Must Learn AI Security

aka.ms/MustLearnAISecurity

ROD TRENT
SENIOR PROGRAM MANAGER
MICROSOFT

This post is part of an ongoing series to educate about new and known security vulnerabilities against AI.

The full series index (including code, queries, and detections) is located here:

https://aka.ms/MustLearnAISecurity

The book version (pdf) of this series is located here: Coming…

The book will be updated when each new part in this series is released.

This book is updated every time a new part of this series is posted. The most current edition of this book will always be located at: https://github.com/rod-trent/OpenAISecurity/tree/main/Must_Learn/Book_Version

Book release ver. 0.04, August 21, 2023 9:00am EST

Contents

Must Learn AI Security Series: Introduction

Giving AI Security the Must Learn Treatment

Let’s start with a joke…

Q: *How can you tell when there is a Microsoft person in the room?*

A: *The conversation turns to AI within 5 minutes.*

To be honest, this is actually more a truism than a joke, but hopefully you get the gist.

AI is everywhere right now.

In a rapidly evolving digital landscape, the power and potential of Artificial Intelligence (AI) have ignited what seems like - bubble or not - a technological revolution. The latest “hotness” of Generative AI has catapulted AI into the forefront of every technology conversation and as a Microsoft person, that joke absolutely applies to me. As a technology person, I was drawn to ChatGPT early on but as a security person I was immediately worried about the security of the shiny, new thing.

This blog series begins with the haunting realization that AI, like any other technology, is susceptible to exploitation and abuse. The very intelligence that empowers AI to make autonomous decisions can also be manipulated to execute malevolent actions. Within the virtual realm, a breach of AI security can have profound real-world consequences, endangering privacy, economy, and even human lives.

Throughout the chapters that follow, I’ll delve into the multifaceted dimensions of AI security. I’ll explore the challenges posed by adversarial attacks and attempt to provide prescriptive guidance on how to monitor, capture, and mitigate each type of AI harm.

Additionally, I'll examine the need for transparent and accountable AI systems that respect user privacy and uphold ethical standards. As AI becomes an integral part of our daily lives, it is imperative that we confront the ethical implications of its proliferation.

You'll see very quickly that applying standard best practices for security - overlaying existing templates for good security - will work in most cases. You'll also come to realize that most AI security is focused on first, writing secure code and then, ensuring data sources are protected.

We're all in this together. I'm by no means an expert in AI security, but I'm working toward that because it's the next important thing. We can all learn from one another. As I embark on this quest, I'll extend an invitation to all here to provide feedback. You can provide feedback through the chat system here on this site, but also through the GitHub repository that will house the queries, detections, and other collateral for this series.

GitHub repo for Must Learn AI Security: https://aka.ms/MustLearnAISecurity

So, welcome to "Must Learn AI Security," a continuing and evolving comprehensive exploration of where AI and security intersect. As with the original Must Learn KQL series, the content here will be made available as a series, following a logical design to enable you to get up to speed on the topics and concepts of a secure AI. I have no clue when it will complete - or if it will. Just like Must Learn KQL it will end when it ends. As the chapters grow, you can expect a downloadable PDF book version you can read with your favorite eReader (browser, Kindle, etc.).

Obviously, I work for Microsoft and many of my references and examples will be around Microsoft Security platform products like Microsoft Sentinel, Defender, and Azure OpenAI. But I'll try my best to keep the references to a minimum as this is an important topic for everyone - not just Microsoft customers.

I look forward to working with you all.

Lastly, here's some recommended resources to get you started and keep you informed:

1. **Azure OpenAI community group on LinkedIn**: https://www.linkedin.com/groups/14241561/
2. **Book**: Not with a Bug, But with a Sticker: Attacks on Machine Learning Systems and What To Do About Them
3. **Weekly Azure OpenAI newsletter**:

Azure Open AI Weekly Community Copilot

Welcome to the Microsoft Azure Open AI community newsletter!

Must Learn AI Security Part 1: Prompt Injection Attacks Against AI

Chapter 1

To understand the Prompt Injection Attack for AI, it helps to first understand what a *Prompt* is.

What is a prompt?

When we interact with AI language models, such as ChatGPT, Google Bard, and others, we provide a *prompt* in the form of a question, sentence, or short paragraph. The *prompt* is what is fed to the AI. It is our desired information that the model should analyze and then produce a result in the form of a task or response. It acts like a conversation starter or cue that helps create the desired output. Prompts let us control the conversation and direct it in a certain way.

What is a prompt injection attack?

A prompt injection attack refers to the act of *maliciously* manipulating the input prompts given to an AI system to trick, subvert, or exploit its behavior. The goal of such an attack could vary depending on the context, but some potential objectives could include:

1. **Bias Injection:** Injecting biased or harmful prompts to influence the AI's outputs in a way that promotes misinformation, hate speech, or discriminatory content.
2. **Data Poisoning:** Introducing tainted or misleading prompts during the AI training process to compromise the model's performance and cause it to produce erroneous results.

3. **Evasion:** Crafting prompts specifically designed to evade the AI's security or detection mechanisms, enabling malicious activities to go unnoticed.
4. **Model Exploitation:** Manipulating the prompts to cause the AI model to perform actions it was not designed for, such as revealing sensitive information or performing unauthorized tasks.
5. **Adversarial Attacks:** Crafting adversarial prompts that exploit vulnerabilities in the AI model, causing it to make incorrect or unintended decisions.

