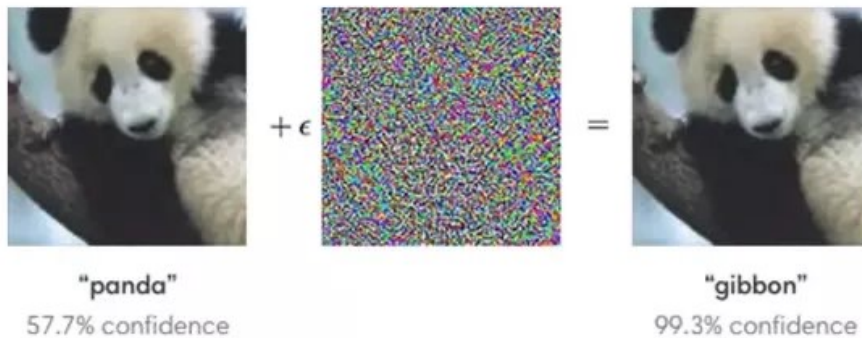


# Security of Intelligent Systems

Dr. Lotfi ben Othmane  
University of North Texas

# Commonly known Attack - Data Perturbation



# Practical Example 1: Attack of Face Recognition

A group of tax scammers hacked a government-run facial recognition system to fake tax invoices and make millions of yuan in the process, according to a report by the Xinhua Daily Telegraph.

## Chinese government-run facial recognition system hacked by tax fraudsters: report

- A group of tax scammers hacked a government-run identity verification system to fake tax invoices
- The fake tax invoices from the criminal group were valued at US\$76.2 million

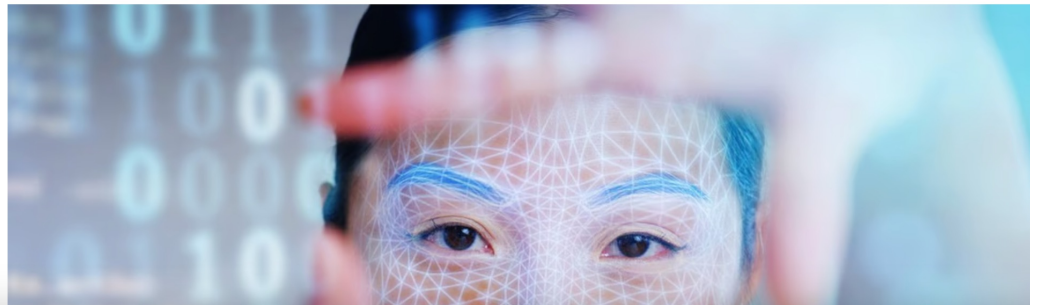


Masha Borak

+ FOLLOW

Published: 7:00am, 31 Mar, 2021

Why you can trust SCMP



<https://www.scmp.com/tech/tech-trends/article/3127645/chinese-government-run-facial-recognition-system-hacked-tax>

# Adversarial Machine Learning

Adversarial ML is subverting machine learning systems.

Adversaries **manipulate** **AI systems** to **change** their **behavior** and serve a **malicious end goal**--Do what is not intended to do.

- This happens by exploiting vulnerabilities in the design and input of the AI system.

