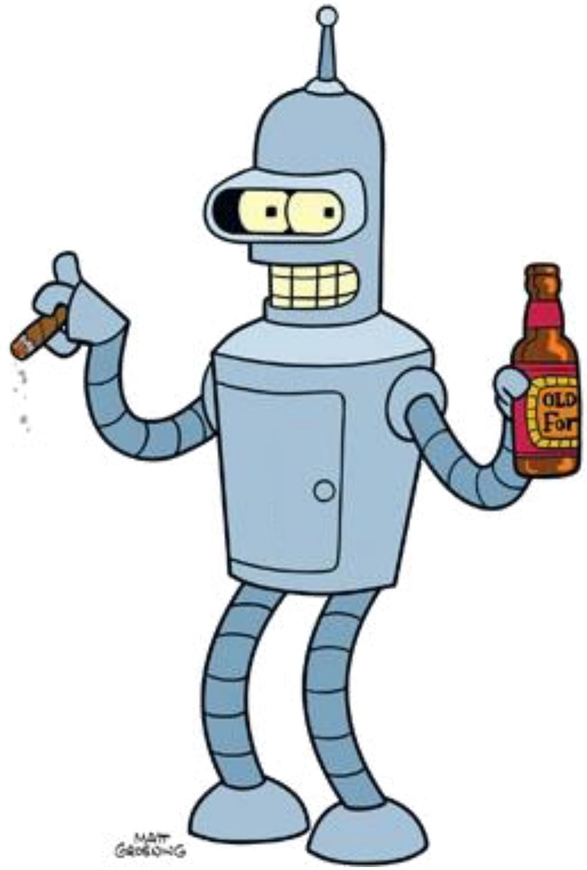
Assumptions:

Attacker has copy of AV and ML software

Attacker is unable to reverse engineer the software

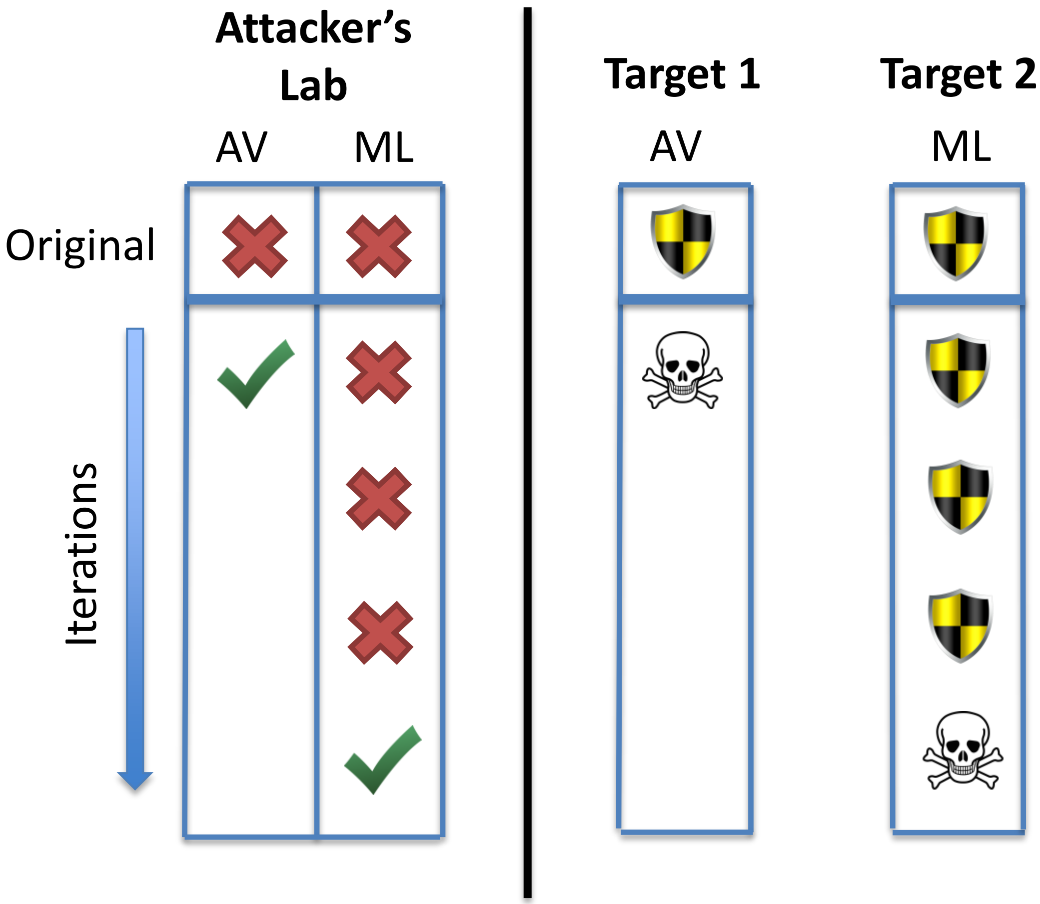
DEMO: AV vs ML, Attacker's Perspective

Demo: Lessons Learned



So what happened?