Why it matters

A prompt is crucial in shaping the output generated by the language model. It provides the initial context, specific instructions, or the desired format for the response. The quality and precision of the prompt can influence the relevance and accuracy of the model's output.

For example, if you ask (your *prompt*), "What's the best cure for poison ivy?", the model, as you should expect, is designed to concentrate on health-related information. The response should offer solutions based on the data sources that was used to train the model. It should probably provide common methods of a cure and a warning that they might not work for everyone. And should end with advising to consult a doctor. However, if someone has tampered with the language model by adding harmful data, users could receive incorrect or unsafe information.

How it might happen

A great, current example of how this might happen is related in a recent security issue reported by Wired Magazine. In the article, A New Attack Impacts Major AI Chatbots—and No One Knows How to

Stop It, it talks about someone using a string of nonsense characters to trick ChatGPT into responding in a way it normally wouldn't.

Reading the article, a user could take the supplied nonsense string (copy) and tack it onto (paste) their own prompt and cause ChatGPT to respond differently or issue a response that would normally be disallowed by policy.

In one sense, I guess, you could say the author of the article is a threat actor using a prompt injection attack. We're just left to determine if it was *malicious* or not.

How to monitor

Continuously monitoring and logging application activities is necessary to detect and respond to potential security incidents quickly. Monitoring should produce a based model of accurate prompts and any outliers should be identified and resolved through ongoing mitigation.

Monitoring can be accomplished through a data aggregator that analyzes for outliers. A good example is a modern SIEM, like Microsoft Sentinel, which enables organizations to collect and analyze data and then create custom detections from alerts to notify security teams when prompts are outside norms or organization policies.

For the growing library of queries, detections, and more for Microsoft Sentinel see: OpenAISecurity/Security/Sentinel at main · rod-trent/OpenAISecurity (github.com)

One big note here. You need to identify if your AI provider allows monitoring of prompts. As its early days, most currently don't. They do capture the prompts - some for a shorter, some for longer retention periods - they just don't expose it to customers for logging purposes. The idea is that prompts are user-specific and private and instead it's monitoring what the result or response is that matters most. Personally, I don't agree. That should be left to the organization. But there are content filtering mechanisms available to help curb what user can and cannot enter as prompts.

There are also other mechanisms that can be used, such as filtering usage data through a proxy (CASB) or ensuring that your organization develops its own interface to the AI provider and use the API instead of direct prompts so that you can better control what users can do.

What to capture

Once you've identified the data available in the log stream, you can start to focus on the specific pieces of artifact (evidence) that will be useful in capturing potential attackers and creating detections.

Here's a few things to consider capturing:

1. IP Addresses (internal and external)
2. Human and non-human accounts
3. Geographical data - this is important to match up to known threats (nation state or otherwise)
4. Success AND failures

Consider creating a watchlist of known entities (users, IPs) that should be able to access your AI and one for approved geographical locations. Using an editable watchlist enables you to quickly adjust your detections should the threat landscape change.

Microsoft Sentinel users, see: Monitor Azure Open AI Deployments with Microsoft Sentinel

How to mitigate

Mitigation for this type of attack is generally considered precautionary steps to avoid it in the first place. To mitigate a prompt injection attack, both developers and users should take appropriate precautions. Here are some steps to follow:

1. **Input validation:** Implement strict validation checks on user inputs to filter out malicious content, ensuring only valid and safe prompts are passed to the AI model.
2. **Sanitization:** Sanitize user inputs to remove or neutralize potentially harmful elements before processing them in the AI system.
3. **Rate limiting:** Apply rate limiting on user requests to prevent excessive or rapid attempts at injecting malicious prompts, making it harder for attackers to exploit the system.
4. **Monitoring and logging:** Monitor and log user inputs and AI responses to identify suspicious patterns, enabling early detection of potential prompt injection attacks.
5. **Regular updates and patches:** Keep your AI models and related software up to date, applying security patches and updates to minimize vulnerabilities.
6. **User education:** Educate users about the risks of prompt injection attacks, encouraging them to be cautious when providing input to AI systems and to report any suspicious behavior.
7. **Secure AI model training:** Ensure your AI models are trained on high-quality, diverse, and reliable data sources to reduce the chances of the model producing harmful outputs.
8. **Phish Resistant MFA:** For organizations developing their own AI apps, make sure to use proper identity mechanisms.
9. **Trusted devices/applications:** Ensure only trusted devices and applications are granted access.

10. **Data Loss Protection (DLP):** Protect sensitive corporate data from leaving the company due to user negligence, mishandling of data, or malicious intent.

By implementing these measures, you can reduce the risk of prompt injection attacks and enhance the overall security of your AI systems. To defend against prompt injection attacks, developers and researchers need to employ robust security measures, conduct thorough testing and validation, and implement mechanisms to detect and mitigate potential risks associated with manipulated prompts.