# Importance of Adversarial learning

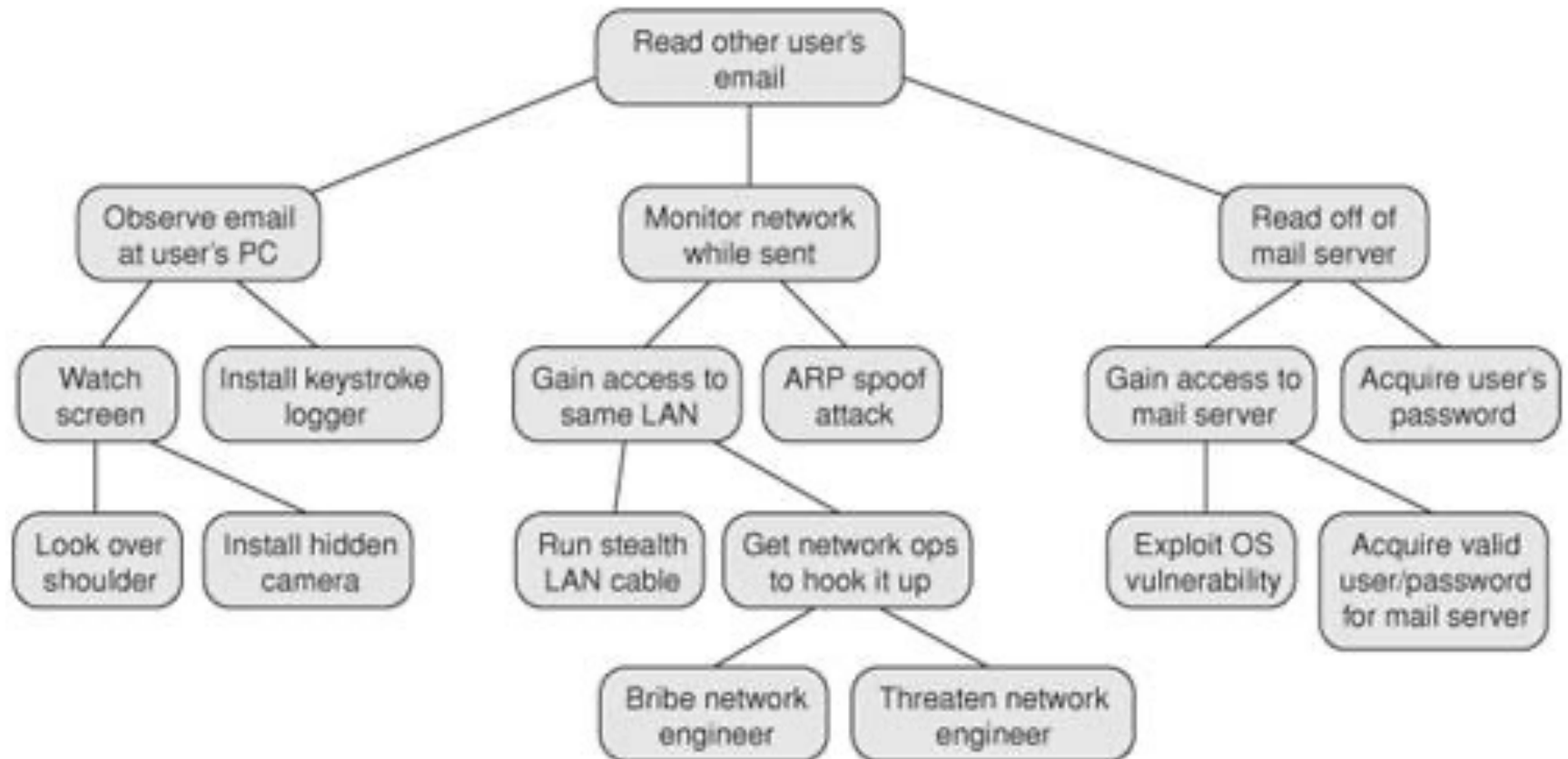
- Critical in the case of autonomous agents that use decision-making in security-critical contexts, such as military or healthcare applications.
- The systems shall be designed to ensure that they cannot be hacked or manipulated to make decisions that could have severe consequences.



<https://innovationatwork.ieee.org/how-can-autonomous-vehicles-be-protected-against-cyber-security-threats/>

