

Zen and the Art of Adversarial Machine Learning



Will Pearce

Red Team Lead
Azure Trustworthy ML
@moo_hax



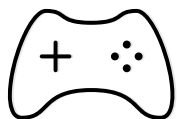
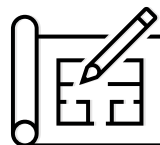
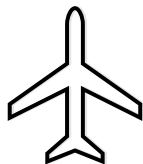
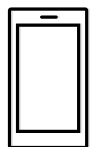
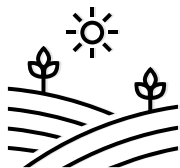
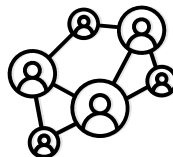
Giorgio Severi

PhD student
Northeastern University
@cloned_tweets

Operational Guidance

- Attacks, best practices, and where to start.
- Terms and gotchas to be aware of.
- Building capabilities.





Algorithms are empty

Models are not

Effectively Protection

**“Jim was our only
security person.
We cloned him
with AI. 100%
Return on
investment.”**



Endgame ReSec SourceDefense Strixus LogRhythm
Symantec Jask Armis ZecOps Perspecta ElasticSearch
Bromium Forcepoint CrowdStrike SovereignIntel
FireEye Zimperium NyoTron InfoBlox Patternx
F-Secure Splunk Sift CyberReason Panda Security Checkpoint
PerimeterX Palo Alto Versive Securonix Dell Lookout
Defender Mimecast Netsurion Vectra WhiteOps BlueCoat
DarkTrace Cynet Securiti Sepio Systems Vicarius Kaspersky Agari
InterSet Cyware TrUU GoSecure MobileIron
TrendMicro McAfee Cujo AI CyberBit Cylance Balbix Tessian
Code42 Webroot Shape Security Obsidian Security
Anomali Cyr3con Heimdel Solarwinds Rapid7
SparkCognition IBM Fortinet High-Tech Bridge
VadeSecure Prelert MalwareBytes
Intel Monkey Sophos Lastline CounterTack
DeepInstinct InterceptX Digital Guardian
Tanium RSA SilverTail

**95% of CISOs agree
that it might work!**

Adversarial ML

“Subdiscipline that specifically attacks ML algorithms”

- Find PII in large language models
 - Bypass classifiers
- Denial of Service with sponge examples
- Functional extraction for model theft

“I get my POCs on arXiv”

Thanks Professor!

How does a model representation of data align with current risk frameworks?

Is an ML system an Information System, and if so, who is responsible for securing it?

The background of the slide is a dark gray topographic map with white contour lines. The lines are irregular and wavy, creating a complex, organic pattern. A dark gray horizontal band is positioned across the lower third of the image, serving as a background for the title.

Attacks

Pre-Deployment



Poisoning

Post-Deployment



Extraction
Evasion
Inference
Inversion

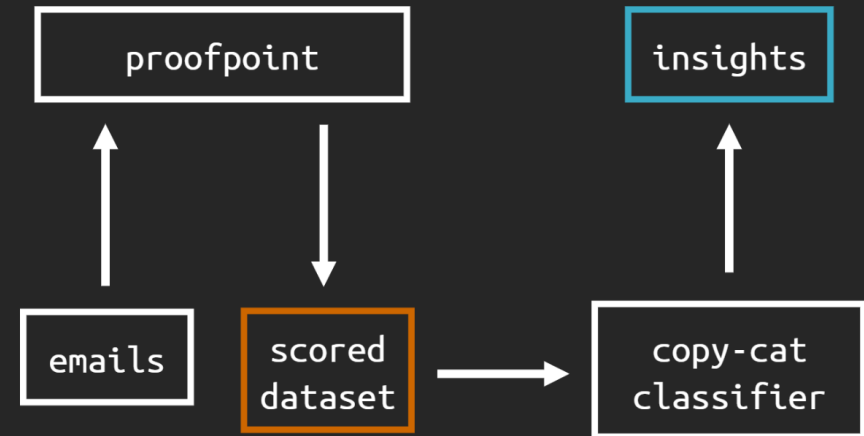
01 | Extraction

Creating a functionally equivalent model [7] is the **most fundamental** attack primitive.

1. Control over all inputs.
2. No adversarial examples*.
3. Transferability.
4. Provides options.
5. A simple attack.

@monoxgas

proofpoint - Attack (a)



Confirming our insights - Texts

Top 10 **highest** scoring words:

999

Random 10 words from the middle:

640

Top 10 **lowest** scoring words:

...