Demo: Lessons Learned



Attacker's Advantages:

- Confident model has not changed
- Confident all targets have the same model

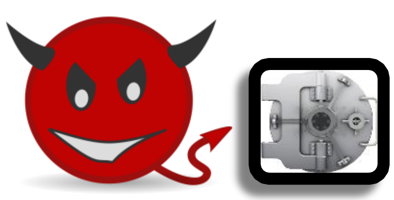
All it takes is persistence

Attacker holds significant advantages and can defeat target with enough persistence

How can we do better?

Traditional Defense

Moving Defense



Why hasn't this been done before?

- Logistical difficulty
- Cost to vendors
- Perceived risk to vendors

The Moving Defense concept addresses the issue but has not been widely implemented

Machine Learning: A Moving Defense

Feature Space

Learning Algorithm

Data Input

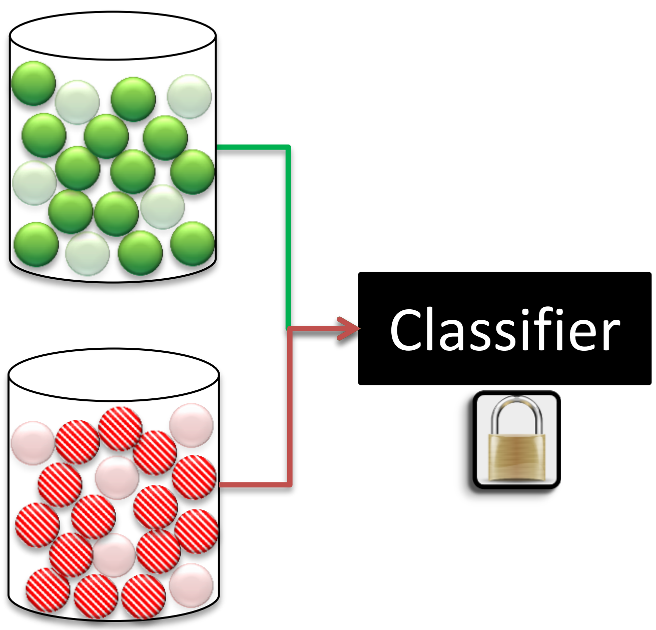
There are many ways to permute machine learning classifiers

Classifier Generation and Use

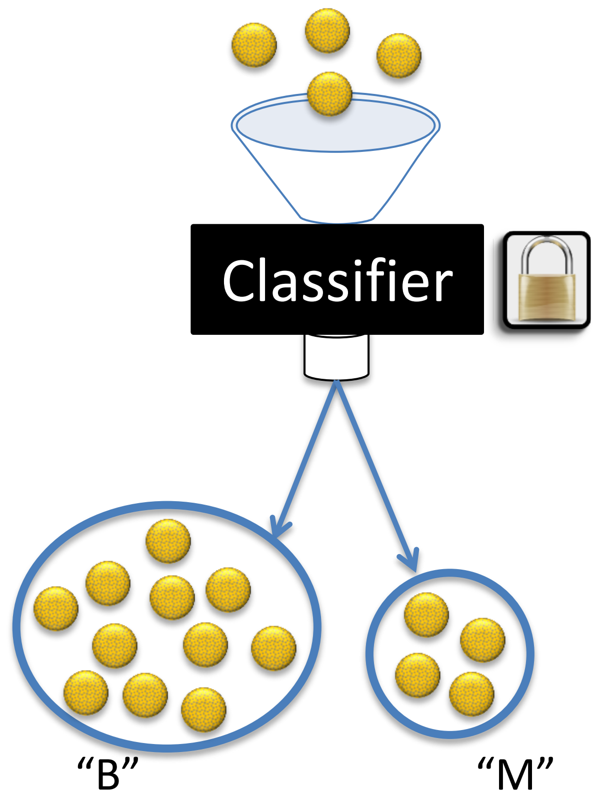
Vendor Lab

Library of
Benign Data

Library of
Malicious Data



User Environment



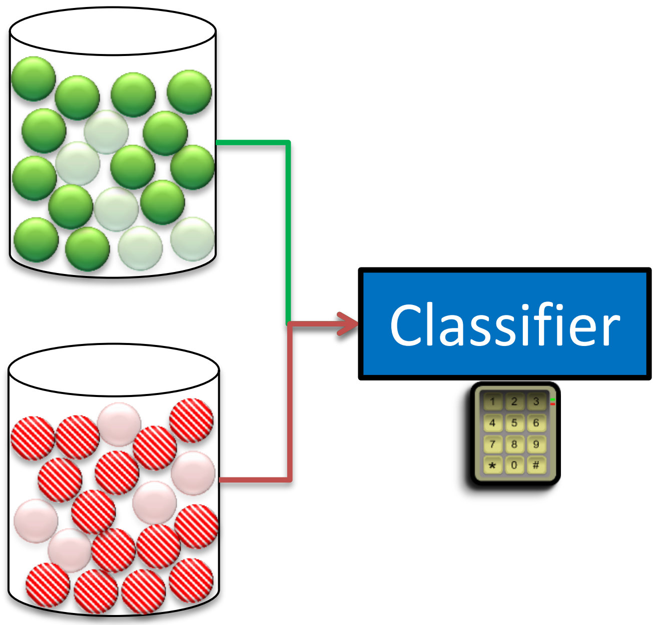
Moving Defense for ML: different data → different classifiers

