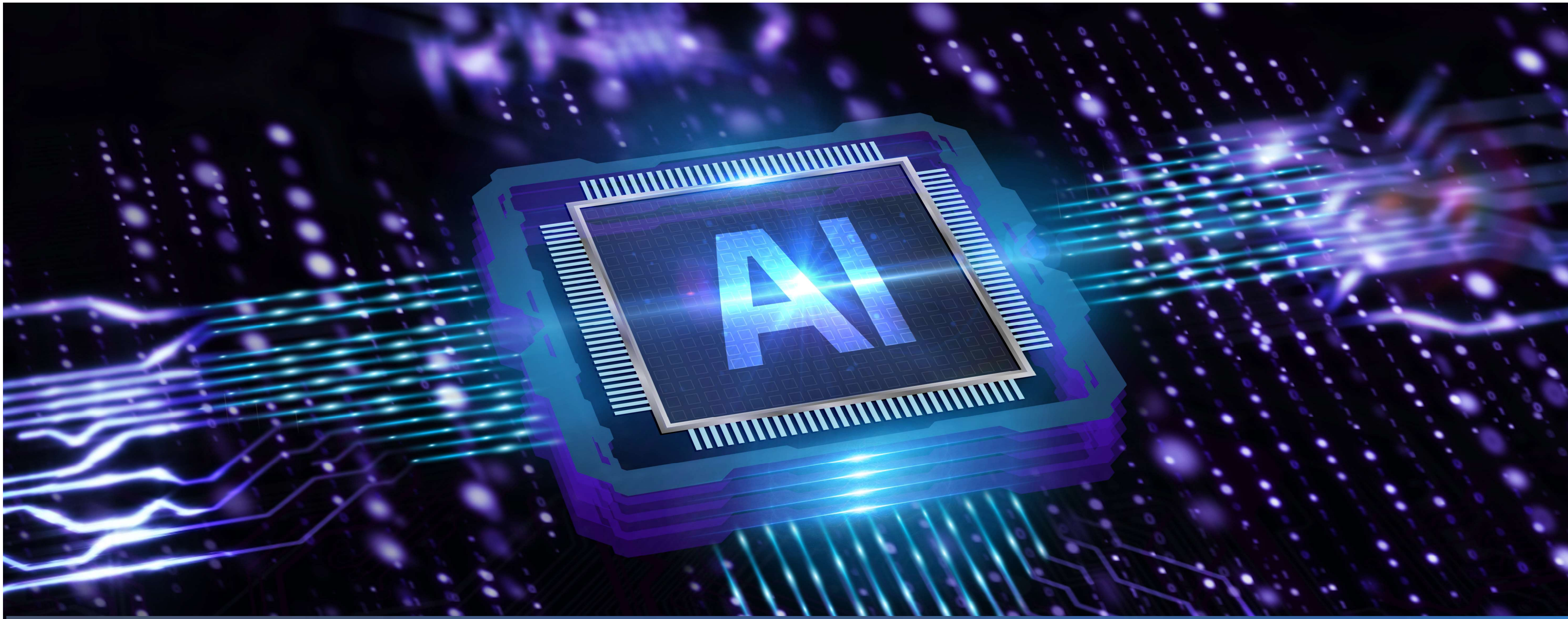
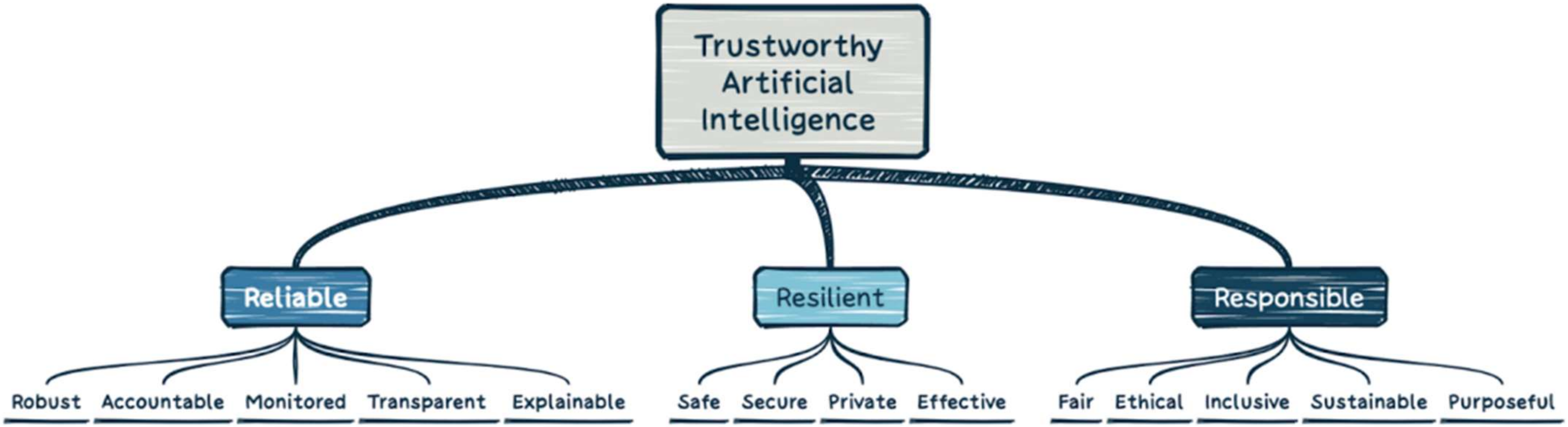


Appendix 12: More on AI Threat Frameworks

THE PILLARS OF TRUSTWORTHY AI



https://owasp.org/www-project-top-10-for-large-language-model-applications/llm-top-10-governance-doc/LLM_AI_Security_and_Governance_Checklist-v1.pdf

OVERVIEW OF FRAMEWORKS

SAIF (Secure AI Framework)

The Secure AI Framework (SAIF) is designed to provide guidelines and best practices for building secure AI systems. It emphasizes the security aspects of AI development and deployment, ensuring that AI systems are robust against adversarial attacks and vulnerabilities.

MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems)

Developed by MITRE, ATLAS provides a knowledge base of adversarial tactics, techniques, and case studies specific to AI systems. It is designed to help organizations understand, mitigate, and defend against AI threats.

NIST AI Risk Management Framework

Developed by the National Institute of Standards and Technology (NIST), this framework aims to manage risks associated with AI by providing guidelines for AI system development, deployment, and operation. It covers aspects such as reliability, security, and accountability.

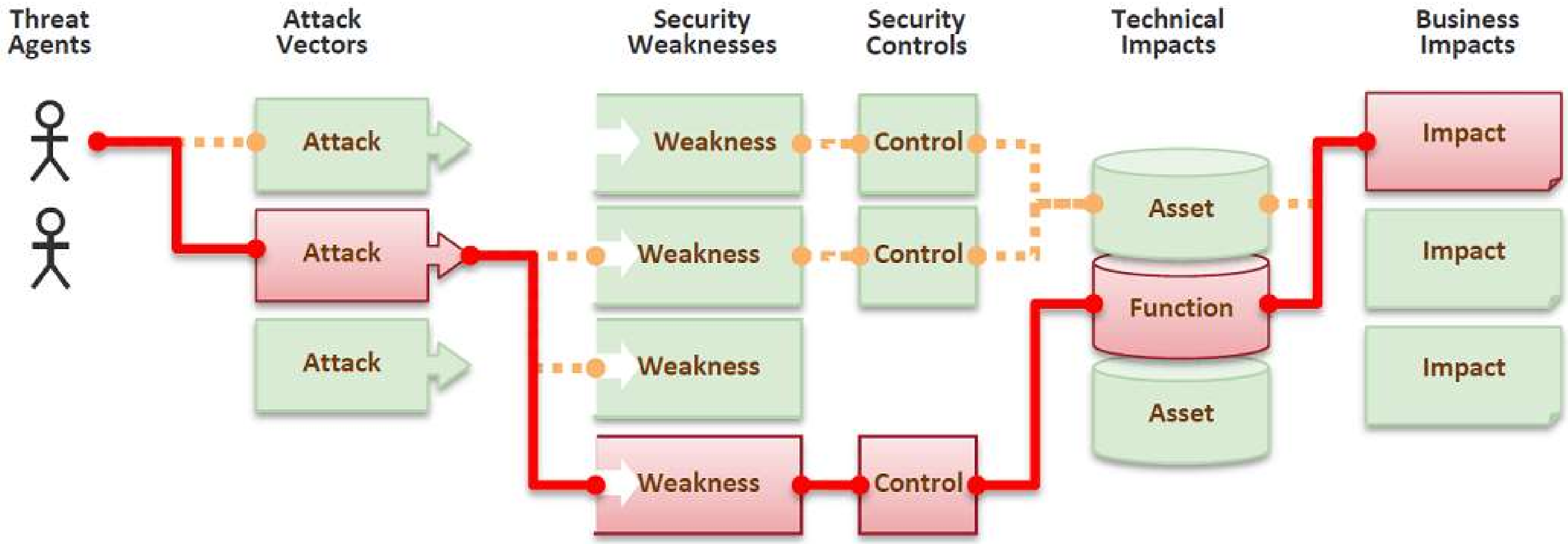
ISO/IEC JTC 1/SC 42

This international standard, developed by the ISO and IEC, provides guidelines for AI security and trustworthiness. It addresses various aspects of AI system security, including data integrity, privacy, and resilience against attacks.