**mx0a-000a1001.pphosted.com gave this error:
This message looks too much like SPAM to accept.**

01 | Extraction

Requirements

- Initial dataset
- Ability to submit input and observe output

Outcome

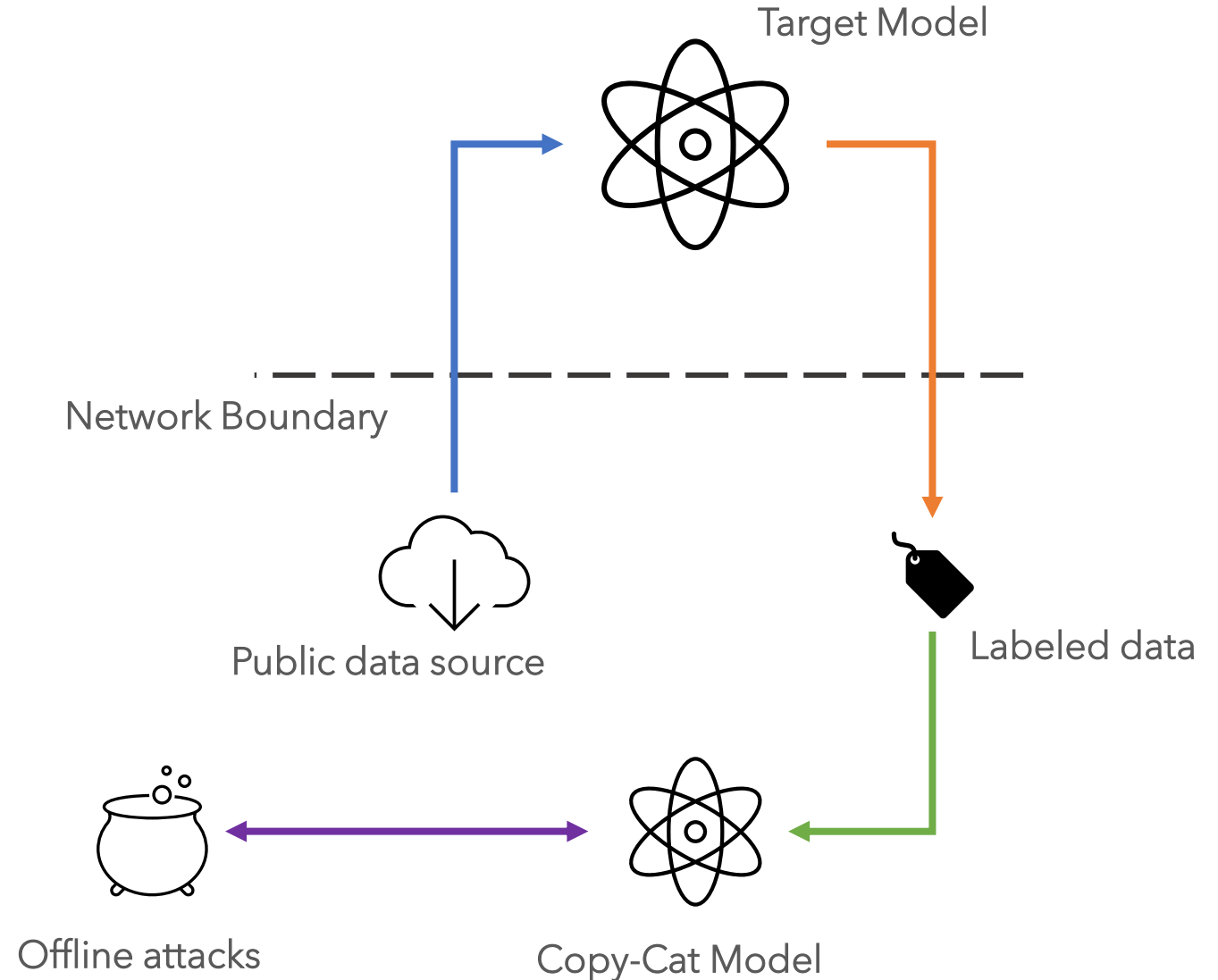
- Local copy of a model

Algos

- CopycatCNN [5]
- Functionally Equivalent Extraction [8]

When to use it?

- All the time



02 | Evasion

Adversarial 101. Is most concerned with bypassing classifiers.

1. Control over the initial samples.
2. Noisy inputs.
3. More direct than extraction.
4. Lots of variations.
5. Parameters are make-or-break