Classifier Generation and Use

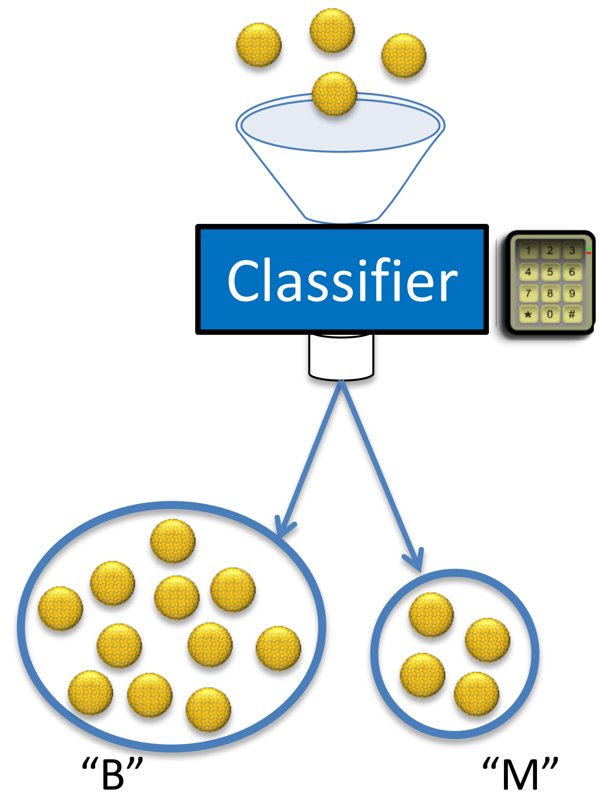
Vendor Lab

Library of
Benign Data

Library of
Malicious Data



User Environment



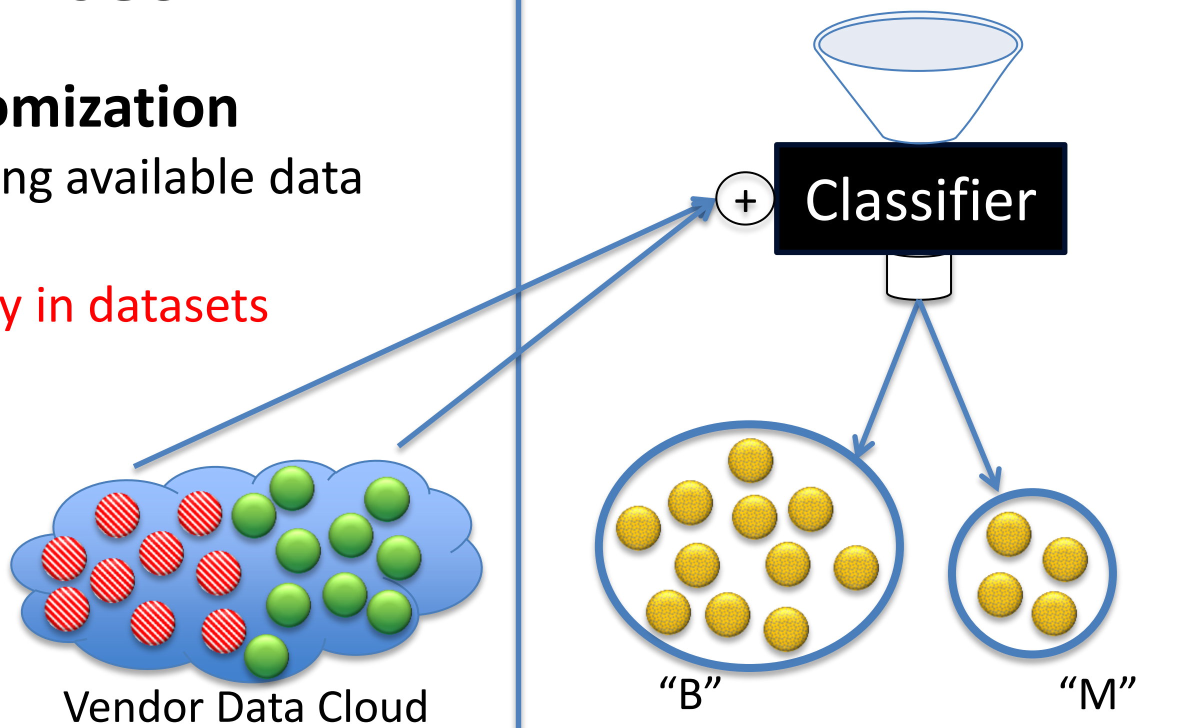
Moving Defense for ML: different data → different classifiers

Instantiating a Moving Defense Using Machine Learning

Data Sources

- **Vendor: Model Randomization**
  - ➢ Randomly select among available data provided by vendor
  - X No additional diversity in datasets