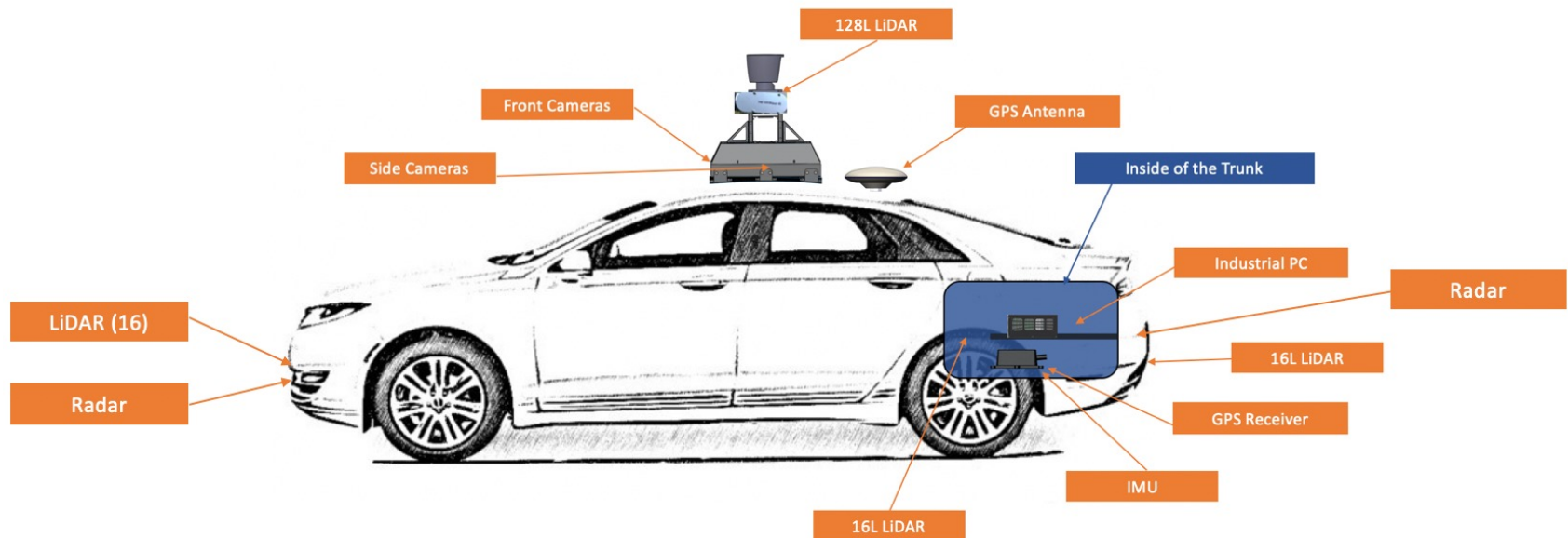
# System Assessment - Threat Modeling

Threat modeling is about identifying potential threats to a given system.