OWASP AI SECURITY AND PRIVACY GUIDE

- OWASP AI security & privacy guide. It has two parts:
  - How to address AI security
  - How to address AI privacy
- https://owasp.org/www-project-ai-security-and-privacy-guide/

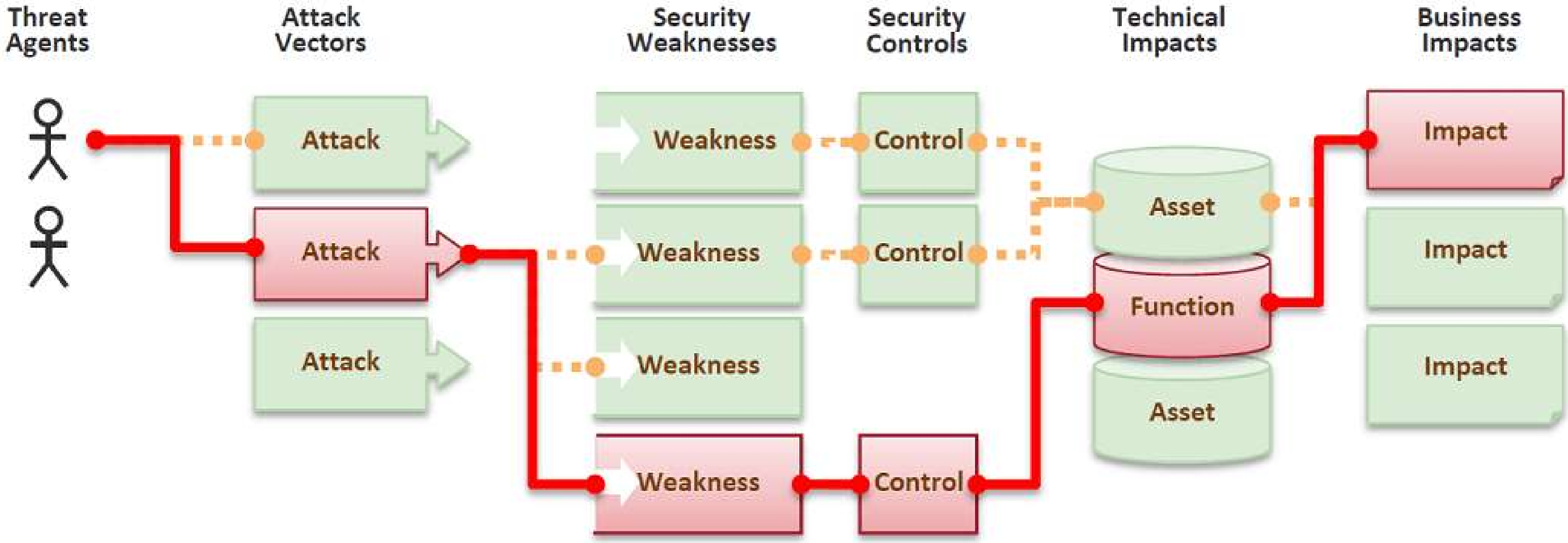
ANALYSIS OF LLM/GENAI CYBERSECURITY



An attack vector is the path or method that a cybercriminal uses when attempting to gain illegitimate access to a product or a system. Most attack vectors attempt to exploit a vulnerability in a system or application.

- An attack vector is the method a cybercriminal uses to gain unauthorized access. An attack surface is a set of points on the boundary of a system, a system element, or an environment where an attacker can try to enter, cause an effect on, or extract data from that system, or system element,.
    - The most common types of attack vectors in embedded systems include compromised weak passwords or credentials, misconfigurations, malware, security vulnerabilities, malicious insider and supply chain threats, weak encryption, malicious code, unpatched vulnerabilities in operating systems or computer systems, zero-day attacks that result in data breaches or confidential information leaks, and denial-of-service attacks.