HopSkipJump targeted attack

02 | Evasion

Requirements

- Initial sample
- Ability to modify the input

Outcome

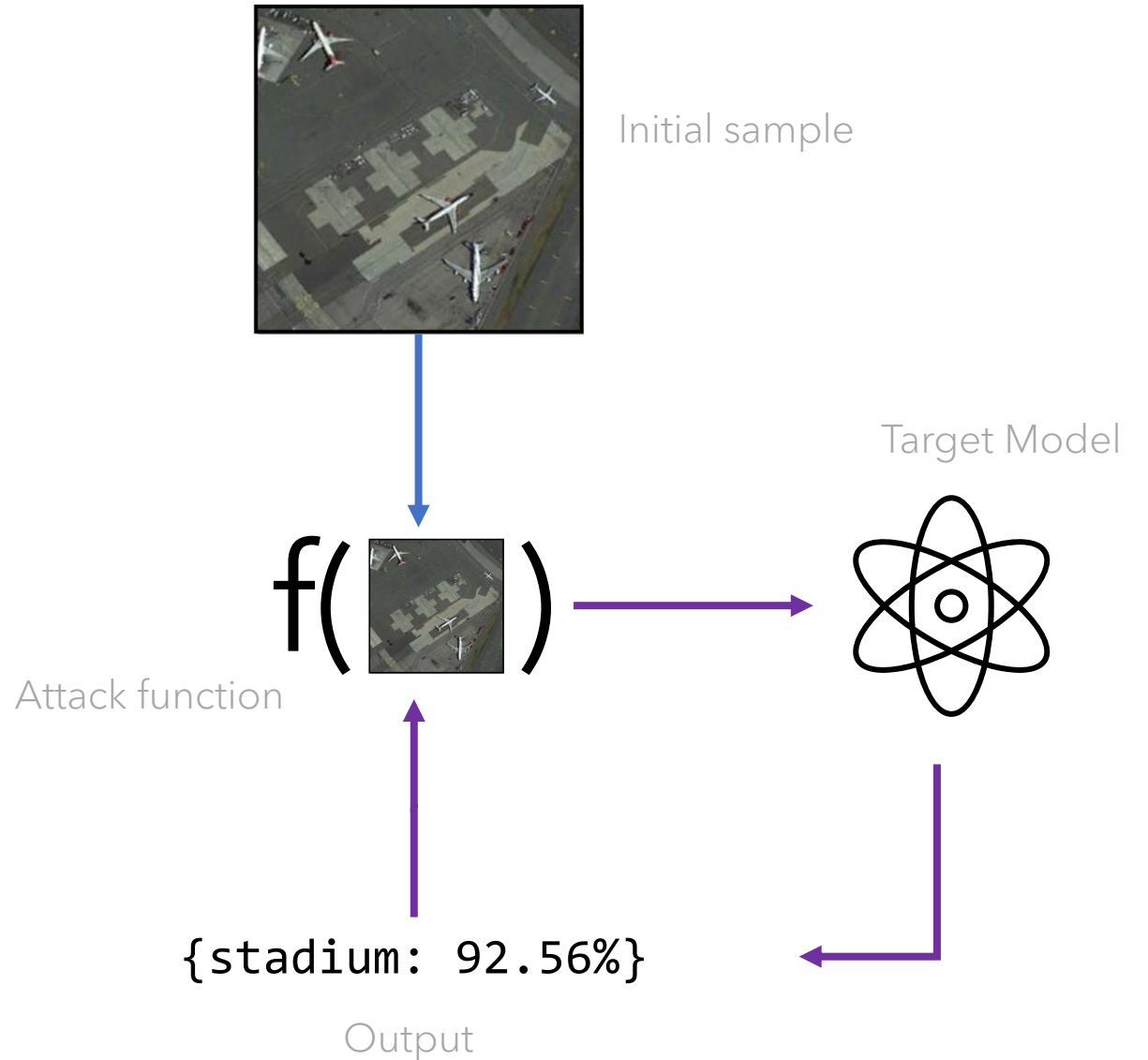
- Misclassified sample

Algos

- HopSkipJump (hard labels) [3]
- Square Attack (requires scores) [2]

When to use it?

- Bypass classifiers (spam, malware, auth)



03 | Inversion

Recover training data from a trained model. Requires knowledge of labels.

1. Can only reconstruct a representation of data for images.

2. Large language models become a valuable target.

Original



Recovered



Fredrikson et al, 2015

03 | Inversion

Requirements

- Information of a target label
- Ability to submit input and observe outputs that include confidence scores