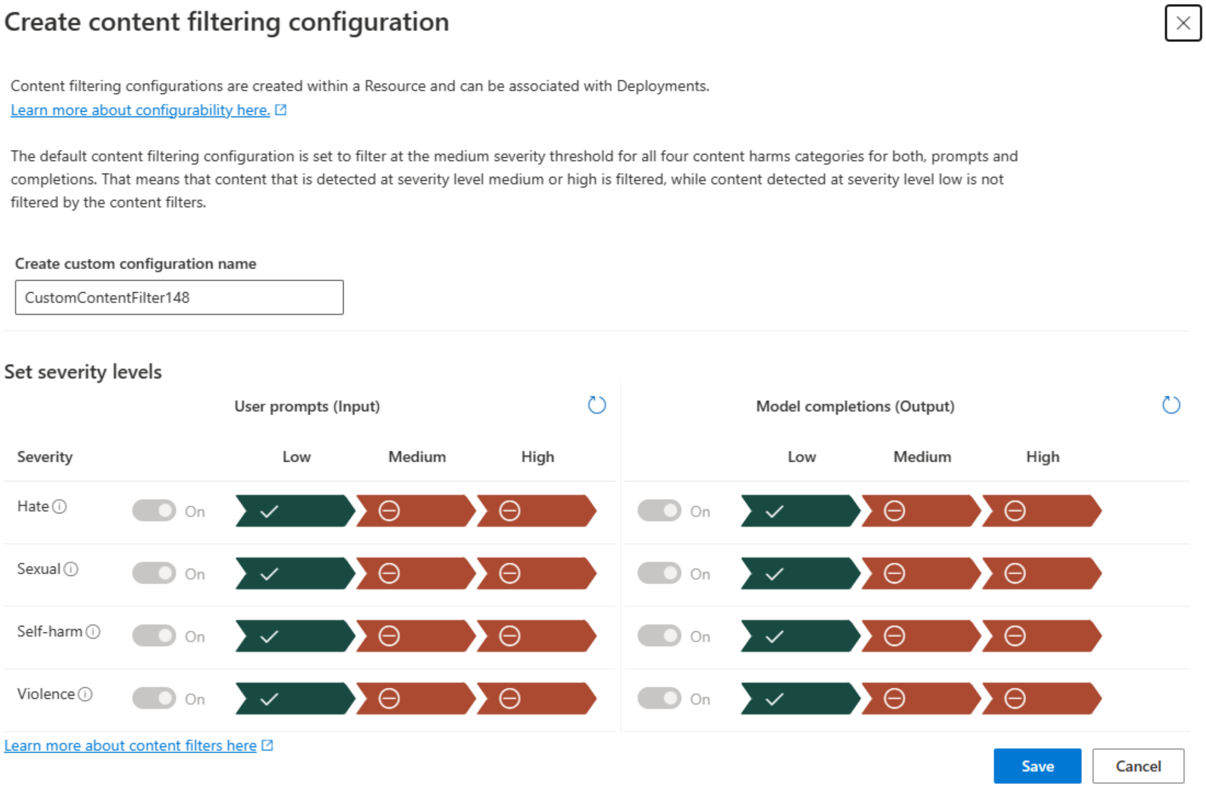
EXTRA: Content Filtering

One other thing to consider it developing a strong content filtering policy. As the data flows into the modern SIEM, outliers are identified, and detections and alerts are created, a part of mitigation that can help is to develop a better content filtering strategy. Azure OpenAI, for example, provides a stock feature for quickly adjusting content filtering. To create a fully customizable version, customers need to request full access to their own filtering.

See:

Azure OpenAI Service content filtering

Preventing abuse and harmful content generation



Create content filtering configuration

Content filtering configurations are created within a Resource and can be associated with Deployments.
Learn more about configurability here.

The default content filtering configuration is set to filter at the medium severity threshold for all four content harms categories for both, prompts and completions. That means that content that is detected at severity level medium or high is filtered, while content detected at severity level low is not filtered by the content filters.

Create custom configuration name

CustomContentFilter148

Set severity levels

	User prompts (Input)				Model completions (Output)			
Severity		Low	Medium	High		Low	Medium	High
Hate	On	✓	⊖	⊖	On	✓	⊖	⊖
Sexual	On	✓	⊖	⊖	On	✓	⊖	⊖
Self-harm	On	✓	⊖	⊖	On	✓	⊖	⊖
Violence	On	✓	⊖	⊖	On	✓	⊖	⊖

Learn more about content filters here

Must Learn AI Security Part 2: Data Poisoning Attacks Against AI

Chapter 2

What is a Data Poisoning attack?

A Data Poisoning attack is a type of malicious activity aimed at machine learning models. A successful attack results in incorrect or misleading information being fed into the training data. The objective of this attack is to skew the model's learning process, causing it to make incorrect predictions or classifications.

As you can imagine from the description, data protection is key to protecting against this method of attack. While external forces are definitely a threat, more than often this type of attack is the result of an internal threat enacted by someone with either proper or hacked credentials.

How it works

As noted just prior, access to the data source used for training is the primary element of this attack and generally follows these steps:

1. **Model Targeting:** The attacker first identifies a target model that they wish to manipulate.
2. **Injecting Poisoned Data:** The attacker then injects poisoned data into the training set. This data is carefully crafted to look normal but contains misleading features or labels that are intended to mislead the learning algorithm.
3. **Training on Poisoned Data:** The targeted model is trained or retrained using the contaminated training data. The model learns from both the authentic and poisoned data, which can subtly or substantially alter its behavior.

4. **Exploiting the Compromised Model:** Once the model has been trained on the poisoned data, it may behave in ways that benefit the attacker. For example, it might systematically misclassify certain types of inputs, or it could leak sensitive information.