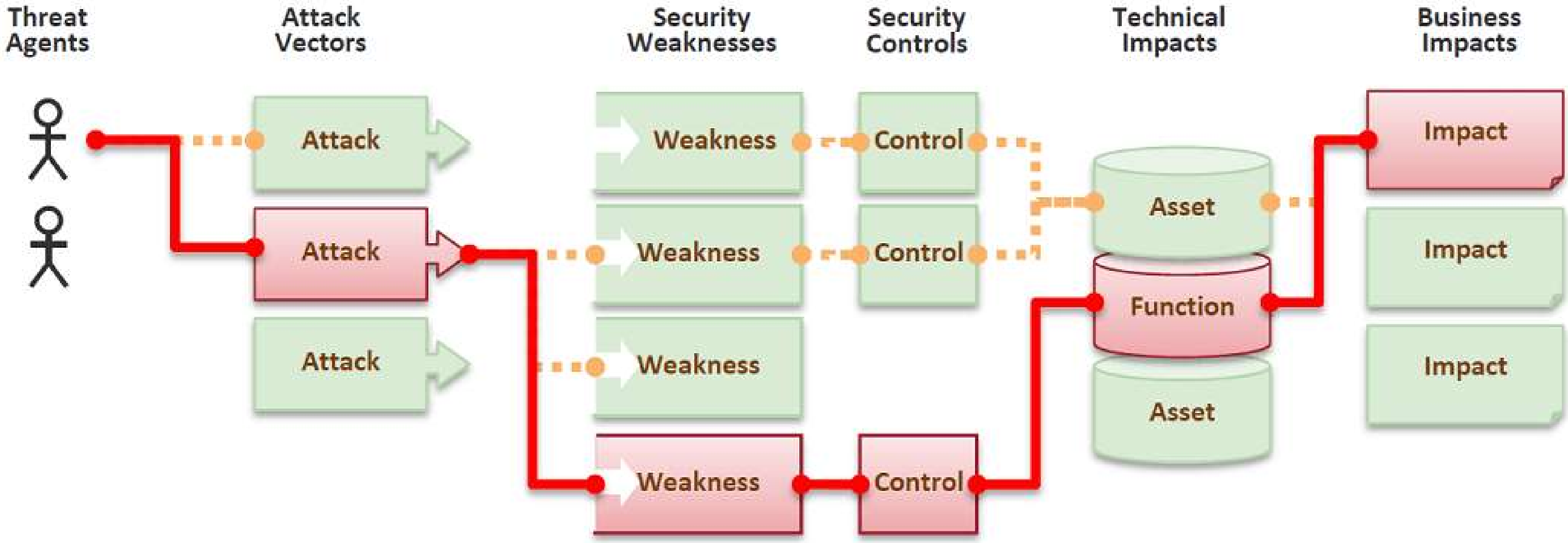
ANALYSIS OF LLM/GENAI SECURITY

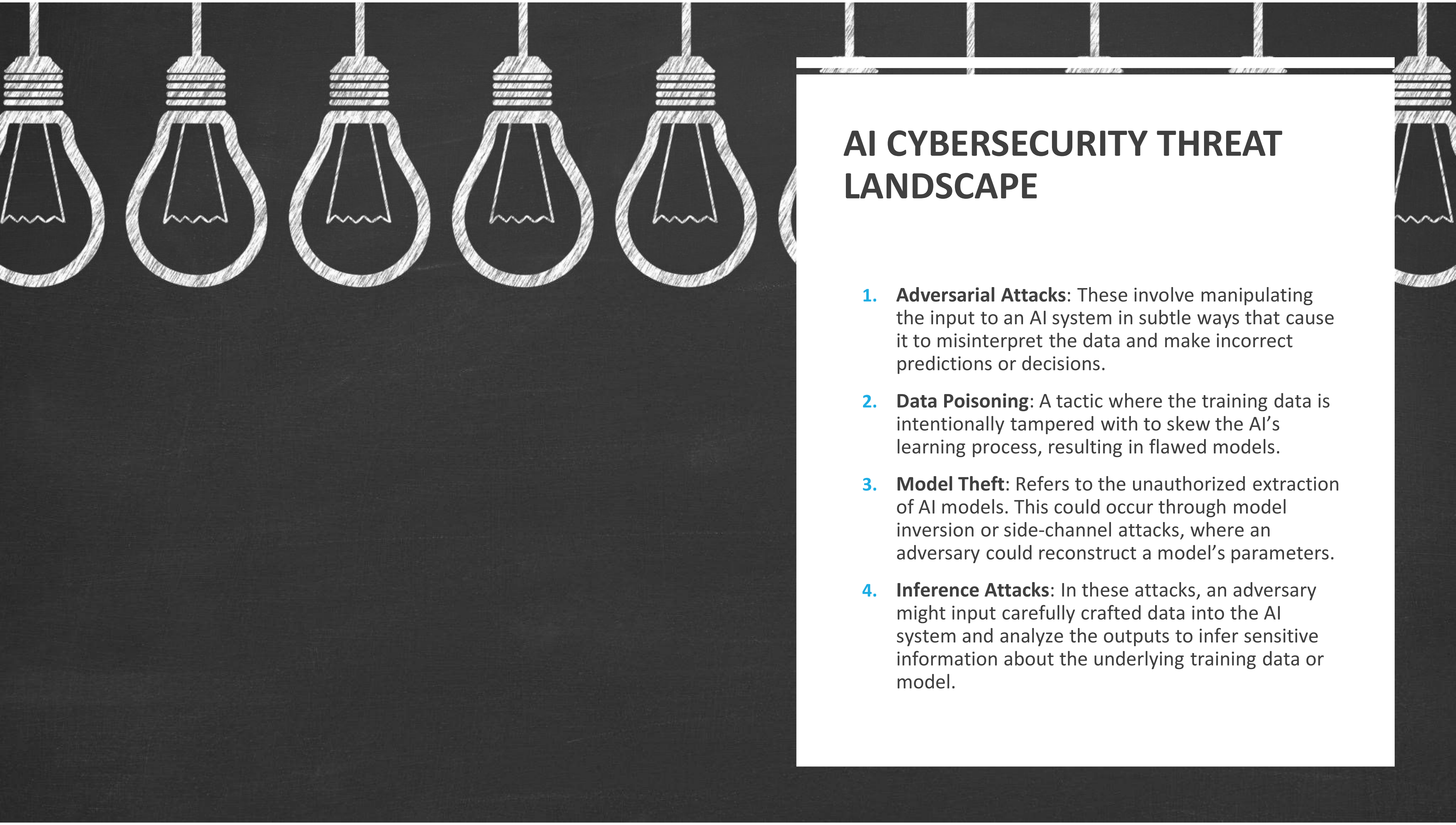


EXAMPLE OF LLM/GENAI ASSETS, THREAT AGENTS AND CONTROLS

Characteristic	Example
LLM/GenAI Assets	
	AI-Powered Chatbot System
	NLP algorithms trained on vast data
	Backend infrastructure (servers, databases, APIs)
Threat Agents	
	Malicious Users
	Cybercriminals
	Competitors
	AI-Enhanced Threat Actors
Controls	
	User Authentication and Authorization
	Input Validation and Sanitization
	Model Governance and Monitoring
	Data Privacy and Compliance
	Threat Intelligence and Response
	Incident Response Plan

SIMPLE ANALYSIS OF LLM/GENAI SECURITY





AI CYBERSECURITY THREAT LANDSCAPE

1. **Adversarial Attacks**: These involve manipulating the input to an AI system in subtle ways that cause it to misinterpret the data and make incorrect predictions or decisions.
2. **Data Poisoning**: A tactic where the training data is intentionally tampered with to skew the AI's learning process, resulting in flawed models.
3. **Model Theft**: Refers to the unauthorized extraction of AI models. This could occur through model inversion or side-channel attacks, where an adversary could reconstruct a model's parameters.
4. **Inference Attacks**: In these attacks, an adversary might input carefully crafted data into the AI system and analyze the outputs to infer sensitive information about the underlying training data or model.

LLM/GENAI THREAT AGENT, ATTACK VECTORS, SECURITY WEAKNESSES, SECURITY CONTROL, TECHNICAL IMPACTS AND BUSINESS IMPACT

1. **Threat Agent**: This refers to the entity or factor that has the potential to exploit a vulnerability in your system's security and cause harm. Threat agents can be individuals, groups, organizations, or automated systems.
2. **Attack Vectors**: These are the paths or means by which a threat agent can exploit vulnerabilities in a system. Attack vectors can include methods such as malware, phishing emails, software exploits, physical intrusion, etc.
3. **Security Weaknesses**: These are vulnerabilities or gaps in a system's security defenses that could be exploited by threat agents. Weaknesses can exist at various levels of a system, including hardware, software, network configurations, human practices, etc.
4. **Security Controls**: These are measures or mechanisms put in place to mitigate security risks and protect against threats. Security controls can include things like firewalls, encryption, access controls, intrusion detection systems, security policies, training programs, etc.
5. **Technical Impacts**: These are the consequences of a security breach or successful attack on a system from a technical standpoint. Technical impacts can include data loss or theft, system downtime, unauthorized access, corruption of data or software, etc.
6. **Business Impact**: This refers to the effects that a security incident can have on the business or organization, beyond just the technical consequences. Business impacts can include financial losses, damage to reputation, legal liabilities, regulatory fines, loss of customer trust, etc.

EXAMPLE OF LLM/GENAI ASSETS

- A sophisticated conversational AI chatbot system developed using LLM/GenAI technology.
- Natural language processing (NLP) algorithms trained on vast amounts of data to understand and respond to user queries.
- Backend infrastructure for hosting and maintaining the chatbot system, including servers, databases, and APIs.

Threat Agents:

•**Malicious Users**: Individuals or groups who aim to exploit vulnerabilities in the chatbot system for personal gain or to cause harm.

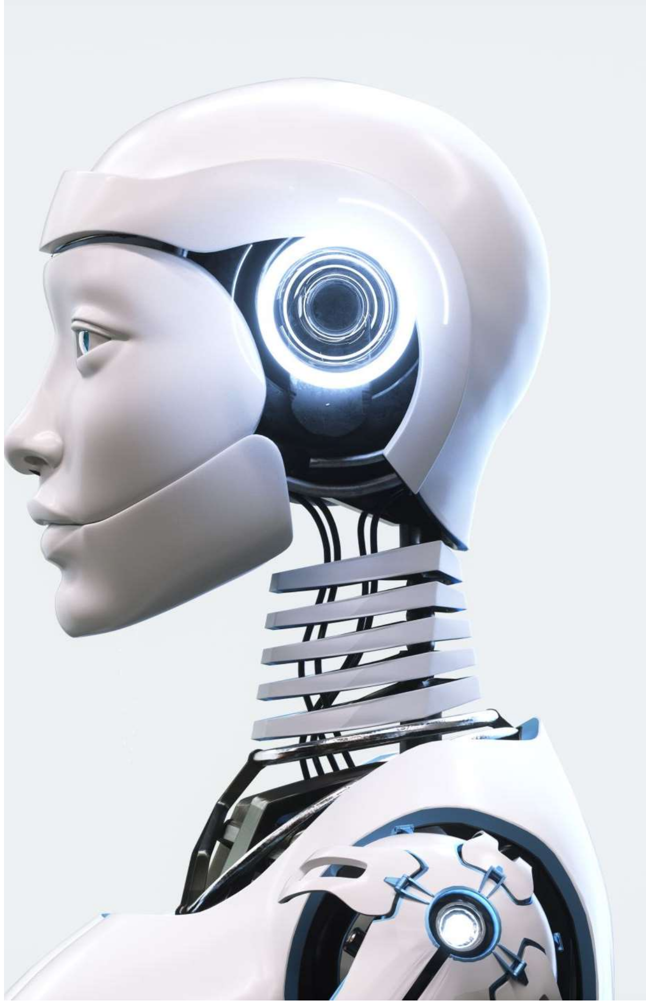
•**Cybercriminals**: Hackers who may attempt to infiltrate the system to steal sensitive data, spread malware, or launch denial-of-service attacks.

•**Competitors**: Rival companies or entities seeking to disrupt the chatbot service to gain a competitive advantage.

•**AI-Enhanced Threat Actors**: Adversaries who leverage AI technologies, including LLM/GenAI, to craft sophisticated attacks targeting the chatbot system.

LLM/GENAI (LARGE LANGUAGE MODEL/GENERATIVE ARTIFICIAL INTELLIGENCE) THREAT AGENTS

1. **AI-Powered Malware**: Malicious actors can use LLM/GENAI to develop sophisticated malware that can adapt and evolve to evade traditional security measures. These AI-powered malware can be programmed to learn and mimic user behavior, making detection and mitigation more challenging.
2. **Automated Social Engineering**: LLM/GENAI can be used to generate highly convincing and personalized phishing emails, messages, or social media posts. These automated social engineering attacks can trick users into revealing sensitive information or performing actions that compromise security.
3. **Fake News and Disinformation**: Threat actors can leverage LLM/GENAI to generate fake news articles, videos, or social media posts aimed at spreading disinformation, manipulating public opinion, or inciting social unrest. This can have serious implications for political stability, public trust, and social cohesion.
4. **AI-Enhanced Spear Phishing**: LLM/GENAI can assist attackers in crafting targeted spear phishing attacks by analyzing publicly available information about individuals and organizations. This enables attackers to create highly personalized and convincing messages tailored to specific targets, increasing the likelihood of success.
5. **AI-Driven Insider Threats**: Insiders with malicious intent can use LLM/GENAI to bypass security controls and exfiltrate sensitive data or sabotage systems. For example, an employee with access to LLM/GENAI could use it to generate fake credentials, manipulate data, or create backdoors within the system.