Outcome

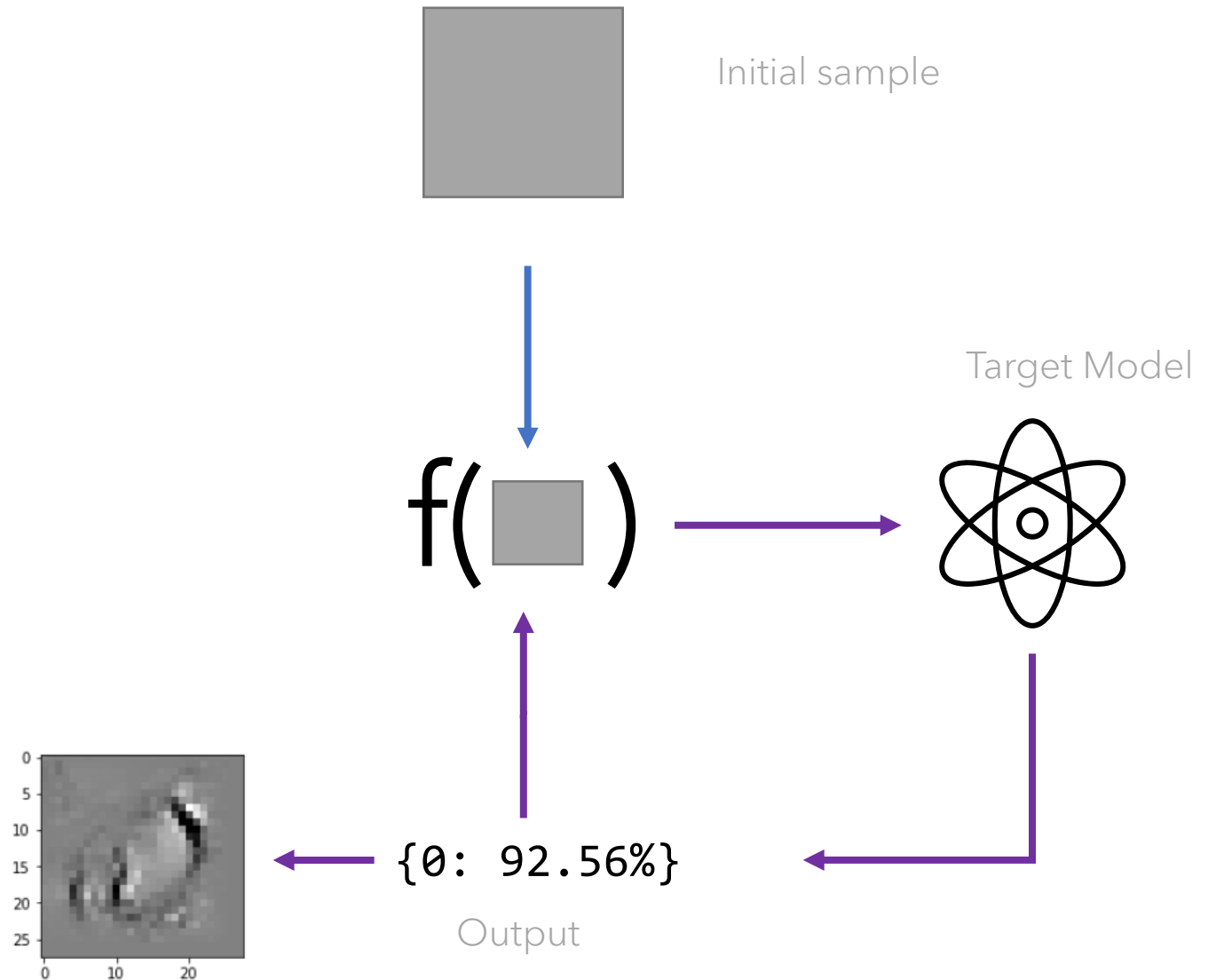
- Representation of the target class

Algos

- MI-Face [6]

When to use it?

- Looking for private data



04 | Inference

Determine if a data point was in a training set. Exploits confidence about inputs a model has seen before.

1. Two types, membership and attribute inference.
2. Triangulation of information
3. Blackbox



04 | Inference

Requirements

- Any data point.

Outcome

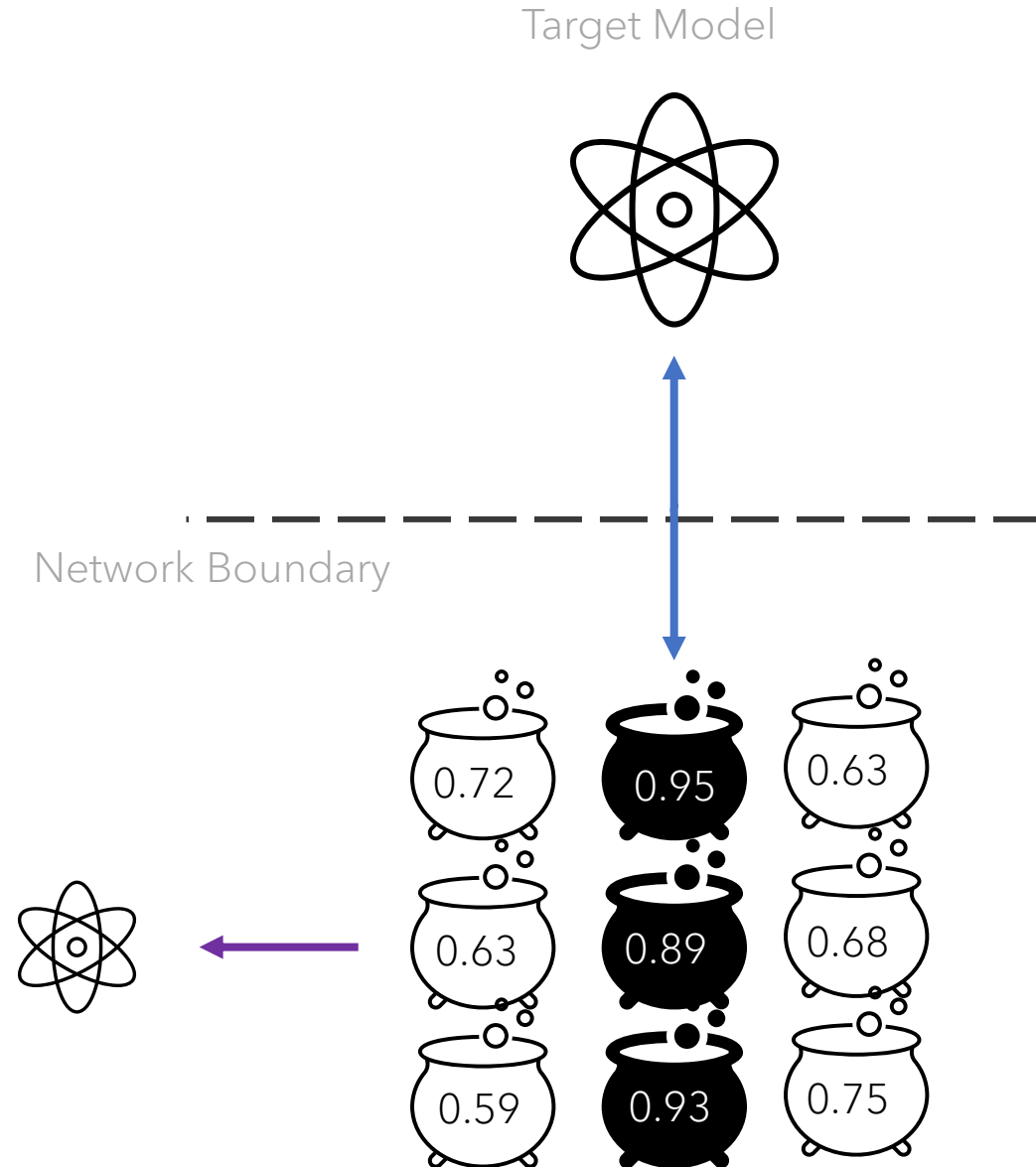
- Confirmation a data point was in the training set