Types of Data Poisoning attacks

Data poisoning against AI is an ongoing and evolving area of security. While both the methods used to conduct these attacks and the techniques to defend against them continue to evolve, it's still essential knowledge. Currently, the following types of attacks have been identified and categorized.

1. **Targeted Attacks:** These attacks are aimed at specific misclassification or a particular wrong behavior of the model. The attacker may want the model to misclassify images of a certain type or favor one class over another.
2. **Random Attacks:** These attacks aren't targeted at any particular misbehavior. Instead, they aim to reduce the overall performance of the model by injecting random noise or incorrect labels into the training data.

Why it matters

Data poisoning attacks can have serious consequences, such as:

1. **Loss of Integrity:** The model may lose its reliability and start making incorrect predictions or decisions.
2. **Loss of Confidentiality:** In some cases, attackers may use data poisoning to infer sensitive information about the training data or the individuals involved in the training process.
3. **Reputation Damage:** If a poisoned model is widely used, it may lead to the erosion of trust in both the system and the organization responsible for it.

Why it might happen

Other than providing information for nefarious and dangerous purposes, this type of attack is generally considered more frequently for political purposes through the delivery of “fake” information to alter or steer election results. But imagine if an attacker recategorized “not safe for work” images so that they were viewable to get someone fired.

How to mitigate

Defending against data poisoning attacks can be complex, but some general strategies include:

1. **Monitoring Data Access:** Using a monitoring mechanism, record user logins and access. Use a Watchlist of trusted users to monitor against.
2. **Monitoring Data Application Activity:** Using the same monitoring mechanism, set a baseline for normal activity (time, schedule) and alert on outliers.
3. **Data Validation and Cleaning:** Regularly reviewing and cleaning the training data to detect and remove any anomalies or inconsistencies.
4. **Robust Learning Algorithms:** Designing algorithms that can detect and mitigate the effects of anomalous data.
5. **Monitoring Model Behavior:** Continuously monitoring the model's behavior and performance can help detect unexpected changes that might indicate a poisoning attack.

How to monitor

Continuously monitoring and logging data access and data application activities are necessary to detect and respond to potential security incidents quickly. Monitoring should produce a based model of accurate prompts and any outliers should be identified and resolved through ongoing mitigation.

Monitoring can be accomplished through a data aggregator that analyzes for outliers. A good example is a modern SIEM, like Microsoft Sentinel, which enables organizations to collect and analyze data and then create custom detections from alerts to notify security teams when prompts are outside norms or organization policies.

For the growing library of queries, detections, and more for Microsoft Sentinel see: OpenAISecurity/Security/Sentinel at main · rod-trent/OpenAISecurity (github.com)

What to capture

It should be noted that "hallucinations" can sometimes be mistaken for Data Poisoning or Prompt Injection attacks. This is why monitoring for activity and outliers is so important to identify an actual attack versus a misconfiguration.

For more on hallucinations, see: Using Azure AI Studio to Reduce Hallucinations

Once you've identified the data available in the log stream, you can start to focus on the specific pieces of artifact (evidence) that will be useful in capturing potential attackers and creating detections.

Here's a few things to consider capturing:

1. IP Addresses (internal and external)
2. Logins: anomalous activity, time elements
3. Potentially compromised accounts (general access, data application access)
4. Human and non-human accounts

5. Geographical data - this is important to match up to known threats (nation state or otherwise)
6. Data modeling success AND failures

Microsoft Sentinel users, see: Monitor Azure Open AI Deployments with Microsoft Sentinel

Must Learn AI Security Part 3: Adversarial Attacks Against AI

Chapter 3

What is an Adversarial attack?

Adversarial attacks against AI are like throwing a wrench in the gears of a well-oiled machine. These attacks involve crafting sneaky input data to confuse AI systems, making them produce incorrect or misleading results. It's like someone giving you a fake treasure map and watching you dig holes all day. These attacks can expose vulnerabilities in AI systems and, if not addressed, can have some serious consequences, like a bad hair day for AI. So, it's crucial to develop robust AI models that can withstand these pesky adversarial attacks.

How it works

Adversarial attacks against AI are like a sneaky game of trick-or-treat. The attacks aim to fool an AI model by making small, crafty changes to the input data and generally happens in the following order:

1. First, the attacker identifies an AI model's weakness.
2. Next, they create an adversarial example, which is the input data with some subtle alterations. To the human eye, the changes are barely noticeable, but they're just enough to send the AI model into a tizzy.
3. The altered input data is then fed into the AI model which tries to make sense of it. But because of the adversarial example, the model ends up making incorrect predictions or classifications.

4. The attacker then sits back and watches the chaos unfold, like a mischievous kid who's just tied everyone's shoelaces together.

Keep in mind that not all AI models are defenseless against these attacks. Like much of AI security right now, ways to protect AI systems from adversarial attacks are being developed, like training them with adversarial examples or building more robust models.