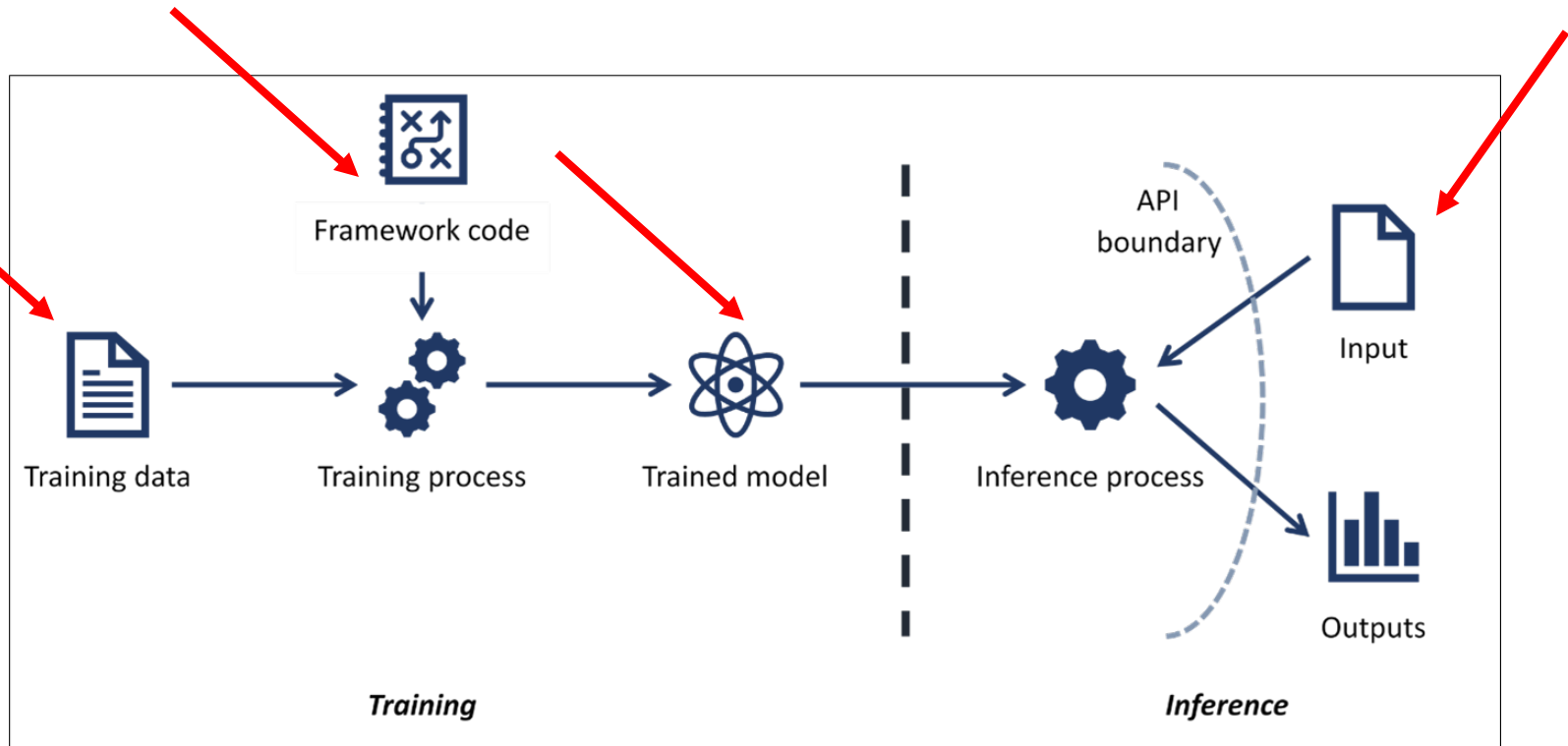


# System Assessment - Attack Surfaces

Attack surface is the set of points on the boundary of a system where an attacker can try to enter, cause an effect on, or extract data from, that system.



# AI Attack Surface

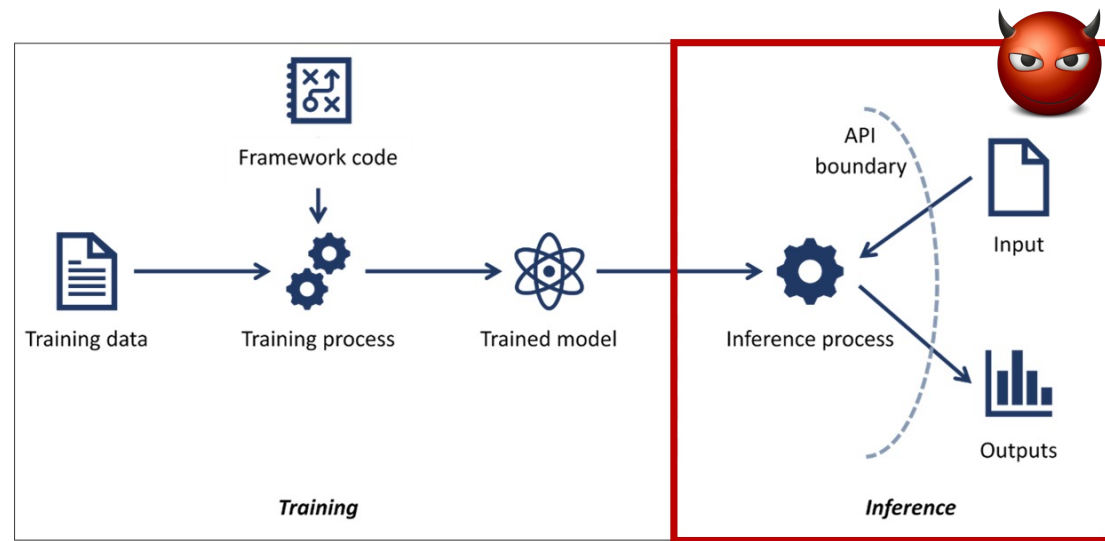


<https://github.com/mitre/advmthreatmatrix/>



# Attack Scenario #1: Inference Attack

- The model is deployed as an API.
- The attacker can only query the model and observe the response.
- The attacker controls the input to the model, but the attacker does not know how it is processed.



<https://github.com/mitre/advm1threatmatrix/>

# Scenario 1 - Model Evasion Attack

The goal of an evasion attack is to avoid detection by security mechanisms and allow the attacker to carry out their intended objectives.