Algos

- Label-Only Boundary Distance Attack [4]

When to use it?

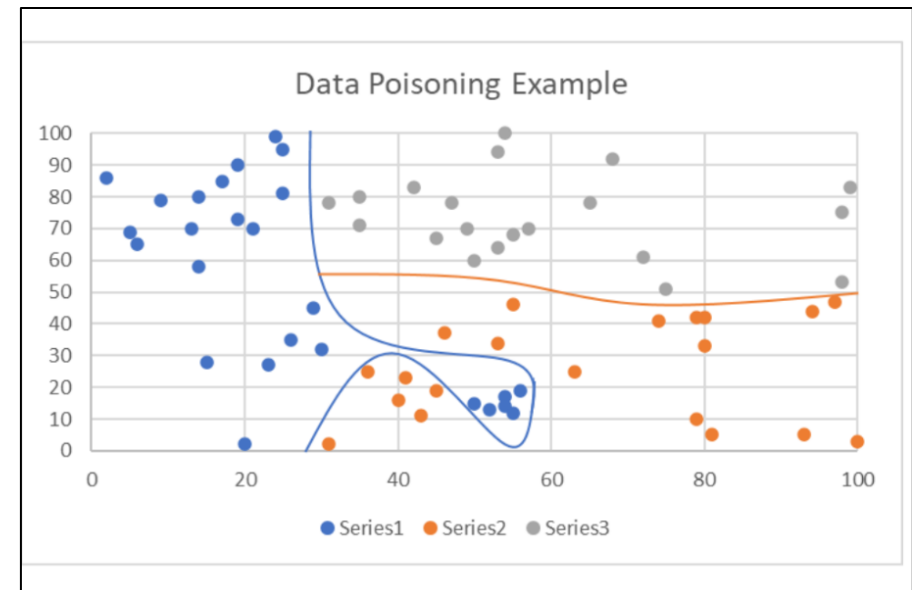
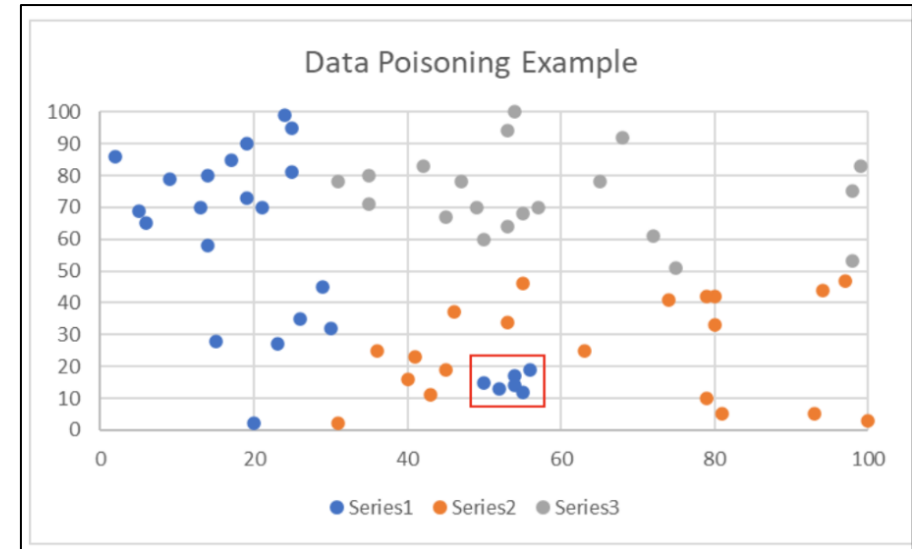
- Infer private information about a participant in the training set.