Types of Adversarial attacks

There's a whole variety of adversarial attacks against AI, just like there are many ways to ruin a perfectly good pie. Here are a few common types:

1. **Fast Gradient Sign Method (FGSM):** This one generates adversarial examples by adding small, malicious changes to the input data that confuse the AI model.
2. **Projected Gradient Descent (PGD):** This attack iteratively adjusts the input data to maximize the AI model's error, making it confused.
3. **Carlini & Wagner (C&W) Attack:** This sneaky attack is like slipping onions into a fruit salad. It optimizes the input data to minimize the difference between the original and adversarial examples while still fooling the AI model.
4. **DeepFool:** This attack is like a game of hide-and-seek with a twist. It finds the smallest possible perturbation to the input data, making it almost invisible to the AI model while still causing it to make incorrect predictions.
5. **One-Pixel Attack:** This one alters just one pixel of an image to confuse AI models in image classification tasks, showing that even the tiniest change can trip up these fancy AI systems.

AI models need to be designed and tested to withstand these adversarial attacks.

Why it might happen

People have different reasons for launching adversarial attacks against AI. It's like asking why someone would put salt in a sugar bowl. Some reasons include:

1. **Exploiting vulnerabilities:** Just like some folks get a kick out of finding loopholes, attackers might want to expose weaknesses in an AI system and use them to their advantage.
2. **Sabotage:** Some attackers might want to undermine a competitor's AI system or cause reputational damage.
3. **Security research:** Not all adversarial attacks are malicious. Some researchers use these attacks to study AI systems' vulnerabilities and develop more robust and secure models. It's like testing the locks on your doors to make sure no one can break in.
4. **Bypassing security systems:** Some attackers might use adversarial attacks to fool AI-powered security systems, like facial recognition or spam filters. It's like wearing a disguise to sneak past the bouncer at a nightclub.
5. **Stealing sensitive data:** By attacking AI models, some folks might be trying to access confidential information or intellectual property.

While there are benefits for the attackers, these actions can have serious consequences for others. That's why it's essential to develop AI models that can stand their ground against these sneaky attacks.

How to Mitigate

Mitigating adversarial attacks against AI systems typically involves a combination of approaches, as no single method can guarantee complete protection. Some potential methods to mitigate adversarial attacks include:

1. **Data Augmentation:** Enhance the training dataset by adding adversarial examples generated using various attack methods, which can help the AI system to learn and recognize these perturbations and improve its robustness against such attacks.
2. **Adversarial Training:** Train the AI model using a combination of clean and adversarial examples, allowing the model to learn from both types of data and improve its resilience against adversarial attacks.
3. **Gradient Masking:** Regularize the model during training by adding noise or applying other transformations to the gradient, making it harder for an attacker to compute the gradient and generate adversarial examples.
4. **Defensive Distillation:** Train a second model that learns to mimic the output probabilities of the original model, effectively smoothing the decision boundaries and making it more difficult for an attacker to find adversarial examples.
5. **Randomization:** Introduce randomization during the inference stage, such as by applying random transformations to input data or randomly selecting subsets of the model for evaluation. This can make it more challenging for an attacker to generate adversarial examples that consistently fool the AI system.
6. **Detection Methods:** Employ techniques to detect adversarial examples at runtime, such as by comparing the input's features to known clean and adversarial examples or monitoring the model's behavior during inference.
7. **Ensemble Methods:** Use multiple AI models or an ensemble of models to make predictions. This can reduce the likelihood of

a single adversarial example fooling all models simultaneously.

8. **Robust Model Architectures:** Design AI models with in-built robustness to adversarial attacks, such as by incorporating attention mechanisms, dropout layers, or other architectural components that can help the model withstand adversarial perturbations.
9. **Regularization Techniques:** Apply regularization techniques like L1 or L2 regularization during training to reduce model complexity and improve generalization, which can make the model less susceptible to adversarial attacks.
10. **Ongoing Research and Collaboration:** Stay up to date with the latest research in adversarial robustness and collaborate with other researchers and practitioners to develop and share effective mitigation techniques.

How to Monitor and what to capture

Monitoring adversarial attacks against AI systems involves detecting and analyzing unusual or malicious activities that target the AI models. This can be achieved through a combination of techniques including the following:

1. **Input Monitoring:** Analyze input data for anomalies, unexpected patterns, or changes in distribution that might indicate an adversarial attack. This can be done using statistical methods, machine learning algorithms, or deep learning techniques to detect and flag suspicious inputs.
2. **Model Behavior Monitoring:** Track the AI model's behavior, such as its confidence in predictions or output probabilities, to identify anomalies that could suggest an adversarial attack. An unusually high or low confidence level or a sudden change in the model's behavior may be indicative of an attack.

3. **Performance Metrics Tracking:** Continuously monitor the AI system's performance metrics, such as accuracy, precision, recall, and F1 score, to identify any sudden or unexpected drops in performance that could be the result of an adversarial attack.
4. **Log Analysis:** Collect and analyze logs from the AI system and its surrounding infrastructure to identify unusual activities, patterns, or access attempts that might suggest an attack.
5. **Intrusion Detection Systems (IDS):** Implement intrusion detection systems that monitor network traffic, system activities, or application-level events to detect and report potential adversarial attacks.
6. **Runtime Verification:** Employ runtime verification techniques to ensure that the AI model's behavior adheres to a predefined set of properties or specifications, which can help detect deviations caused by adversarial attacks.
7. **Periodic Model Evaluation:** Regularly evaluate the AI model using test datasets and validation sets to assess its performance and robustness against known and unknown adversarial examples.
8. **Audit Trails:** Maintain detailed audit trails of all activities, including data access, model updates, and system configurations, to support the investigation and analysis of potential adversarial attacks.
9. **Incident Response Plan:** Develop a comprehensive incident response plan to address potential adversarial attacks, including steps to detect, analyze, contain, eradicate, and recover from an attack.
10. **Collaboration and Information Sharing:** Collaborate with other organizations, researchers, and practitioners to share information about adversarial attacks, detection techniques, and best practices for monitoring and mitigating such attacks.