User Environment

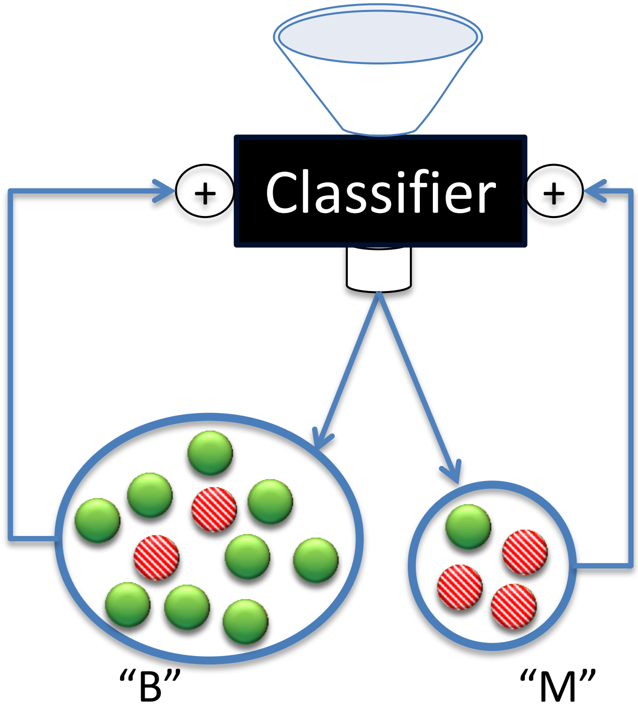


Instantiating a Moving Defense Using Machine Learning

Data Sources

- **Vendor: Model Randomization**
  - ➢ Randomly select among available data provided by vendor
  - X No additional diversity in datasets
- **Local: Model Reinforcement**
  - ➢ Feed back classifier-labeled samples into training set
  - X Only reinforces what the classifier already “thinks” it knows