LLM/GENAI ATTACK VECTORS

- **AI-Enhanced Phishing**:
  - Attackers can use LLM/GENAI to generate highly convincing phishing emails, messages, or websites. These phishing attempts can be tailored to specific targets using AI-driven personalization techniques, making them more likely to succeed in tricking users into revealing sensitive information such as login credentials or financial details.
- **AI-Driven Social Engineering**:
  - LLM/GENAI can be utilized to create fake social media profiles or automated chatbots that engage with users to extract sensitive information or manipulate them into taking malicious actions. These AI-driven social engineering tactics can be difficult to detect due to their human-like conversational abilities.
- **AI-Generated Malware**:
  - Malicious actors can leverage LLM/GENAI to develop sophisticated malware variants that can adapt and evolve over time. AI-generated malware may employ evasion techniques to bypass traditional security measures and exploit vulnerabilities in systems, leading to data theft, system compromise, or disruption of services.
- **AI-Powered Reconnaissance**:
  - Attackers can use LLM/GENAI to conduct automated reconnaissance and intelligence gathering. By analyzing large volumes of data from various sources, including social media, public records, and online forums, AI-powered reconnaissance can provide attackers with valuable insights for planning targeted cyberattacks or social engineering campaigns.
- **AI-Driven Content Generation**:
  - LLM/GENAI can be employed to generate fake news articles, reviews, or product listings that are designed to deceive or manipulate readers. These AI-generated content pieces can be used for disinformation campaigns, reputation attacks, or influencing public opinion in malicious ways.
- **AI-Assisted Brute Force Attacks**:
  - Attackers can utilize LLM/GENAI to enhance brute force attacks by generating and testing a large number of password or encryption key combinations. AI-driven brute force attacks can be more efficient and effective in cracking weak credentials or cryptographic algorithms, leading to unauthorized access or data decryption.

LLM/GENAI SECURITY WEAKNESSES

Data Privacy Concerns:

LLM/GENAI models often require large amounts of data for training, which can include sensitive or confidential information. Inadequate data privacy measures during the data collection, storage, or processing stages can lead to privacy breaches or data leaks.

Bias and Fairness Issues:

LLM/GENAI models may inherit biases from the training data, leading to biased outputs or decisions. These biases can result in unfair treatment, discrimination, or misrepresentation, especially in applications such as hiring, lending, or criminal justice.

Adversarial Attacks:

LLM/GENAI models are susceptible to adversarial attacks where malicious inputs are carefully crafted to deceive the model and produce incorrect outputs. Adversarial examples can be used to bypass security mechanisms, such as spam filters or image recognition systems.

Data Poisoning:

Malicious actors can manipulate or inject poisoned data into LLM/GENAI training datasets to influence model behavior negatively. Data poisoning attacks can compromise the integrity and reliability of the model's predictions or classifications.

Model Vulnerabilities:

LLM/GENAI models may contain vulnerabilities that can be exploited by attackers to compromise their functionality or manipulate their outputs. Vulnerabilities such as input validation errors, buffer overflows, or logic flaws can be exploited to launch attacks, including model inversion, model extraction, or model inversion attacks.

Transfer Learning Risks:

Transfer learning, a technique used to fine-tune pre-trained LLM/GENAI models for specific tasks, can introduce security risks if not properly managed. Unauthorized access to fine-tuned models or transfer learning processes can lead to intellectual property theft or model misuse.

Explainability and Interpretability:

LLM/GENAI models often lack explainability and interpretability, making it challenging to understand how they arrive at their decisions or predictions. This opacity can hinder accountability, transparency, and trust in AI-driven systems, especially in critical applications such as healthcare or finance.

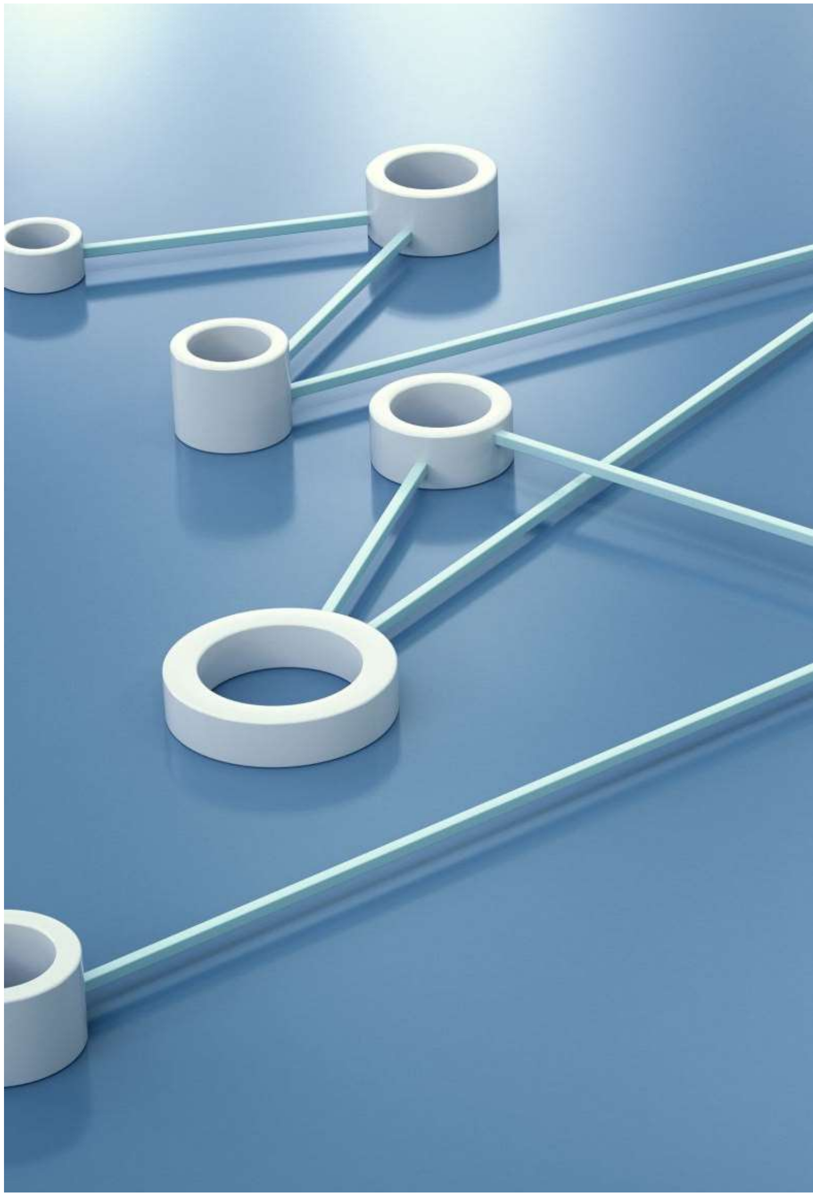
EXAMPLE OF CONTROLS

- **User Authentication and Authorization**: Implement robust authentication mechanisms to verify the identity of users interacting with the chatbot. Use access control mechanisms to ensure that users only have access to authorized functionalities and data.
- **Input Validation and Sanitization**: Validate and sanitize user inputs to prevent injection attacks, such as SQL injection or cross-site scripting (XSS). Use AI-powered anomaly detection techniques to identify and block suspicious inputs.
- **Model Governance and Monitoring**: Establish model governance practices to monitor the performance and behavior of the LLM/GenAI models powering the chatbot. Implement mechanisms for continuous monitoring, auditing, and version control to detect and mitigate potential biases, errors, or adversarial attacks.
- **Data Privacy and Compliance**: Ensure compliance with data protection regulations (e.g., GDPR, CCPA) by implementing robust data privacy measures. Encrypt sensitive user data at rest and in transit and enforce strict access controls to protect against unauthorized access or data breaches.
- **Threat Intelligence and Response**: Deploy AI-driven threat intelligence solutions to detect and respond to emerging threats targeting the chatbot system. Leverage machine learning algorithms to analyze user behavior, detect anomalous activities, and proactively mitigate security incidents.
- **Incident Response Plan**: Develop and regularly update an incident response plan outlining procedures for responding to security incidents and breaches affecting the chatbot system. Conduct regular tabletop exercises and simulations to test the effectiveness of the response plan and ensure readiness to address potential threats.

General Data Protection Regulation(GDPR) & the California Consumer Privacy Act (CCPA): CCPA and GDPR are compliance laws that aim at protecting user data from unauthorized access and processing. CCPA has often been called the 'GDPR lite' version in the compliance communities and there is a fairly supportive logical reasoning to that debate.

GENAI SECURITY BEST PRACTICES & FRAMEWORKS

- Google has released the Secure AI framework (SAIF) for organizations to provide a conceptual framework for securing AI systems. The framework mandates to:
  - **Proactive threat detection** and response for LLMs, leveraging threat intelligence, and automating defenses against LLM threats.
  - **Harmonize platform security** controls to ensure consistency such as enforcing least privilege permissions for LLM usage and development.
  - **Adaptation of application security controls** to LLM-specific threats and risks
  - **Feedback loop** when deploying and releasing LLM applications.
  - **Contextualize AI risks** in surrounding business processes.
- By integrating these principles from the SAIF, organizations can improve their security posture in LLM applications.