05 | Poisoning

Influence the creation or acceptance of a model for exploitation in a deployed setting.

1. Spectrum of objects
2. Need control over training data
3. Impact vs Stealth trade-off



05 | Poisoning

Requirements

- Ability to tamper with the training process
- Usually by injecting or modifying data in the training set

Outcome

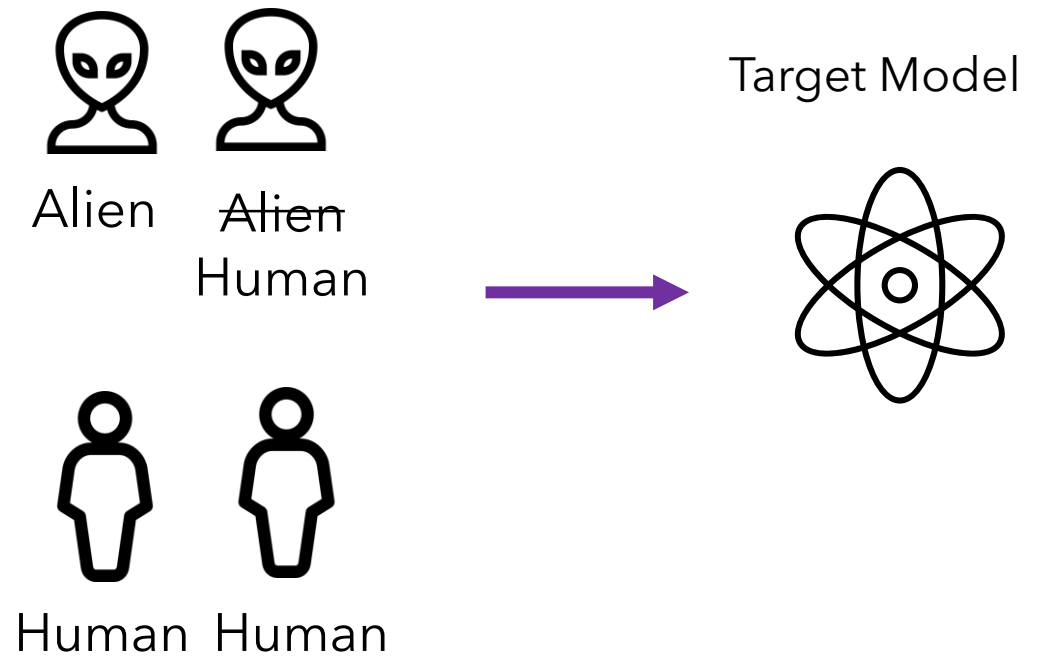
- Corrupted deployed model

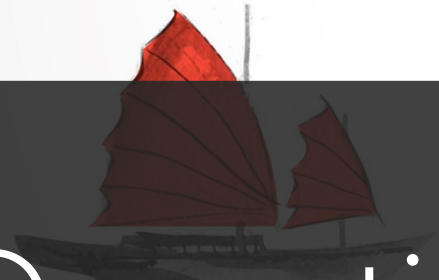
Algos

- BadNets (Backdoor) [7]
- Bullseye Polytope [1]

When to use it?

- If you understand the model and have access





Operational Guidance

01 | Hard vs Soft Labels

- These provide information about in which direction the changes are moving the classification output.

(More information the better)