User Environment

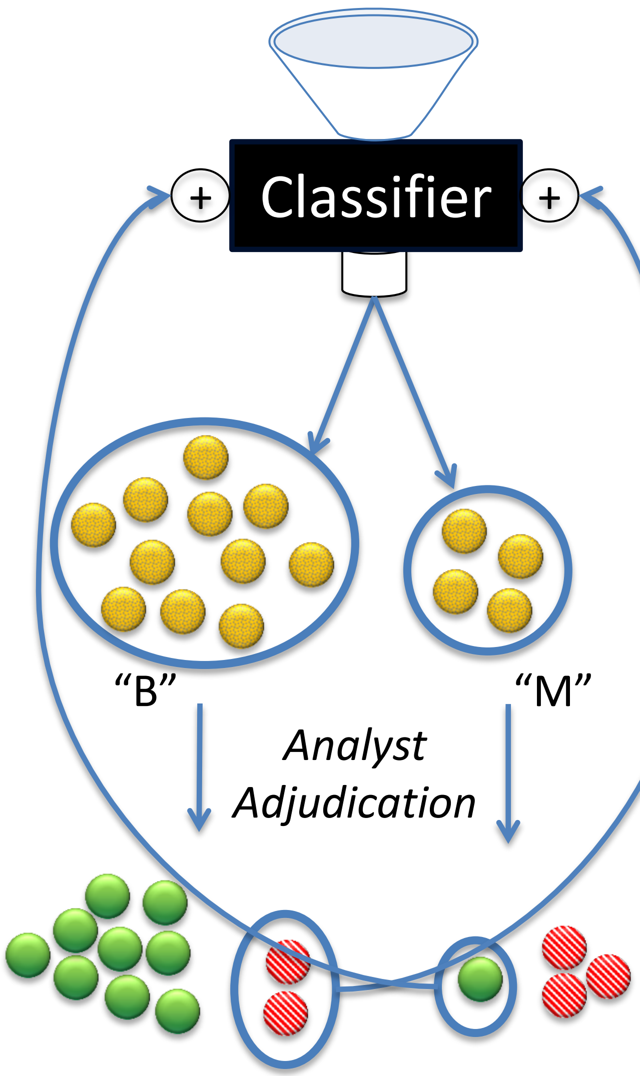


Instantiating a Moving Defense Using Machine Learning

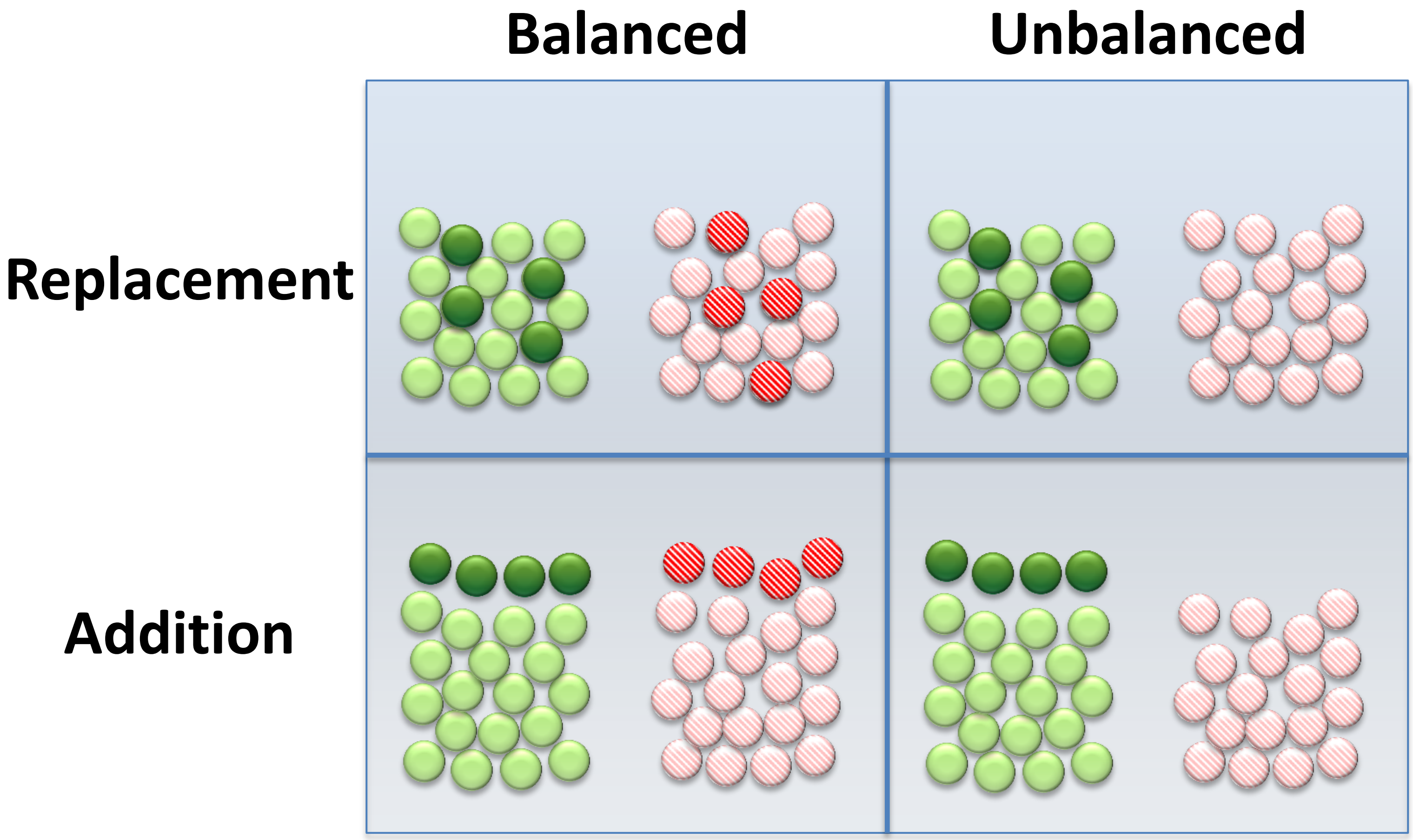
Data Sources

- **Vendor: Model Randomization**
  - ➢ Randomly select among available data provided by vendor
  - X No additional diversity in datasets
- **Local: Model Reinforcement**
  - ➢ Feed back classifier-labeled samples into training set
  - X Only reinforces what the classifier already "thinks" it knows
- **Local: Model Correction *("In-Situ")***
  - ➢ Feed back *errors*, correctly-labeled samples
  - ✓ Introduce new local knowledge to learner

User Environment



Considerations for Implementing In-Situ

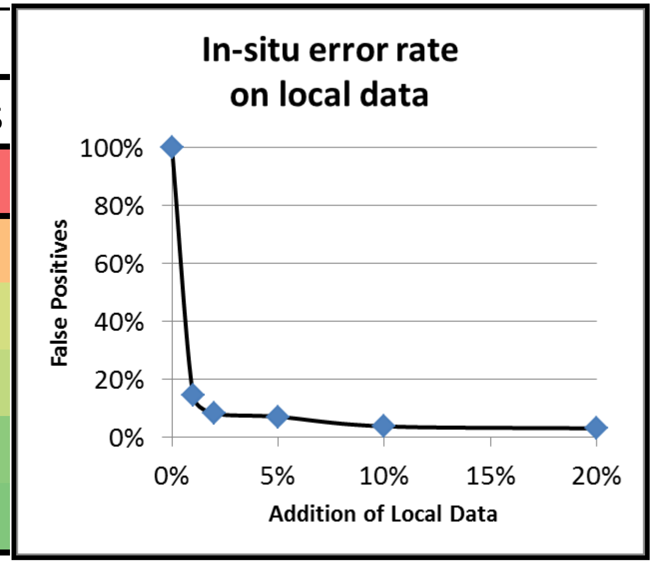


There are many factors to consider when operationally implementing in-situ

Experimental Results for In-Situ

Addition (unbalanced)

	Training Set Size		Test Set Performance
	benign (lab + local)	malware	local data: False Positives
base	20000 + 0	20000	100.0%
1%	20000 + 200	20000	14.4%
2%	20000 + 400	20000	8.3%
5%	20000 + 1000	20000	7.1%
10%	20000 + 2000	20000	3.8%
20%	20000 + 4000	20000	3.1%