## ProofPoint Evasion Exercise

Incident Date: **September 9, 2019**

Actor: **Researchers at Silent Break Security** | Target: **ProofPoint Email Protection System**

 **DOWNLOAD DATA** ▾

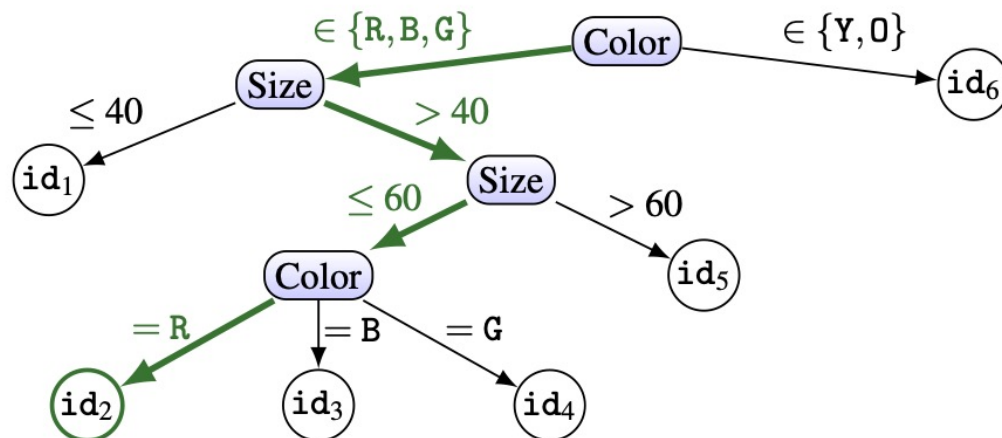
### Summary

Proof Pudding (CVE-2019-20634) is a code repository that describes how ML researchers evaded ProofPoint's email protection system by first building a copy-cat email protection ML model, and using the insights to bypass the live system. More specifically, the insights allowed researchers to craft malicious emails that received preferable scores, going undetected by the system. Each word in an email is scored numerically based on multiple variables and if the overall score of the email is too low, ProofPoint will output an error, labeling it as SPAM.

# Scenario 1 - Model Inversion Attack

Attacker recovers the features used to train the model.

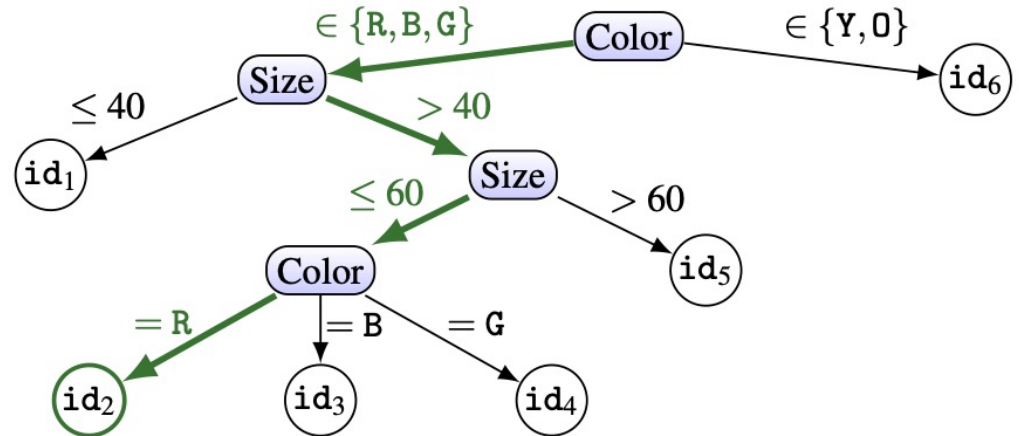
E.g., Find the credit ML



<https://www.cs.cmu.edu/~Emfredrik/papers/fjr2015ccs.pdf>

# Scenario 1 - Extraction Attack

- These attacks are performed by iteratively querying a model and observing the output.
- E.g., Find the credit ML