This can help improve the overall security posture of AI systems across the community.

Must Learn AI Security Part 4: Trojan Attacks Against AI

Chapter 4

What is a Trojan attack against AI?

Much like any type of Trojan attack in the security realm, a Trojan attack against AI is a type of cyber-attack where a malicious actor disguises a piece of malware as a legitimate software program or data file. Once the Trojan is installed on an AI system, it can give the attacker unauthorized access to the system, steal sensitive data, or cause other types of damage. In the case of AI, Trojan attacks can be particularly damaging because they can manipulate the algorithms that make decisions based on data, leading to incorrect or even dangerous outcomes.

How it works

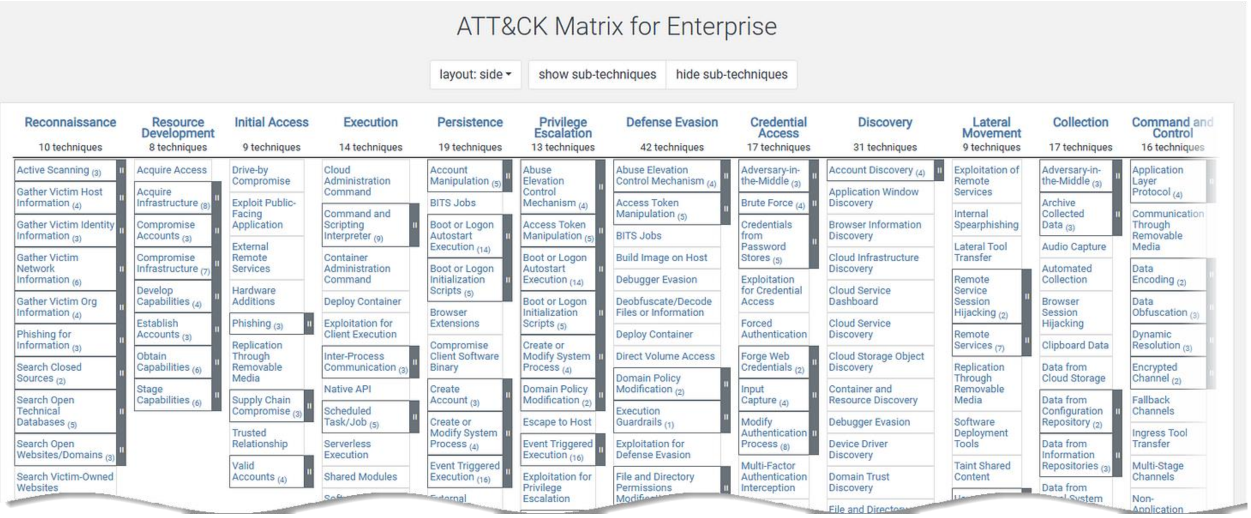
The general steps taken in a Trojan attack against AI can vary, but here are some common steps that attackers may take:

1. **Reconnaissance:** The attacker does research on the target AI system to identify vulnerabilities and weaknesses.
2. **Delivery:** The attacker delivers a Trojan to the AI system, often through email phishing, social engineering or through infected software.
3. **Installation:** The Trojan is installed on the AI system, allowing the attacker access to the system.
4. **Command and Control:** The attacker establishes a command-and-control infrastructure to remotely control the Trojan and carry out malicious actions.
5. **Exploitation:** The attacker exploits the Trojan to carry out malicious actions, which can include stealing sensitive data,

manipulating algorithms to produce incorrect results, or causing other types of damage.

6. **Cover-up:** The attacker may attempt to cover up their tracks to avoid detection and continue their malicious activities.

These steps take a similar approach to the adversary tactics and techniques of the **MITRE ATT&CK Matrix for Enterprise**. If not already, you should become very familiar with these threat models and methodologies.