In-situ classifiers perform equal or better than the base classifier

Experimental Results for In-Situ

Addition (unbalanced)

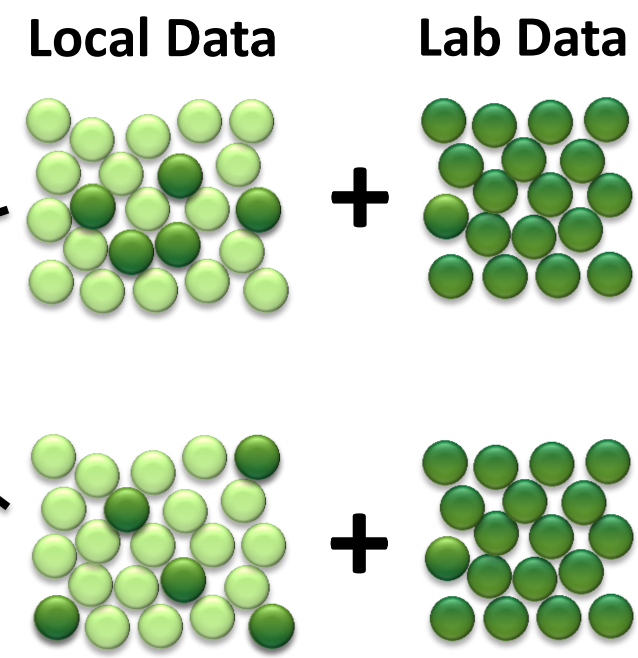
	Training Set Size		Test Set Performance		
	benign (lab + local)	malware	local data	lab data	
			False Positives	False Positives	False Negatives
base	20000 + 0	20000	100.0%	2.1%	3.3%
1%	20000 + 200	20000	14.4%	2.0%	3.8%
2%	20000 + 400	20000	8.3%	1.5%	4.2%
5%	20000 + 1000	20000	7.1%	2.5%	3.1%
10%	20000 + 2000	20000	3.8%	1.2%	3.9%
20%	20000 + 4000	20000	3.1%	1.9%	3.4%

In-situ classifiers perform equal or better than the base classifier

Experimental Results for In-Situ

Test Set Performance

	local data	lab data	
	False Positives	False Positives	False Negatives
base	100.0%	2.1%	3.3%
r1	6.9%	2.0%	3.3%
r2	7.1%	2.5%	2.9%



In-situ classifiers have equivalent performance between trials black hat USA 2015

Experimental Results for In-Situ

Test Set Performance

	local data	lab data	
	False Positives	False Positives	False Negatives
base	100.0%	2.1%	3.3%
r1	6.9%	2.0%	3.3%
r2	7.1%	2.5%	2.9%
r3	6.7%	2.2%	3.6%
r4	5.8%	1.7%	3.8%
r5	5.9%	2.4%	3.2%
r6	6.3%	2.3%	3.1%
r7	5.4%	1.6%	3.8%
r8	6.8%	2.4%	2.9%
r9	8.4%	3.5%	2.2%
r10	7.2%	2.0%	2.9%
MEAN:	**6.7%**	**2.3%**	**3.2%**
STDEV	**0.9%**	**0.5%**	**0.5%**

Generated 10 random in-situ classifiers using **5% addition (unbalanced)**

All in-situ classifiers showed similar overall performance

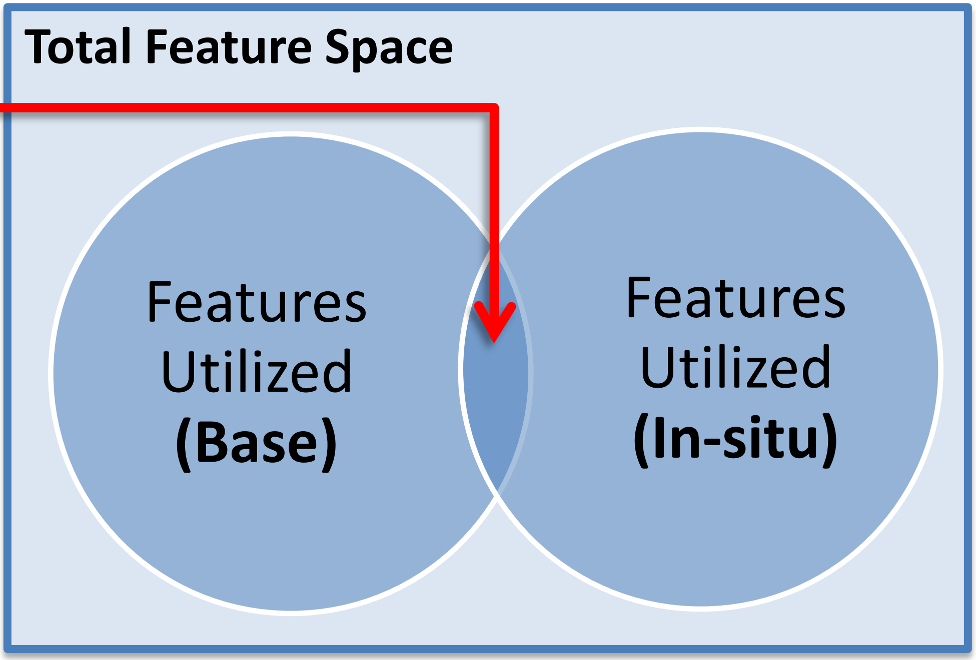
In-situ classifiers have equivalent performance between trials