<https://www.cs.cmu.edu/%7Emfredrik/papers/fjr2015ccs.pdf>

# Scenario 1 – Model Extraction/Replication Attack

## GPT-2 Model Replication Exercise

Incident Date: **August 22, 2019**

Actor: **Researchers at Brown University** | Target: **OpenAI GPT-2**

 **DOWNLOAD DATA** ▾

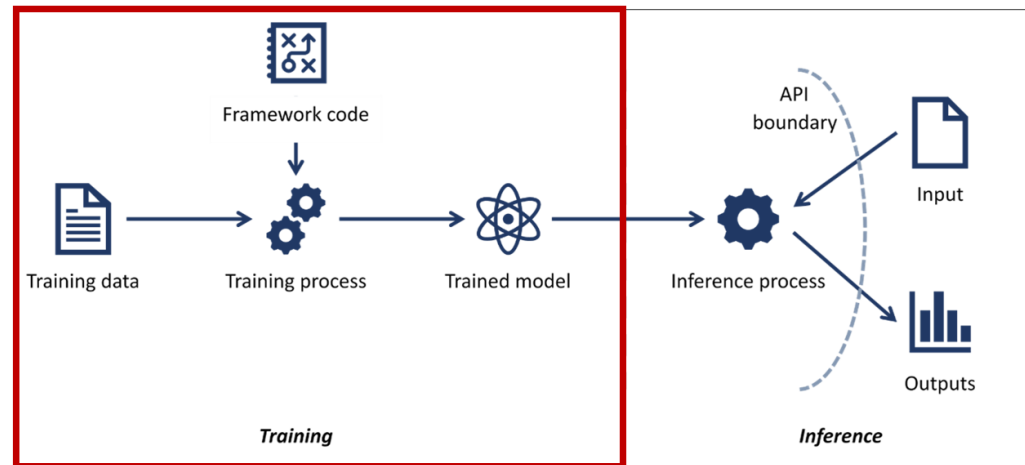
### Summary

OpenAI built GPT-2, a language model capable of generating high quality text samples. Over concerns that GPT-2 could be used for malicious purposes such as impersonating others, or generating misleading news articles, fake social media content, or spam, OpenAI adopted a tiered release schedule. They initially released a smaller, less powerful version of GPT-2 along with a technical description of the approach, but held back the full trained model.

Before the full model was released by OpenAI, researchers at Brown University successfully replicated the model using information released by OpenAI and open source ML artifacts. This demonstrates that a bad actor with sufficient technical skill and compute resources could have replicated GPT-2 and used it for harmful goals before the AI Security community is prepared.

# Attack Scenario #2: Training Time Attack

- The attacker has control over training data.
- This flavor of attack is shown in Tay poisoning case study where the attacker was able to compromise the training data via the feedback mechanism.



<https://github.com/mitre/advmlthreatmatrix/>

# Scenario 2 - Model Poisoning Attack

Attacker contaminates the training data of an ML system in order to get a desired outcome at inference time

## Tay Poisoning

Incident Date: **March 23, 2016** | Reporter: **Microsoft**  
Actor: **4chan Users** | Target: **Microsoft's Tay AI Chatbot**

 **DOWNLOAD DATA** ▾

### Summary

Microsoft created Tay, a Twitter chatbot designed to engage and entertain users. While previous chatbots used pre-programmed scripts to respond to prompts, Tay's machine learning capabilities allowed it to be directly influenced by its conversations.