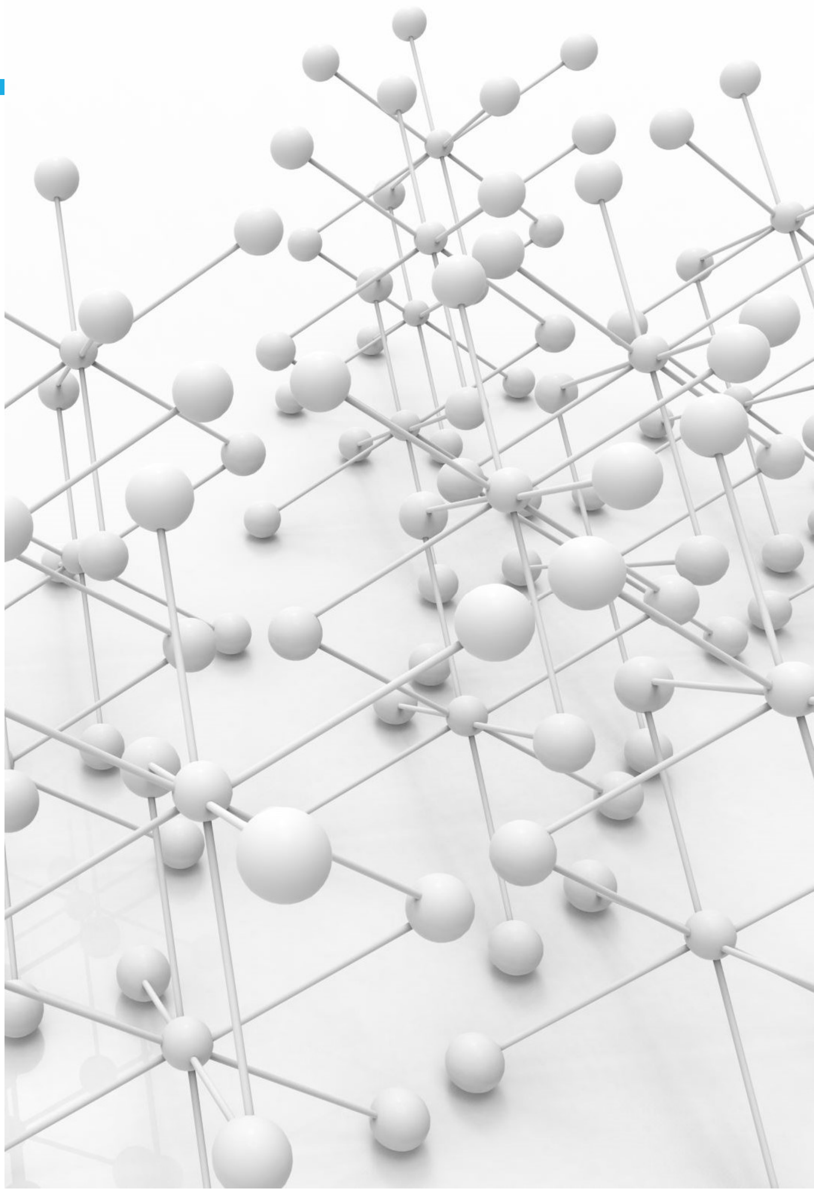


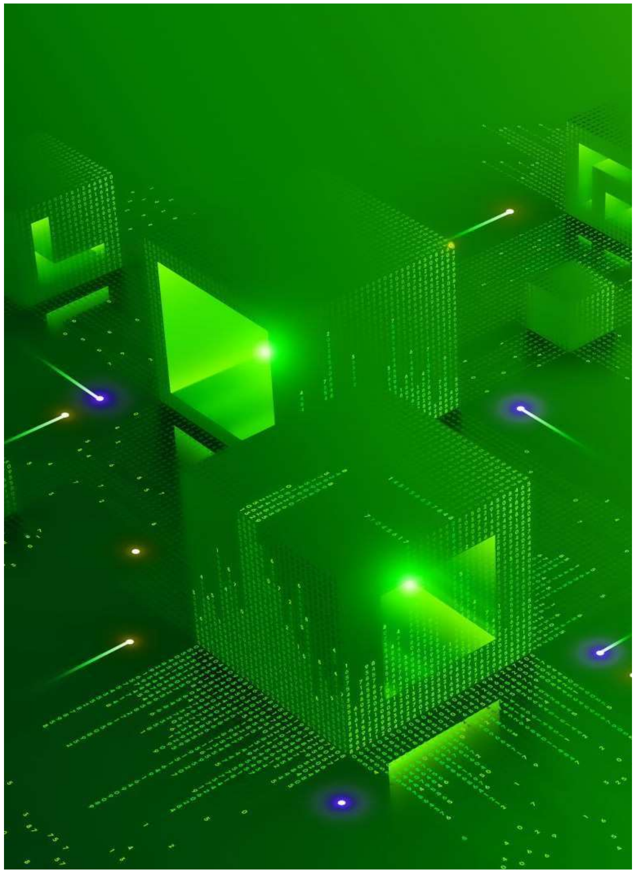
AI RISK MANAGEMENT PROGRAM

- To effectively manage GenAI risk, performing threat modeling for LLM applications is crucial, especially focusing on the major LLM threats discussed previously. To address these challenges comprehensively, an AI Risk Management Program is essential.

- In line with this, **NIST has released the AI Risk Management Framework**, specifically tailored for organizations looking to manage AI risk that engaged in the AI system lifecycle. The core objective of this framework is to manage AI-associated risks effectively and champion the secure and responsible implementation of AI systems.

https://www.nist.gov/itl/ai-risk-management-framework





MITRE ATT&CK®

MITRE ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations.

The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

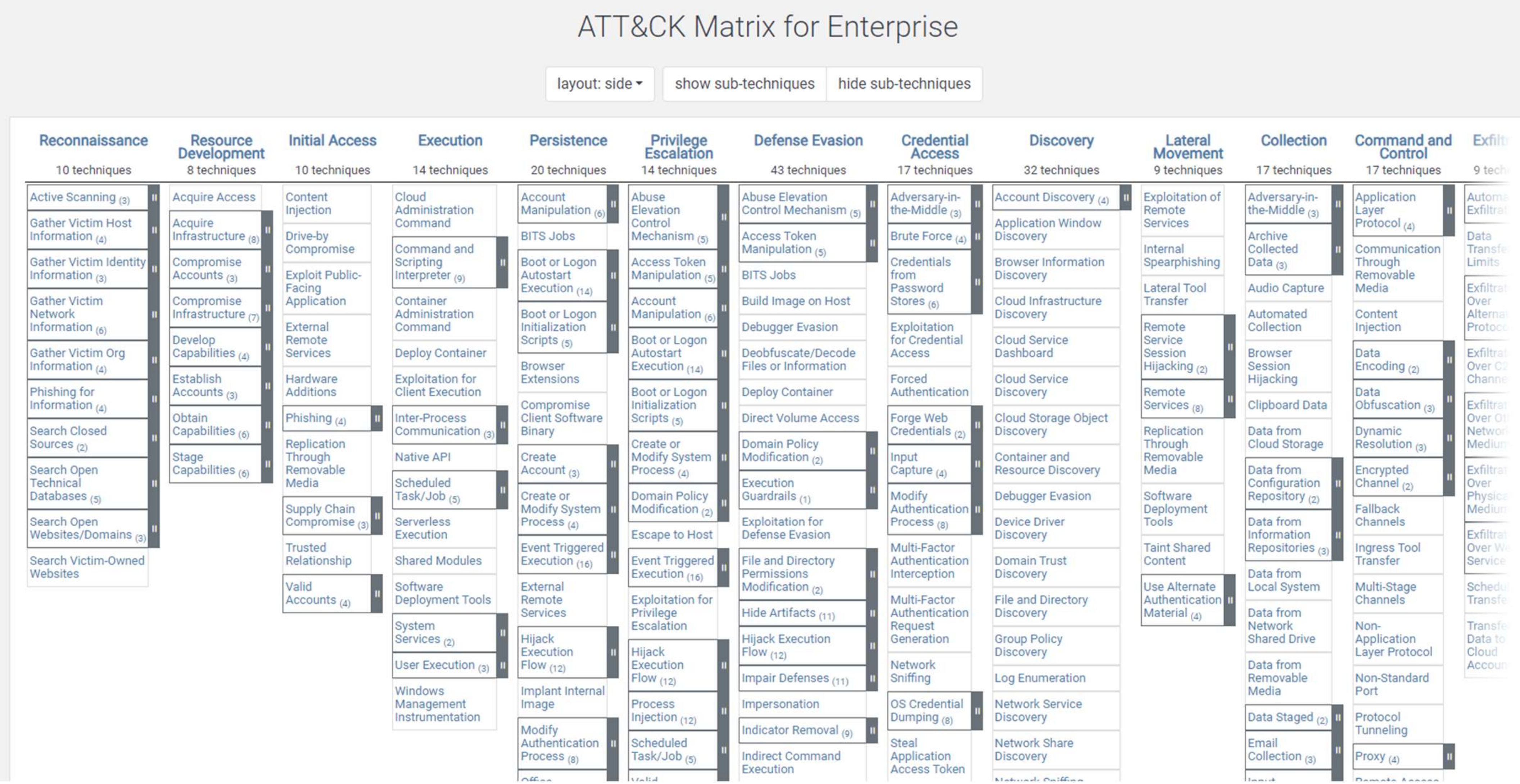
With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge.