Similarity of In-Situ Classifiers

Averaging across 10 in-situ models, compared to their base classifiers…

29%

Utilized feature space commonality



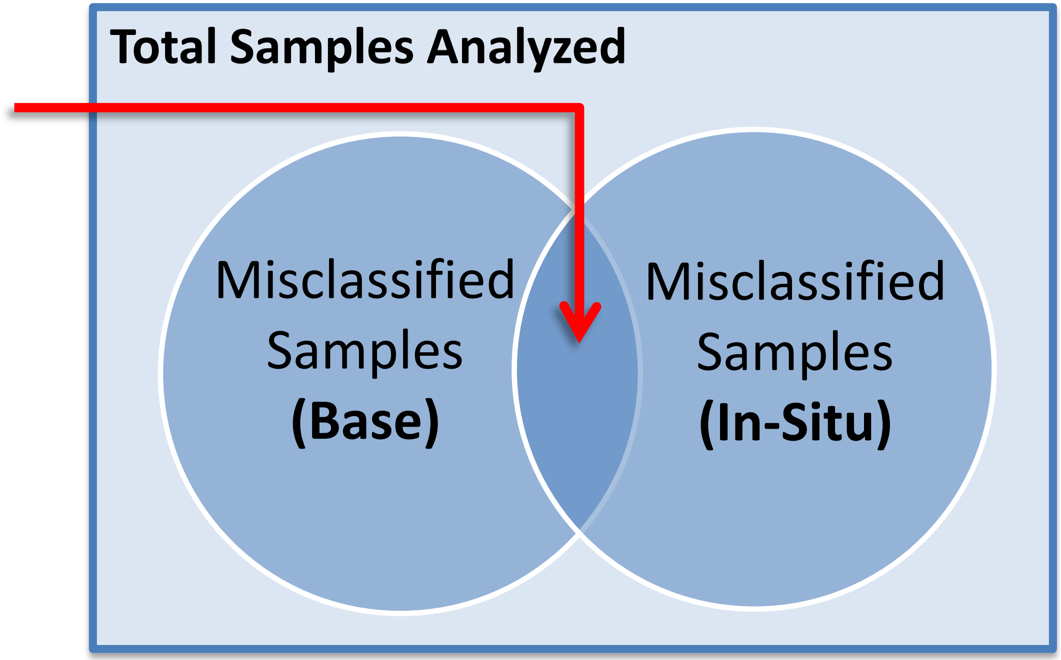
In-situ classifiers are very diverse from their base classifiers

Similarity of In-Situ Classifiers

Averaging across 10 in-situ models, compared to their base classifiers…

46%

Overlapping misclassifications



Misclassification = False Positive ***or*** *False Negative*

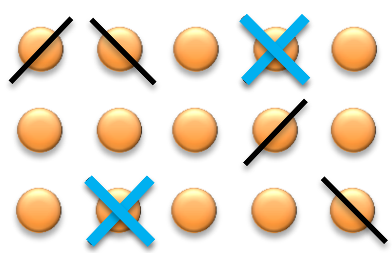
In-situ classifiers are very diverse from their base classifiers

Similarity of In-Situ Classifiers

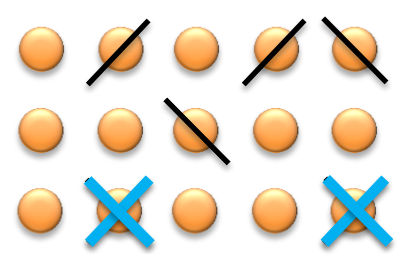
Overlapping Misclassifications

In-Situ	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
r1	100%	47%	46%	45%	43%	44%	42%	46%	40%	44%
r2		100%	48%	46%	51%	51%	45%	51%	50%	49%
r3			100%	48%	47%	44%	45%	42%	45%	46%
r4				100%	46%	48%	47%	46%	40%	48%
r5					100%	47%	47%	49%	44%	45%
r6						100%	45%	47%	44%	49%
r7							100%	41%	37%	44%
r8								100%	46%	45%
r9									100%	44%
r10										100%

r1 vs r2



r2 vs r4



In-situ classifiers show large diversity relative to other retrained classifiers

Similarity of In-Situ Classifiers

Overlapping Misclassifications

In-Situ	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10
r1	100%	47%	46%	45%	43%	44%	42%	46%	40%	44%
r2		100%	48%	46%	51%	51%	45%	51%	50%	49%
r3			100%	48%	47%	44%	45%	42%	45%	46%
r4				100%	46%	48%	47%	46%	40%	48%
r5					100%	47%	47%	49%	44%	45%
r6						100%	45%	47%	44%	49%
r7							100%	41%	37%	44%
r8								100%	46%	45%
r9									100%	44%
r10										100%