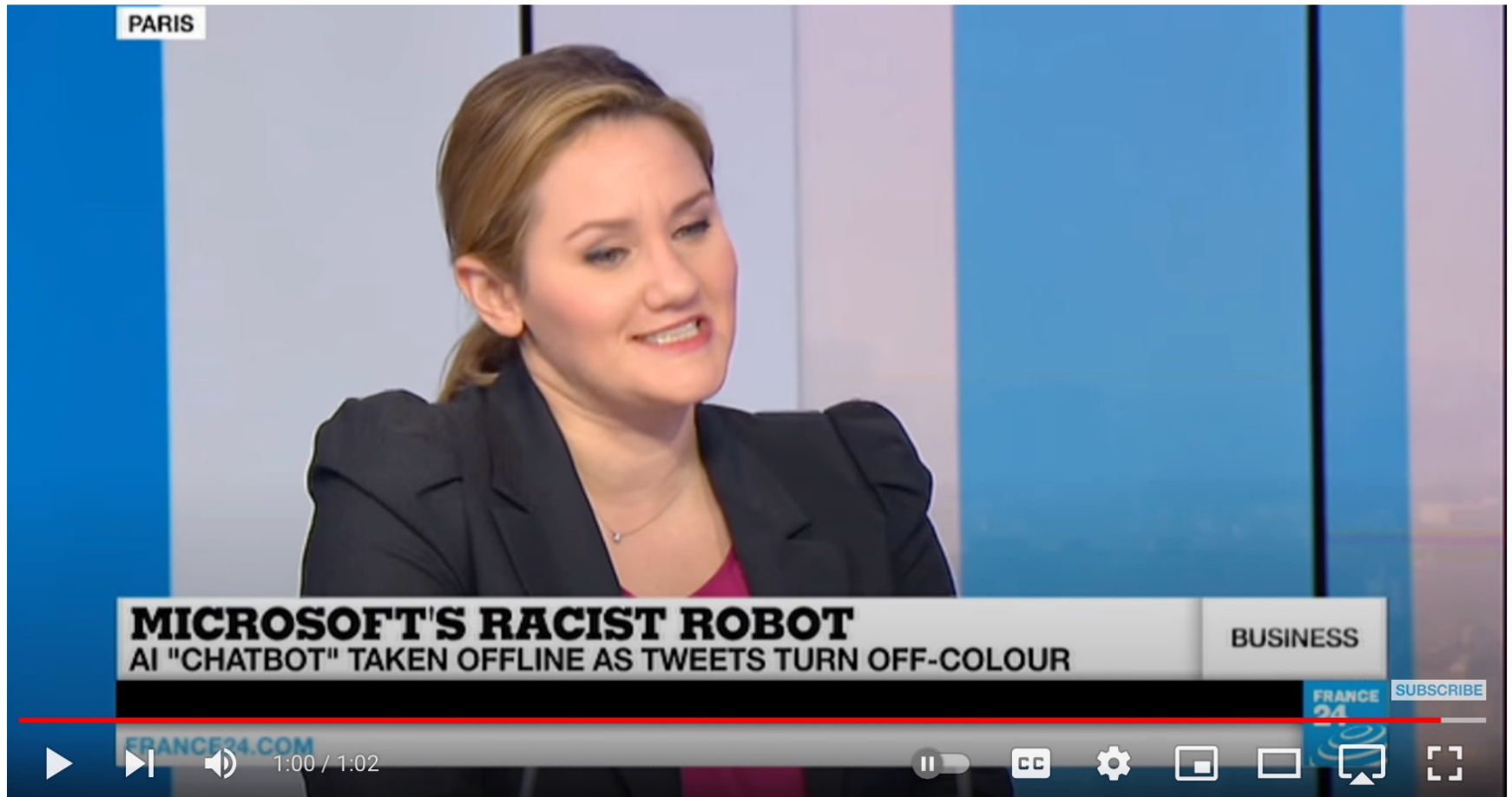
A coordinated attack encouraged malicious users to tweet abusive and offensive language at Tay, which eventually led to Tay generating similarly inflammatory content towards other users.

Microsoft decommissioned Tay within 24 hours of its launch and issued a public apology with lessons learned from the bot's failure.

# Practical Example 2: Tay Poisoning



Search

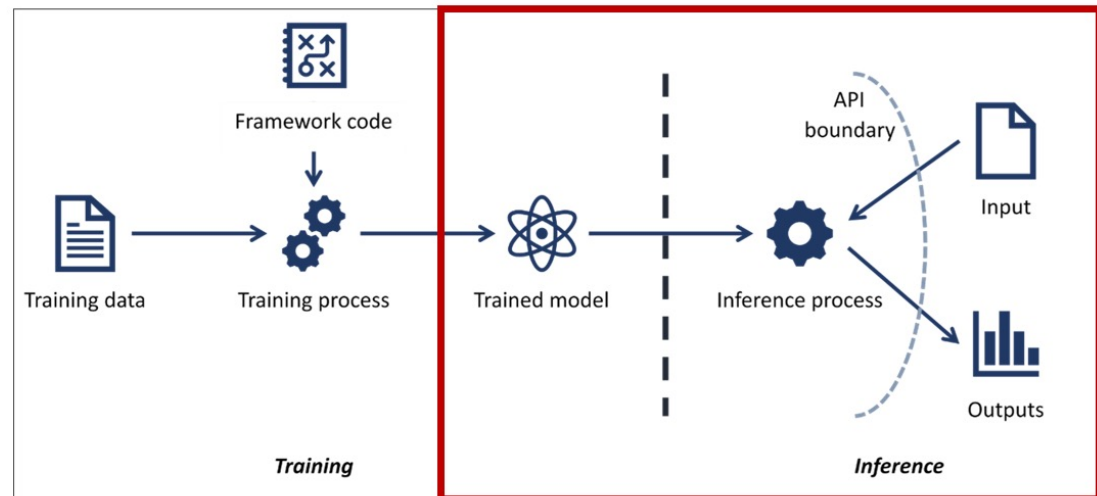


<https://www.youtube.com/watch?v=jTSn7f4sEko>



# Attack Scenario #3: Attack on Edge/Client

- The model exists on a client (like a phone) or on the edge (such as IoT) .
- An attacker might have access to model code through reversing the service on the client.