Model Outputs

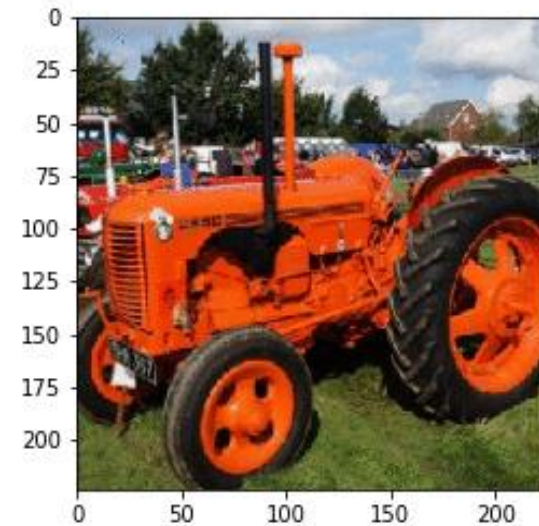
`{label: cat}` Good

`{cat: 999}` Better

`{cat: 850, dog: 149}` Best

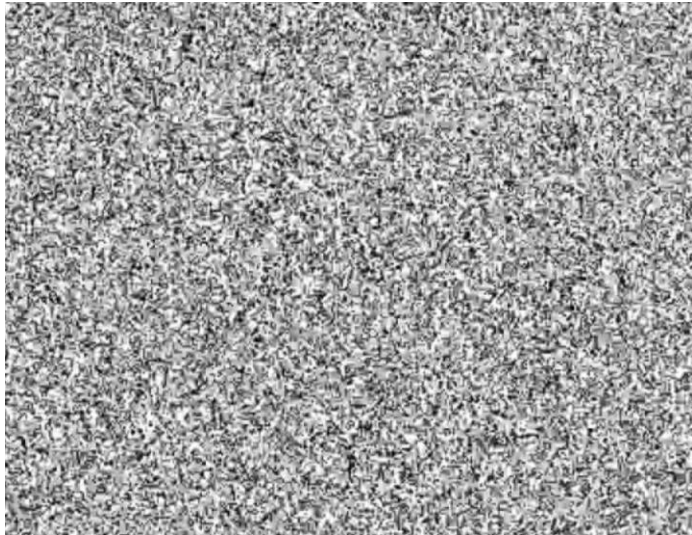
02 | Lossy Compression

1. Run an attack
 2. Save image as JPG
 3. JPG runs compression algorithm and ruins your work.
- Anyone ever had a payload they encoded only to get something slightly different on the other side?



03 | Algorithm Behavior

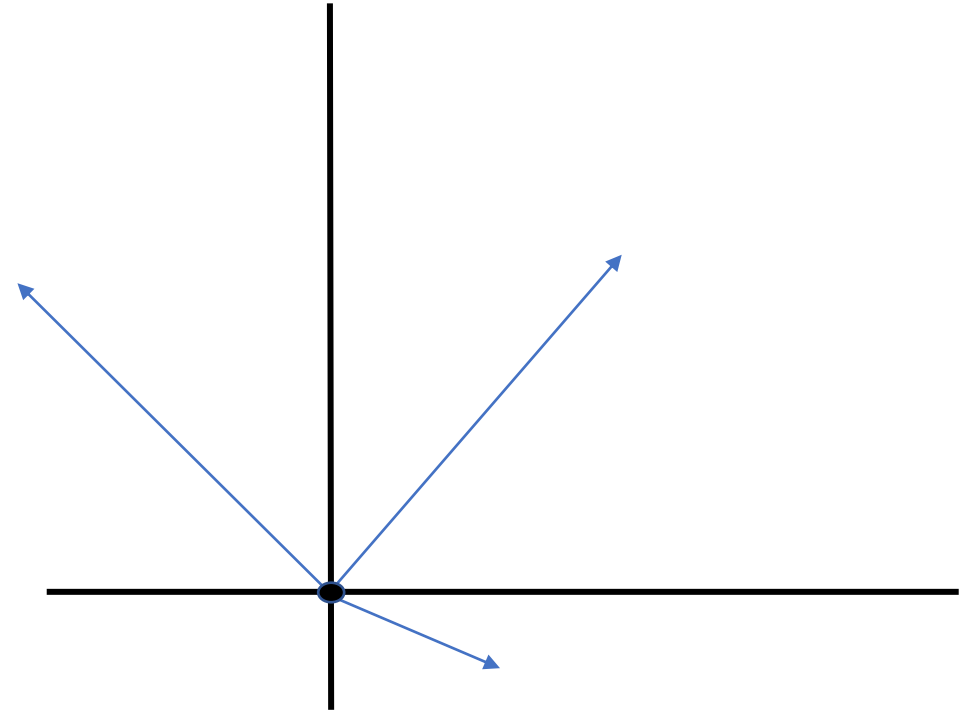
- First Hop Skip Jump image vs final (can you change it?)



04 | Distance Metrics

- Euclidean
- Manhattan
- Infinity

- In the offsec space, we do this with new techniques. Instead, do it with an algorithm.





Attack Surface

01 | OSINT

Sites & Dorks

- Greyhatwarefare.io, shodan.io
- inurl: score

Tools

- Fingerprinting servers with LobotoML from @alkae_t
- Match patent sources with Arxiv submission

Documentation

- Sophos Intercept X Docs
- Adaptive MFA Doc
- Windows Hello Docs

Filename
semistructstore/chlaksh/L1/20200616_06...d35-810a-e29cedf9e53a/pytorch_model.bin
semistructstore/chlaksh/L1/20200626_03...921-a907-2c14a8ed89f1/pytorch_model.bin
semistructstore/chlaksh/L1/20200626_03...9f1/checkpoint-230000/pytorch_model.bin
semistructstore/chlaksh/L1/20200626_03...9f1/checkpoint-200000/pytorch_model.bin
semistructstore/chlaksh/L1/20200626_03...9f1/checkpoint-180000/pytorch_model.bin
semistructstore/chlaksh/L1/20200623_21...2e6-8581-ab90b4b8cbc1/pytorch_model.bin