Any two given in-situ classifiers have a **46 ± 3%** overlap in misclassifications

In-situ classifiers show large diversity relative to other retrained classifiers

Experimental Setup

AV Software:

ClamWin 0.98.7

Machine Learning Model:

Training list: 20,000 benign + 20,000 malicious samples

Test list holdout performance

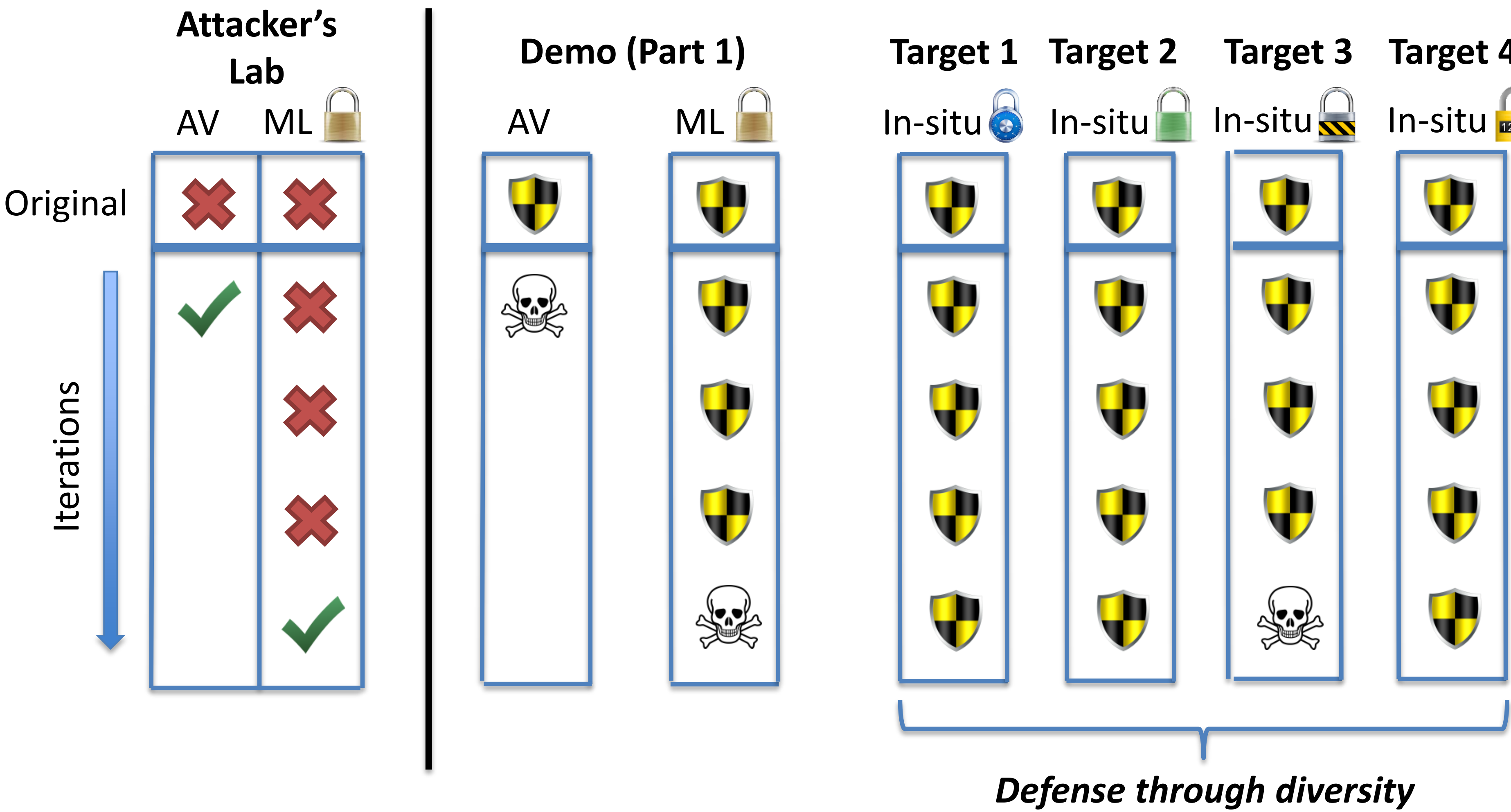
Filetype	False Positives	False Negatives
PE32	3.5%	3.8%

In-Situ Models:

Use 4 of the random models using 5% addition (unbalanced)

DEMO: In-situ Models, Attacker's Perspective

Demo: Lessons Learned



In-situ classifiers provide a moving defense against malware that defeats base model

Summary of benefits of in-situ

- Diversity of defense
- Environment-specific tailoring, performance
- Increased responsiveness
- No need to share personal or proprietary data

Black Hat Sound Bytes

- Improvements in ML methods for malware detection are weakened by their reliance on the traditional deployment paradigm
- The concept of a moving defense addresses this shared-model vulnerability and may be naturally applied to some ML solutions
- The diversity offered by a moving defense is “better for the herd” – users should engage with their vendors about its implementation

black hat®

USA 2015

UBM Tech