<https://github.com/mitre/advmlthreatmatrix/>

# Traditional Attacks

The attacks use software and network vulnerabilities to get access to the system and then manipulate the ML models.



The winning team got to keep the Model 3 it hacked. (Tesla)

# Adversarial ML Threat Matrix

MITRE framework for AI attacks.

1. Reconnaissance
2. Initial access
3. Execution
4. Persistence
5. Model Evasion
6. Exfiltration
7. Impact

Reconnaissance	Initial Access	Execution	Persistence	Model Evasion	Exfiltration	Impact	
Acquire OSINT information: (Sub Techniques) 1. Arxiv 2. Public blogs 3. Press Releases 4. Conference Proceedings 5. Github Repository 6. Tweets	Pre-trained ML model with backdoor	Execute unsafe ML models (Sub Techniques) 1. ML models from compromised sources 2. Pickle embedding	Execute unsafe ML models (Sub Techniques) 1. ML models from compromised sources 2. Pickle embedding	Evasion Attack (Sub Techniques) 1. Offline Evasion 2. Online Evasion	Exfiltrate Training Data (Sub Techniques) 1. Membership inference attack 2. Model inversion	Defacement	
ML Model Discovery (Sub Techniques) 1. Reveal ML model ontology – 2. Reveal ML model family –	Valid account	Execution via API	Account Manipulation		Model Stealing	Denial of Service	
Gathering datasets	Phishing	Traditional Software attacks	Implant Container Image	Model Poisoning	Insecure Storage 1. Model File 2. Training data	Stolen Intellectual Property	
Exploit physical environment	External remote services			Data Poisoning (Sub Techniques) 1. Tainting data from acquisition – Label corruption 2. Tainting data from open source supply chains 3. Tainting data from acquisition – Chaff data 4. Tainting data in training environment – Label corruption		Data Encrypted for Impact Defacement	
Model Replication (Sub Techniques) 1. Exploit API – Shadow Model 2. Alter publicly available, pre-trained weights	Exploit public facing application					Stop System Shutdown/Reboot	
Model Stealing	Trusted Relationship						

<https://github.com/mitre/advmthreatmatrix>

# Use of the ML Threat Matrix – Example 1

## Camera Hijack Attack on Facial Recognition System

- The attackers bought customized low-end mobile phones, customized android ROMs, a specific virtual camera application, identity information and face photos.
- The attackers used software to turn static photos into videos, adding realistic effects such as blinking eyes. Then, the attackers use the purchased low-end mobile phone to import the generated video into the virtual camera app.
- The attackers registered an account with the victim's identity information. In the verification phase, the face recognition system called the camera API, but because the system was hooked or rooted, the video stream given to the face recognition system was actually provided by the virtual camera app.
- The attackers successfully **evaded** the face recognition system and impersonated the victim.

# Defense Mechanisms

- Data validation and filtering - Validate and filter the data used to train the AI model
- Model robustness - Design models more resistant to adversarial attacks by e.g., adding noise to the input data or using defensive distillation techniques.
- Use explainable AI - Use transparent and understandable AI models. This can help to identify and correct any potential biases or errors in the AI system.

# To Learn More

- <https://github.com/mitre/advmlthreatmatrix>
- <https://learn.microsoft.com/en-us/security/engineering/failure-modes-in-machine-learning>
- <https://dcai.csail.mit.edu/lectures/data-privacy-security/>
- <https://adversarial-ml-tutorial.org/introduction/>

# Conclusions

- Adversaries manipulate AI systems to change their behavior and serve a malicious end goal.
- Adversarial ML are important for autonomous agents that use ML for decision making
- Adversarial attacks include evasion, extraction/replication, inversion, and poisoning attacks.
- The main mechanisms to mitigate these attacks are model robustness and explainable AI

Thank you

Any Question?