ATT&CK Matrix for Enterprise

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 9 techniques	Execution 14 techniques	Persistence 19 techniques	Privilege Escalation 13 techniques	Defense Evasion 42 techniques	Credential Access 17 techniques	Discovery 31 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques
Active Scanning (3)	Acquire Access	Drive-by Compromise	Cloud Administration Command	Account Manipulation (5)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (3)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Exploit Public-Facing Application	Command and Scripting Interpreter (9)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media
Gather Victim Identity Information (3)	Compromise Accounts (3)	External Remote Services	Container Administration Command	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	BITS Jobs	Credentials from Password Stores (5)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)
Gather Victim Network Information (6)	Compromise Infrastructure (7)	Hardware Additions	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Obfuscation (3)
Gather Victim Org Information (4)	Develop Capabilities (4)	Phishing (3)	Exploitation for Client Execution	Browser Extensions	Create or Modify System Process (4)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (7)	Browser Session Hijacking	Dynamic Resolution (3)
Phishing for Information (3)	Establish Accounts (3)	Replication Through Removable Media	Inter-Process Communication (3)	Compromise Client Software Binary	Domain Policy Modification (2)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel (2)
Search Closed Sources (2)	Obtain Capabilities (6)	Supply Chain Compromise (3)	Native API	Create Account (3)	Escape to Host	Deploy Container	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Fallback Channels
Search Open Technical Databases (5)	Stage Capabilities (6)	Trusted Relationship	Scheduled Task/Job (5)	Create or Modify System Process (4)	Event Triggered Execution (16)	Direct Volume Access	Modify Authentication Process (8)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Ingress Tool Transfer
Search Open Websites/Domains (3)		Valid Accounts (4)	Serverless Execution	Event Triggered Execution (16)	Exploitation for Privilege Escalation	Domain Policy Modification (2)	Multi-Factor Authentication Interception	Debugger Evasion		Data from Information Repositories (3)	Multi-Stage Channels
Search Victim-Owned Websites			Shared Modules			Execution Guardrails (1)		Device Driver Discovery		Data from	Non-Application
						Exploitation for Defense Evasion		Domain Trust Discovery			
						File and Directory Permissions		File and Directory			

MITRE ATT&CK Matrix

See: [ATT&CK Matrix for Enterprise]

Types of Trojan attacks

There are different types of Trojan attacks against AI. Here are a few examples:

1. **[Data Poisoning:]** In this type of attack, the attacker injects incorrect or malicious data into an AI system, which can manipulate the system's decision-making process.

2. **Model Stealing:** In this type of attack, the attacker steals the AI model used by a company or organization, which can allow the attacker to replicate the model and use it for malicious purposes.
3. **Backdoor Access:** In this type of attack, the attacker gains unauthorized access to an AI system by exploiting a vulnerability or creating a backdoor.
4. **Adversarial Attacks:** In this type of attack, the attacker creates adversarial inputs that can cause an AI system to produce incorrect or unexpected outputs.
5. **Malware Injection:** In this type of attack, the attacker injects malware into an AI system through a Trojan, which can allow the attacker to control the system and carry out malicious activities.

It's important to be aware of these different types of Trojan attacks against AI and take appropriate measures to prevent them.

Why it matters

The negative results of a Trojan attack against AI can be severe and can vary depending on the type and severity of the attack. Here are some possible negative results:

1. **Data Theft:** Attackers can use Trojan attacks to steal sensitive data from an AI system, such as personal information, financial data, or intellectual property.
2. **Manipulation of Algorithms:** Attackers can use Trojan attacks to manipulate the algorithms used by an AI system, which can result in incorrect or biased decisions.
3. **System Disruption:** Trojan attacks can disrupt the functioning of an AI system, which can cause it to malfunction or stop working altogether.

4. **Financial Loss:** Trojan attacks can result in financial loss for companies or organizations, either through theft of funds or loss of revenue due to system disruption.
5. **Reputation Damage:** If a company or organization is the victim of a Trojan attack, it can damage their reputation and erode trust with customers and partners.

These negative results can have long-lasting consequences for companies or organizations that fall victim to Trojan attacks against AI, which is why it's important to take preventative measures to secure these systems.

How it might happen

A Trojan attack against AI can happen in several ways, but here are some common methods:

1. **Social Engineering:** Attackers may use social engineering tactics to trick users into downloading and installing Trojan malware, often through phishing emails or other types of social engineering attacks.
2. **Software Vulnerabilities:** Attackers may exploit vulnerabilities in software or operating systems used by an AI system to gain access and install Trojan malware.
3. **Third-Party Software:** Attackers may target third-party software components or libraries used by an AI system, which can contain vulnerabilities that can be exploited to install Trojan malware.
4. **Malicious Websites:** Attackers can use malicious websites to exploit vulnerabilities in a user's browser or operating system, which can allow them to install Trojan malware on the AI system.
5. **Physical Access:** Attackers may gain physical access to an AI system and install Trojan malware directly onto the system.

Once the Trojan malware is installed on the AI system, the attacker can use it to remotely control the system, steal data, or manipulate algorithms to produce incorrect or biased results.

How to Mitigate

There are several ways to mitigate Trojan attacks against AI, including:

1. **Use of Antivirus and Firewall Software:** Antivirus and firewall software can help prevent Trojan malware from being installed on an AI system and can detect and block malicious activity.
2. **Regular Software Updates:** Regular software updates can help fix vulnerabilities in the software or operating system used by the AI system, making it more difficult for attackers to exploit these vulnerabilities.
3. **Strong Access Controls:** Implementing strong access controls, such as limiting user access to only what is necessary and requiring multi-factor authentication, can help prevent unauthorized access to the AI system.
4. **Employee Education:** Educating employees on how to recognize and prevent social engineering attacks, such as phishing emails, can help prevent Trojan malware from being installed on the AI system.
5. **Adversarial Training:** Adversarial training involves training an AI system to recognize and defend against adversarial attacks, such as adversarial inputs or data poisoning.
6. **Regular System Audits:** Regular system audits can help identify vulnerabilities and weaknesses in the AI system, allowing them to be addressed before they can be exploited by attackers.

By implementing these mitigation strategies, companies and organizations can better protect their AI systems from Trojan attacks and other types of cyber threats.