MITRE ATT&CK

- **Focus and Scope:** MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. It is used for threat modeling and cybersecurity defense.

- **Structure:** The framework categorizes tactics (objectives) and techniques (methods) used by cyber adversaries. It is detailed and regularly updated with the latest threat intelligence.

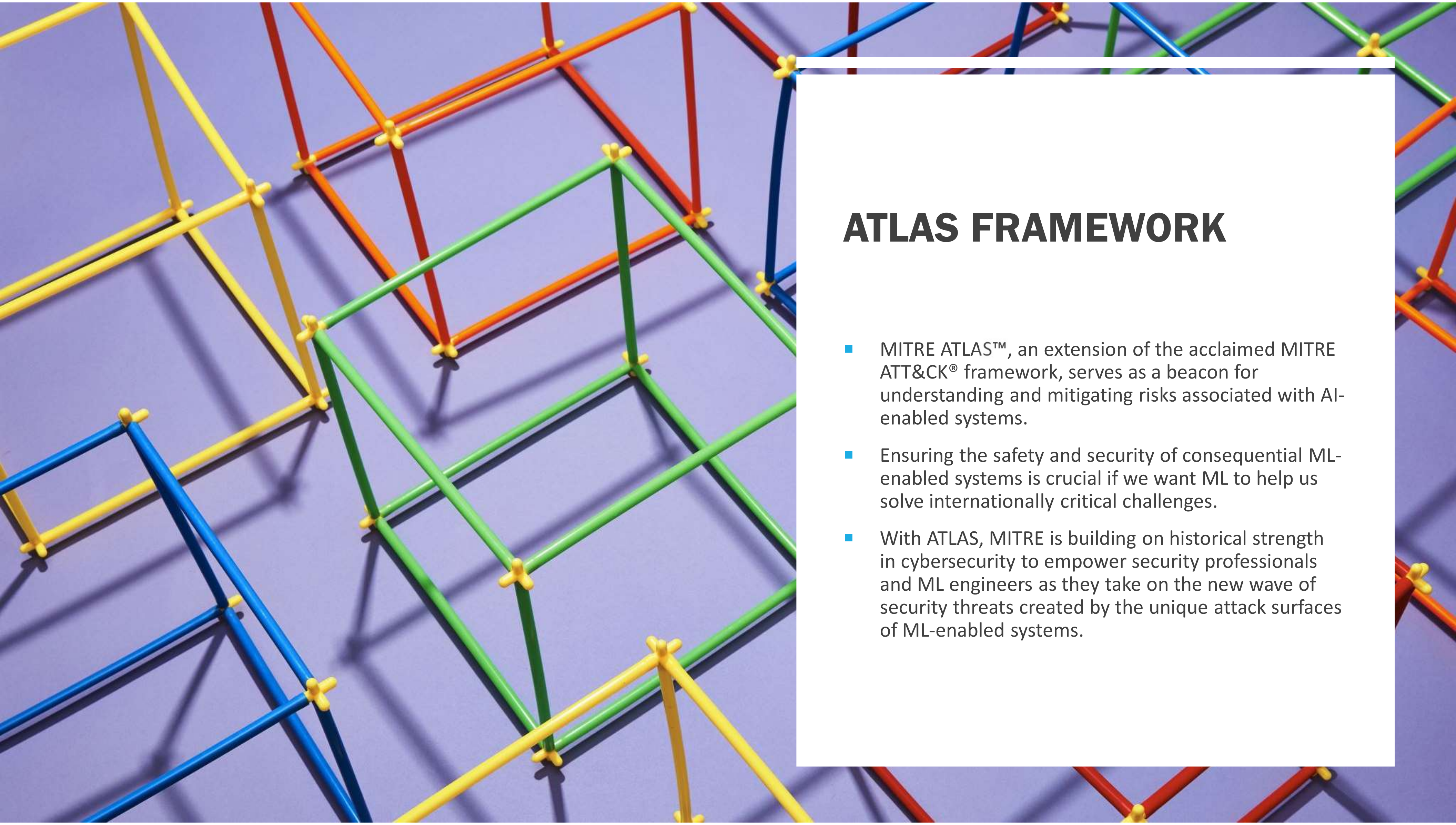
- **Application:** Primarily used for understanding attack behaviors, improving threat detection, and enhancing the cybersecurity posture of organizations against known attack vectors.



ATT&CK Matrix for Enterprise

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 10 techniques	Execution 14 techniques	Persistence 20 techniques	Privilege Escalation 14 techniques	Defense Evasion 43 techniques	Credential Access 17 techniques	Discovery 32 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 17 techniques	Exfilt 9 tech
Active Scanning (3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (6)	Abuse Elevation Control Mechanism (5)	Abuse Elevation Control Mechanism (5)	Adversary-in-the-Middle (3)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)	Automa Exfiltrat
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Drive-by Compromise	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	Data Transfe Limits
Gather Victim Identity Information (3)	Compromise Accounts (3)	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Autostart Execution (14)	Account Manipulation (6)	BITS Jobs	Credentials from Password Stores (6)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Content Injection	Exfiltrat Over Alterna Protoco
Gather Victim Network Information (6)	Compromise Infrastructure (7)	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Encoding (2)	Exfiltrat Over C2 Channe
Gather Victim Org Information (4)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Browser Extensions	Boot or Logon Initialization Scripts (5)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (8)	Browser Session Hijacking	Data Obfuscation (3)	Exfiltrat Over Ot Networ Mediu
Phishing for Information (4)	Establish Accounts (3)	Phishing (4)	Inter-Process Communication (3)	Compromise Client Software Binary	Create or Modify System Process (4)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Dynamic Resolution (3)	Exfiltrat Over Physica Mediu
Search Closed Sources (2)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Create Account (3)	Domain Policy Modification (2)	Deploy Container	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Encrypted Channel (2)	Exfiltrat Over We Service
Search Open Technical Databases (5)	Stage Capabilities (6)	Supply Chain Compromise (3)	Scheduled Task/Job (5)	Create or Modify System Process (4)	Escape to Host	Direct Volume Access	Modify Authentication Process (8)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Fallback Channels	Schedu Transfe
Search Open Websites/Domains (3)		Trusted Relationship	Serverless Execution	Event Triggered Execution (16)	Event Triggered Execution (16)	Domain Policy Modification (2)	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	Ingress Tool Transfer	Transfe Data to Cloud Accoun
Search Victim-Owned Websites		Valid Accounts (4)	Shared Modules	External Remote Services	Exploitation for Privilege Escalation	Execution Guardrails (1)	Multi-Factor Authentication Request Generation	Device Driver Discovery		Data from Local System	Multi-Stage Channels	
			Software Deployment Tools	Hijack Execution Flow (12)	Hijack Execution Flow (12)	Exploitation for Defense Evasion	Network Sniffing	Domain Trust Discovery		Data from Network Shared Drive	Non-Application Layer Protocol	
			System Services (2)	Implant Internal Image	Process Injection (12)	File and Directory Permissions Modification (2)	OS Credential Dumping (8)	File and Directory Discovery		Data from Removable Media	Non-Standard Port	
			User Execution (3)	Modify Authentication Process (8)	Scheduled Task/Job (5)	Hide Artifacts (11)	Steal Application Access Token	Group Policy Discovery		Data Staged (2)	Protocol Tunneling	
			Windows Management Instrumentation	Office	Valid	Hijack Execution Flow (12)		Log Enumeration		Email Collection (3)	Proxy (4)	
						Impair Defenses (11)		Network Service Discovery		Input	Remote Access	
						Impersonation		Network Share Discovery				
						Indicator Removal (9)		Network Sniffing				
						Indirect Command Execution						