02 | Inference Traffic

Headers

- Cloud-Spam-Score
- X-Proofpoint.*

Numeric values in seemingly arbitrary places

- Confidence scores, probabilities
- Labels

```
To: <reciever@domain.com>
From: <sender@domain.com>
Subject: Our Meeting
...
X-Proofpoint-Spam-Details: rule=nodigest_notspam policy=nodigest score=0
malwarescore=0 mlxlogscore=999 mlxscore=0 suspectscore=14 spamscore=0
impostorscore=0 adultscore=0 clxscore=593 priorityscore=0 phishscore=0
bulkscore=97 lowpriorityscore=97 classifier=spam adjust=0 reason=mlx
scancount=1 engine=9.1.0-12345000 definitions=main-12345
```

```
Creating stream object with file name: PowerView.ps1
Calling antimalware->Scan() ...
...
Scan result is 32768. IsMalware: 1
Provider display name: Microsoft Defender Antivirus
Leaving with hr = 0x0
```

03 | Common Files



Common file extensions




- h5, hdf5, avro, ckpt, csv, npy, onnx, pkl, pb, mlmodel, pt, pth, pmml, zip, jsonl, arquet, orc, petastorm, netcdf, yaml, tfrecords, arff, lp, mps, sav, oprm, cpo, mod, dat, oplproject



Framework DLLs

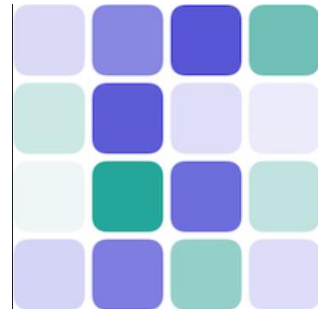
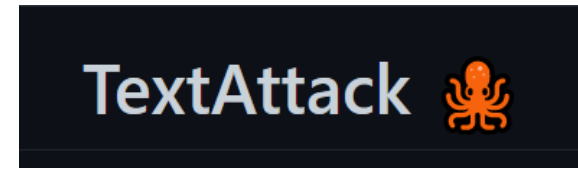
- onnx.dll, tensorflow.dll
- Windows.AI.MachineLearningnamespace



 Windows.AccountsControl.dll	1/16/2021 12:44 PM
 Windows.AI.MachineLearning.dll	9/8/2020 5:03 PM
 Windows.AI.MachineLearning.Preview.dll	1/16/2021 12:44 PM

04 | Tooling

- [Adversarial Robustness Toolbox](#)
- [TextAttack](#)
- [SecML](#)
- [Augly*](#)
- [Foolbox](#)
- [Armory](#)
- [TextFooler](#)
- [Counterfit](#)
- [Cleverhans](#)



Capability Development

- **Collect and store data (or generate)**
 - VBA Macros, Images, PowerShell Scripts
- **Collect and store adversarial examples.**
 - Think of them like TTPs
- **Collect and store algo parameters.**
 - They will lower costs long-term
- **Train and store models**
 - Use them for transferability
- **Build infrastructure to support**
 - Congrats, you're an ML engineer!





Conclusion



The Zen

Everything you can find in a model is
already there.

The same techniques used to build are the
same techniques used to break.

Conclusion

- **These attacks aren't "futuristic"**
 - They're kind of "simple" - Dunning-Kruger
- **A lot of security activities transfer**
 - Logging, access management,
- **Implicit relationship between academia and industry.**
 - New TTPs on Arxiv



Thank You!

References

1. Aghakhani, Hojjat, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. 2021. "Bullseye Polytope: A Scalable Clean-Label Poisoning Attack with Improved Transferability." ArXiv:2005.00191 [Cs, Stat], March. <http://arxiv.org/abs/2005.00191>.
2. Andriushchenko, Maksym, Francesco Croce, Nicolas Flammarion, and Matthias Hein. 2020. "Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search." In European Conference on Computer Vision. ECCV. <http://arxiv.org/abs/1912.00049>.
3. Chen, Jianbo, Michael I. Jordan, and Martin J. Wainwright. 2020. "Hopskipjumpattack: A Query-Efficient Decision-Based Attack." In 2020 IEEE Symposium on Security and Privacy (Sp), 1277-94. IEEE.
4. Choquette-Choo, Christopher A., Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. "Label-Only Membership Inference Attacks." ArXiv:2007.14321 [Cs, Stat], January. <http://arxiv.org/abs/2007.14321>.
5. Correia-Silva, Jacson Rodrigues, Rodrigo F. Berriel, Claudine Badue, Alberto F. de Souza, and Thiago Oliveira-Santos. 2018. "Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data." In 2018 International Joint Conference on Neural Networks (IJCNN), 1-8. <https://doi.org/10.1109/IJCNN.2018.8489592>.
6. Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart. 2015. "Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 1322-33. CCS '15. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2810103.2813677>.
7. Gu, T., K. Liu, B. Dolan-Gavitt, and S. Garg. 2019. "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks." IEEE Access, SPECIAL SECTION ON ADVANCED SOFTWARE AND DATA ENGINEERING FOR SECURE SOCIETIES, 7: 47230-44. <https://doi.org/10.1109/ACCESS.2019.2909068>.
8. Jagielski, Matthew, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. "High Accuracy and High Fidelity Extraction of Neural Networks." ArXiv:1909.01838 [Cs, Stat], March. <http://arxiv.org/abs/1909.01838>.