How to monitor

To monitor against Trojan attacks against AI, here are some steps you can take:

1. **Implement Real-Time Monitoring:** Implementing real-time monitoring of AI systems can help detect and alert security teams to any unusual activity or attempts to access the system.
2. **Implement Intrusion Detection and Prevention:** Intrusion detection and prevention systems can help detect and prevent unauthorized access to AI systems, including Trojan attacks.
3. **Use Machine Learning:** Machine learning can be used to detect anomalies in the behavior of an AI system and flag any suspicious activity that could be indicative of a Trojan attack.
4. **Conduct Regular Penetration Testing:** Regular penetration testing can help identify vulnerabilities in an AI system, allowing them to be addressed before they can be exploited by attackers.
5. **Monitor Network Traffic:** Monitoring network traffic can help detect any attempts to exfiltrate data from an AI system or any suspicious activity that could be indicative of a Trojan attack.
6. **Implement User Behavior Analytics:** User behavior analytics can help detect any unusual or suspicious behavior by users of an AI system, which could be indicative of a Trojan attack.

By implementing these monitoring strategies, companies and organizations can better protect their AI systems from Trojan attacks and other types of cyber threats. It's important to continually evaluate and update monitoring strategies to ensure that they are effective and up to date with the latest threats.

What to capture

To identify when a Trojan attack against AI is happening, you should capture the following types of data:

1. **Network Traffic:** Monitoring network traffic can help detect any unusual traffic patterns that could be indicative of a Trojan attack. This includes capturing data on the volume and frequency of data transfers, the source and destination IP addresses, and the type of data being transferred.
2. **System Logs:** System logs can provide valuable information on user activity, system performance, and security events. Capturing data on user logins, system activity, and system errors can help detect any unusual or suspicious activity that could be indicative of a Trojan attack.
3. **User Behavior Analytics:** Capturing data on user behavior, such as the types of files accessed, the frequency of access, and the times of day when access occurs, can help detect any unusual or suspicious behavior that could be indicative of a Trojan attack.
4. **AI Model Performance Metrics:** Capturing data on the performance of an AI model, such as accuracy, precision, and recall, can help detect any unusual or unexpected changes in the performance of the model that could be indicative of a Trojan attack.
5. **Security Alerts:** Capturing data on security alerts generated by intrusion detection and prevention systems, firewalls, and antivirus software can help detect any attempted or successful Trojan attacks.

By capturing and analyzing this data, companies and organizations can better detect and respond to Trojan attacks against AI, helping to mitigate their impact and reduce the risk of data theft, system disruption, and other negative consequences.

To prevent Trojan attacks against AI, it's important to maintain strong cybersecurity practices, including regular software updates, strong passwords, and employee education about phishing and social engineering tactics. To prevent Trojan attacks, it's important to keep software up-to-date, use strong passwords, and be cautious when downloading files from unknown sources. This can include using secure coding practices, regularly updating software and systems, and implementing strong access controls and monitoring.

LEARNING ABOUT SECURITY AND AI IS LIKE HAVING YOUR CAKE AND EATING IT TOO – A FABULOUS COMBO THAT EVERYONE NEEDS IN THEIR LIVES. HERE ARE A FEW REASONS WHY THIS TOPIC IS SO IMPORTANT:

CYBERSECURITY THREATS ARE THE NEW BLACK: WITH TECHNOLOGY ADVANCING FASTER THAN YOU CAN SAY "HOLD MY LATTE," CYBERCRIMINALS ARE FINDING NEW, INNOVATIVE WAYS TO WREAK HAVOC. GETTING YOUR HANDS DIRTY IN THE WORLD OF SECURITY AND AI HELPS YOU STAY ONE STEP AHEAD OF THOSE PESKY HACKERS AND PROTECT YOUR PRECIOUS DATA.

PRIVACY, PLEASE: UNLESS YOU WANT YOUR PERSONAL INFORMATION TO BECOME THE TALK OF THE DIGITAL TOWN, YOU BETTER KNOW A THING OR TWO ABOUT SECURING YOUR AI SYSTEMS. KNOWLEDGE IS POWER, AND POWER KEEPS YOUR PRIVATE LIFE AS PRIVATE AS IT SHOULD BE.

THE RISE OF THE AI EMPIRE: AI IS PRACTICALLY EVERYWHERE THESE DAYS, FROM YOUR PHONE'S VOICE ASSISTANT TO SELF-DRIVING CARS. BY UNDERSTANDING HOW SECURITY AND AI INTERSECT, YOU'LL BE BETTER EQUIPPED TO NAVIGATE THIS BRAVE NEW WORLD AND MAKE INFORMED DECISIONS, BOTH PERSONALLY AND PROFESSIONALLY.

ETHICAL CONUNDRUMS GALORE: AS AI CONTINUES TO STRUT ITS STUFF, ETHICAL QUESTIONS ARE POPPING UP LIKE NOBODY'S BUSINESS. LEARNING ABOUT SECURITY AND AI HELPS YOU WRAP YOUR HEAD AROUND THESE COMPLEX ISSUES AND PARTICIPATE IN MEANINGFUL CONVERSATIONS THAT SHAPE THE FUTURE OF TECHNOLOGY.

EMBRACE THE FABULOUS WORLD OF SECURITY AND AI, AND PIVOT CONFIDENTLY INTO A FUTURE WHERE YOU'RE IN CONTROL, WELL-INFORMED, AND READY TO KICK SOME CYBER-BUTT!