Overview

- How AI models are different from traditional software
- Traditional endpoint software attacks
- AI model-specific attacks
  - Polluted training data
  - Out-of-sample adversarial data input
  - Back door neurons embedded in benign models

Conclusions
AI models are different from software

• Most of the “programming” is learned from training data
  • Input + Algorithm = Answer (Traditional programming)
  • Data + Answers = Algorithm (Machine learning)

• The model’s knowledge is encoded in weights, activation functions, and a graph of interconnected perceptrons
  o Explainability of answers often difficult to determine
  o Hard for humans to predict what a model will do for any given input

• Today’s AI models are very brittle
  o Great for data similar to what was in the training set
  o Often very bad with other data or problems
Traditional endpoint software attacks

• Malware exploits software vulnerabilities to gain control of an endpoint or steal data

• Maliciously-crafted macros, images, fonts, etc. can exploit these vulnerabilities in common libraries and applications

• Maliciously-crafted data applied to an AI model could exploit vulnerabilities in Machine Learning (ML) libraries
  o Easier to do if attacker has access to the model
  o Much more difficult if model is a black box

• ML libraries not commonly installed, yet
• AI techniques can protect from these attacks
Polluted training data

- AI models learn whatever is in their training data
- Malicious modification of the training data will teach the model something else
- Humans are unlikely to realize that the model has been hacked
  - Weights and activation functions are too obscure
  - Need to detect the hack in the training data set
- More common is unintentionally polluted training data

Landing Ship Tank

Jet Ski
Polluted training data – adversarial inputs

- “Crowd sourcing” for labeling data is efficient
- Need to beware of malicious labeling
- May need a machine learning model to identify fraudulent labels
Out-of-sample adversarial data input

• Exposing a model to data that was not in the original training data can yield malicious results

• Easier to do when the model is a “whitebox,” but “blackbox” attacks have been demonstrated

• Threat is much higher when the AI model’s conclusions are acted on autonomously without a human in the loop
Consumer IoT example

- Home security camera uses AI models to detect unexpected people in the house
- Learns to recognize family members and other authorized visitors
- Fewer false alerts than motion detection
- Likely can be fooled if the burglar knows this device is present and has images of an authorized person
Only SparkPredict™ detected this failure

- Existing statistical and advanced pattern recognition tools did not detect this first-of-a-kind issue
- SparkPredict detected anomalous behavior before the turbine failed
- Model indicated temperature and vibration in this subsystem were different than had been seen before
- Fixing this before failure saved millions of dollars
Possible industrial IoT scenario

• Imagine AI models monitoring streaming sensor data from a power plant

• The system is designed to automatically shutdown equipment that the model predicts are about to fail

• Attacker could shutdown the power plant by altering the sensor data without breaching the software system

• On the other hand, AI models could detect software breaches like Stuxnet
Back door neurons

Embedded in benign models

• Models can maliciously include back door neurons that will trigger for pre-defined inputs

• Much harder to detect than back doors in traditional software

• Like a virus, can be transmitted to another model that is created via transfer learning

Avoiding attack vectors for AI models

Safe software programming practices are still required to avoid traditional vulnerabilities

Additional focus on training data is required

- Verify the source and bias of your training data and keep it secure
- "Fuzz" the training data to insure robustness to out-of-sample data
- Exhaustively train on data that will inform autonomous actions

Understand the provenance of any 3rd party model that you employ