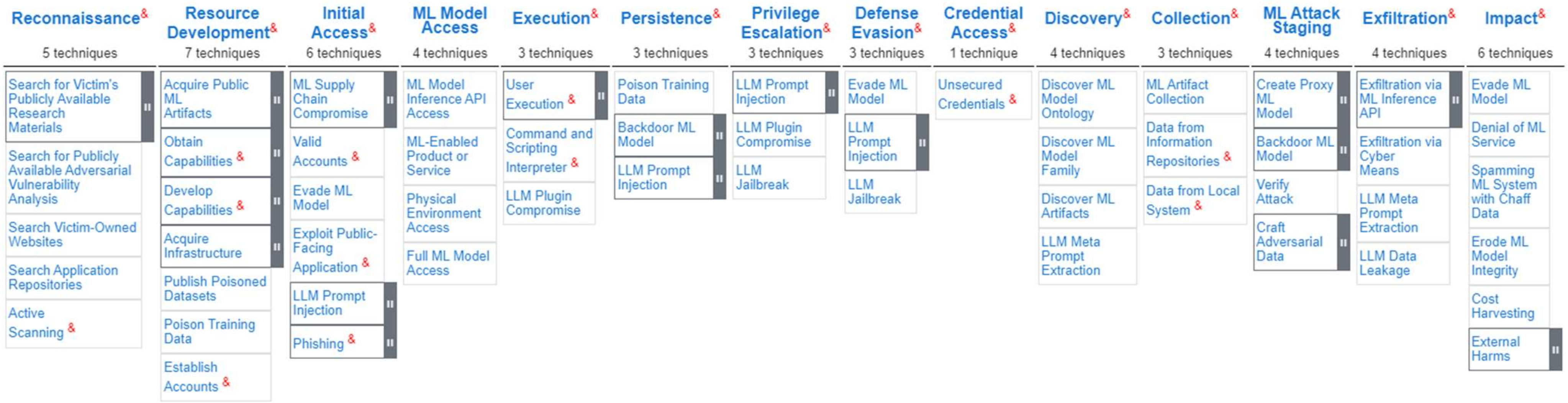
https://attack.mitre.org/



ATLAS FRAMEWORK

- MITRE ATLAS™, an extension of the acclaimed MITRE ATT&CK® framework, serves as a beacon for understanding and mitigating risks associated with AI-enabled systems.
- Ensuring the safety and security of consequential ML-enabled systems is crucial if we want ML to help us solve internationally critical challenges.
- With ATLAS, MITRE is building on historical strength in cybersecurity to empower security professionals and ML engineers as they take on the new wave of security threats created by the unique attack surfaces of ML-enabled systems.

ATLAS™



Reconnaissance&	Resource Development&	Initial Access&	ML Model Access	Execution&	Persistence&	Privilege Escalation&	Defense Evasion&	Credential Access&	Discovery&	Collection&	ML Attack Staging	Exfiltration&	Impact&
5 techniques	7 techniques	6 techniques	4 techniques	3 techniques	3 techniques	3 techniques	3 techniques	1 technique	4 techniques	3 techniques	4 techniques	4 techniques	6 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution &	Poison Training Data	LLM Prompt Injection	Evade ML Model	Unsecured Credentials &	Discover ML Model Ontology	ML Artifact Collection	Create Proxy ML Model	Exfiltration via ML Inference API	Evade ML Model
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities &	Valid Accounts &	ML-Enabled Product or Service	Command and Scripting Interpreter &	Backdoor ML Model	LLM Plugin Compromise	LLM Prompt Injection		Discover ML Model Family	Data from Information Repositories &	Backdoor ML Model	Exfiltration via Cyber Means	Denial of ML Service
Search Victim-Owned Websites	Develop Capabilities &	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection	LLM Jailbreak	LLM Jailbreak		Discover ML Artifacts	Data from Local System &	Verify Attack	LLM Meta Prompt Extraction	Spamming ML System with Chaff Data
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application &	Full ML Model Access						LLM Meta Prompt Extraction		Craft Adversarial Data	LLM Data Leakage	Erode ML Model Integrity
Active Scanning &	Publish Poisoned Datasets	LLM Prompt Injection											Cost Harvesting
	Poison Training Data	Phishing &											External Harms
	Establish Accounts &												

The progression of tactics used in attacks as columns from left to right, with ML techniques belonging to each tactic below. & indicates an adaption from ATT&CK.

https://atlas.mitre.org/tactics

USING MITRE ATLAS FOR STANDARDIZATION

- MITRE's Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS) framework provides a structured approach to understanding and mitigating AI threats. To use MITRE ATLAS effectively:

  - **Identify Relevant Tactics and Techniques:** Map out which tactics and techniques are most relevant to your AI systems and applications.
  - **Scenario-Based Planning:** Use ATLAS to develop scenarios that represent potential threats, aligning with your specific use cases.
  - **Implement Mitigations:** Leverage the mitigations suggested by ATLAS for each technique, customizing them to fit your organization's environment.
  - **Standardize Reporting:** Use the framework to standardize how threats and incidents are reported and analyzed, facilitating better communication and repeatability.

- **Detecting Malicious Behavior and Poisoning**

  - **Monitoring Model Performance:** Sudden changes in model accuracy or decision patterns can indicate an attack. Continuous performance monitoring can help in early detection.
  - **Analyzing Input Data:** Look for statistical anomalies or deviations in input data distributions, which can signal poisoning attempts.
  - **Implementing Watermarking:** Watermarking data and models can help trace back and identify when and where tampering or theft occurred.
  - **Using Explainability Tools:** Tools that provide insights into model decision-making can help identify when a model is making decisions based on manipulated inputs or poisoned data.

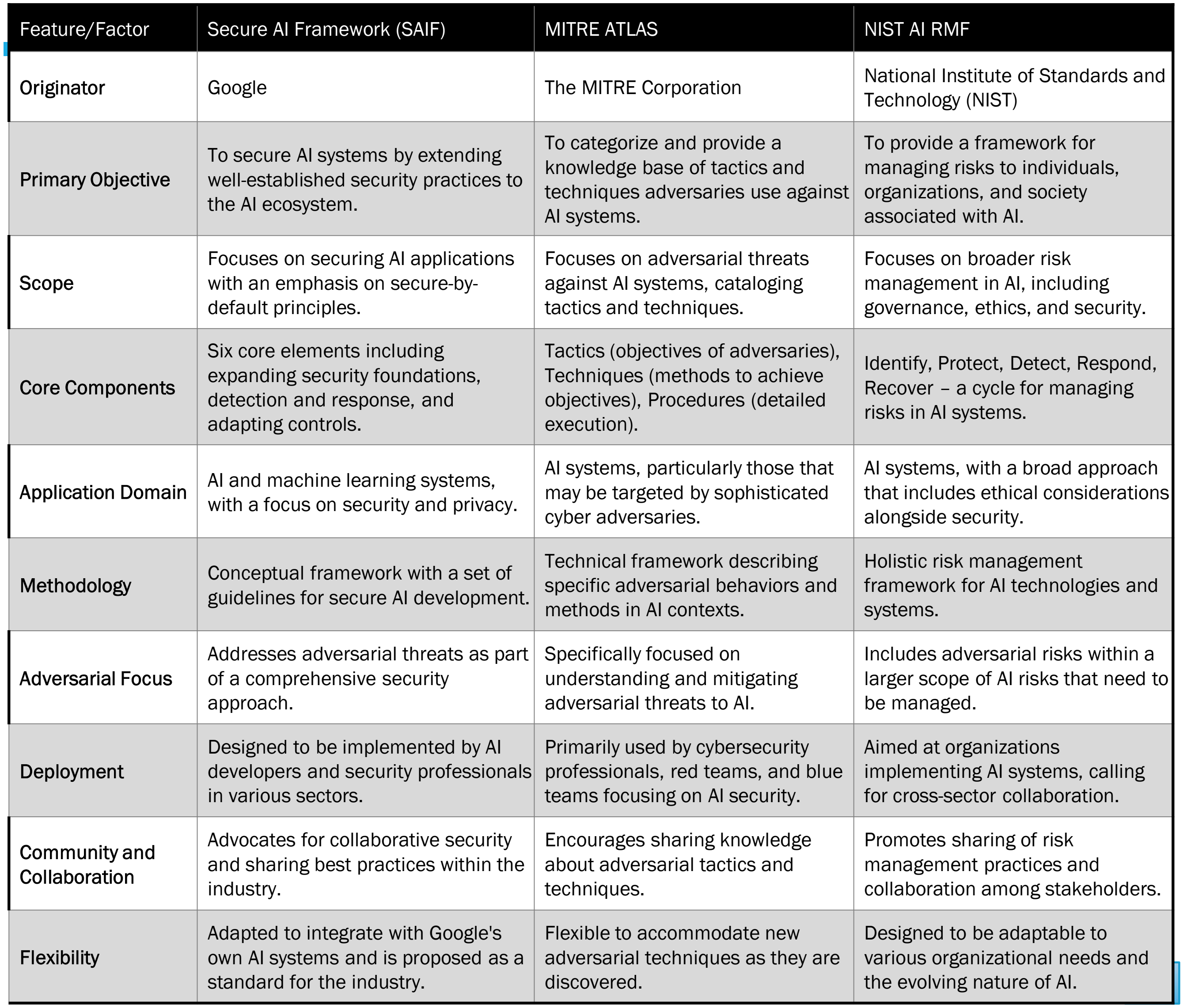
SECURE AI FRAMEWORK (SAIF), MITRE ATLAS AND NIST AI RMF COMPARISON

1. The SAIF focuses on leveraging existing cybersecurity mechanisms and applying them to the AI domain, aiming for secure-by-default AI systems.

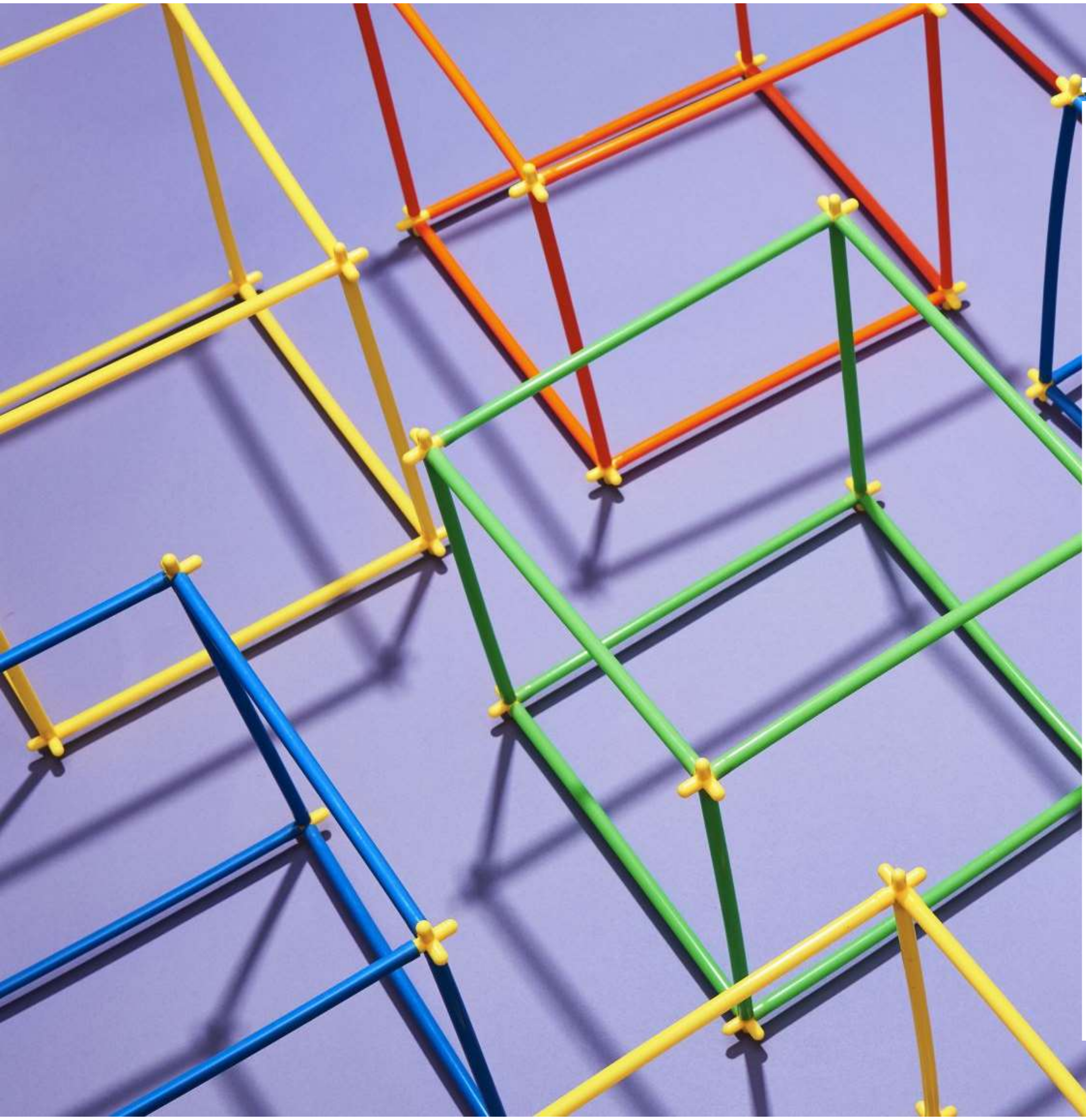
2. MITRE ATLAS is more focused on adversarial tactics and providing detailed techniques, sub techniques, mitigations and categories for cybersecurity experts handling AI threats.

3. The NIST AI RMF provides a comprehensive structure for managing risks in AI, encompassing more than just cybersecurity but also ethical, governance, and compliance aspects.

Each framework has its unique strengths and caters to different aspects of AI security and risk management.



Feature/Factor	Secure AI Framework (SAIF)	MITRE ATLAS	NIST AI RMF
Originator	Google	The MITRE Corporation	National Institute of Standards and Technology (NIST)
Primary Objective	To secure AI systems by extending well-established security practices to the AI ecosystem.	To categorize and provide a knowledge base of tactics and techniques adversaries use against AI systems.	To provide a framework for managing risks to individuals, organizations, and society associated with AI.
Scope	Focuses on securing AI applications with an emphasis on secure-by-default principles.	Focuses on adversarial threats against AI systems, cataloging tactics and techniques.	Focuses on broader risk management in AI, including governance, ethics, and security.
Core Components	Six core elements including expanding security foundations, detection and response, and adapting controls.	Tactics (objectives of adversaries), Techniques (methods to achieve objectives), Procedures (detailed execution).	Identify, Protect, Detect, Respond, Recover – a cycle for managing risks in AI systems.
Application Domain	AI and machine learning systems, with a focus on security and privacy.	AI systems, particularly those that may be targeted by sophisticated cyber adversaries.	AI systems, with a broad approach that includes ethical considerations alongside security.
Methodology	Conceptual framework with a set of guidelines for secure AI development.	Technical framework describing specific adversarial behaviors and methods in AI contexts.	Holistic risk management framework for AI technologies and systems.
Adversarial Focus	Addresses adversarial threats as part of a comprehensive security approach.	Specifically focused on understanding and mitigating adversarial threats to AI.	Includes adversarial risks within a larger scope of AI risks that need to be managed.
Deployment	Designed to be implemented by AI developers and security professionals in various sectors.	Primarily used by cybersecurity professionals, red teams, and blue teams focusing on AI security.	Aimed at organizations implementing AI systems, calling for cross-sector collaboration.
Community and Collaboration	Advocates for collaborative security and sharing best practices within the industry.	Encourages sharing knowledge about adversarial tactics and techniques.	Promotes sharing of risk management practices and collaboration among stakeholders.
Flexibility	Adapted to integrate with Google's own AI systems and is proposed as a standard for the industry.	Flexible to accommodate new adversarial techniques as they are discovered.	Designed to be adaptable to various organizational needs and the evolving nature of AI.



SECURE AI FRAMEWORK (SAIF) IS A CONCEPTUAL FRAMEWORK FOR SECURE ARTIFICIAL INTELLIGENCE (AI) SYSTEMS

- Google’s SAIF framework provides a standardized and holistic approach to integrating security and privacy measures into ML-powered applications and aligns with the 'Security' and 'Privacy' dimensions of building AI responsibly.
  - It ensures that AI models are developed considering the evolving threat landscape and user expectations.
- SAIF is inspired by security best practices such as reviewing, testing and controlling the supply chain
- Google has applied these best practices to software development, while incorporating our understanding of security mega-trends and risks specific to AI systems.

ADDRESSING THE CONCERNS

- The Secure AI Framework (SAIF) introduced by Google is a comprehensive set of guidelines and practices designed to ensure the secure deployment and operation of AI systems. It aims to mitigate risks specific to AI, such as model theft, training data poisoning, prompt injection attacks, and the extraction of confidential information from training data.

- SAIF offers a practical approach to address the concerns that are top of mind for security and risk professionals, such as:

1. Security
2. AI/ML model risk management
3. Privacy and compliance
4. People and organization

SECURITY CONCERNS

a) Access management

b) Network / endpoint security

c) Application / product security

d) Supply chain attacks

e) Data security

f) AI specific threats

g) Threat detection and response

AI/ML MODEL RISK MANAGEMENT

a) Model transparency and accountability

b) Error-prone manual reviews for detecting anomalies

c) Data poisoning

d) Data lineage, retention and governance controls

PRIVACY AND COMPLIANCE

A) DATA PRIVACY AND USAGE OF SENSITIVE DATA

B) EMERGING REGULATIONS

PEOPLE AND ORGANIZATION

A) TALENT GAP

B) GOVERNANCE / BOARD REPORTING

SIX CORE ELEMENTS OF SAIF

- Six core elements of SAIF are:
  1. Expand strong security foundations to the AI ecosystem
  2. Extend detection and response to bring AI into an organization's threat model
  3. Automate defenses to keep pace with existing and new threats
  4. Harmonize platform level controls to ensure consistent security across the organization
  5. Adapt controls to adjust mitigations and create faster feedback loops for AI deployment
  6. Contextualize AI system risks in surrounding business processes

SECURE AI FRAMEWORK (SAIF) AS OUTLINED BY GOOGLE, HIGHLIGHTING THE SIX CORE ELEMENTS

Element Number	Core Element of SAIF	Description
1	Expand strong security foundations to the AI ecosystem	Leverage existing infrastructural protections to secure AI systems. Enhance organizational expertise to keep pace with AI advances.
2	Extend detection and response	Include AI in the threat detection and response process. Monitor generative AI systems' inputs and outputs for anomalies.
3	Automate defenses	Use AI innovations to improve the scale and speed of security incident responses. Anticipate AI-utilizing adversaries and counteract accordingly.
4	Harmonize platform level controls	Ensure consistent security controls across the organization to protect all AI applications effectively.
5	Adapt controls to adjust mitigations	Implement faster feedback loops for AI deployment through continuous testing and learning from incidents and user feedback.
6	Contextualize AI system risks	Perform end-to-end risk assessments for AI deployments, considering business processes and automated performance validation checks.

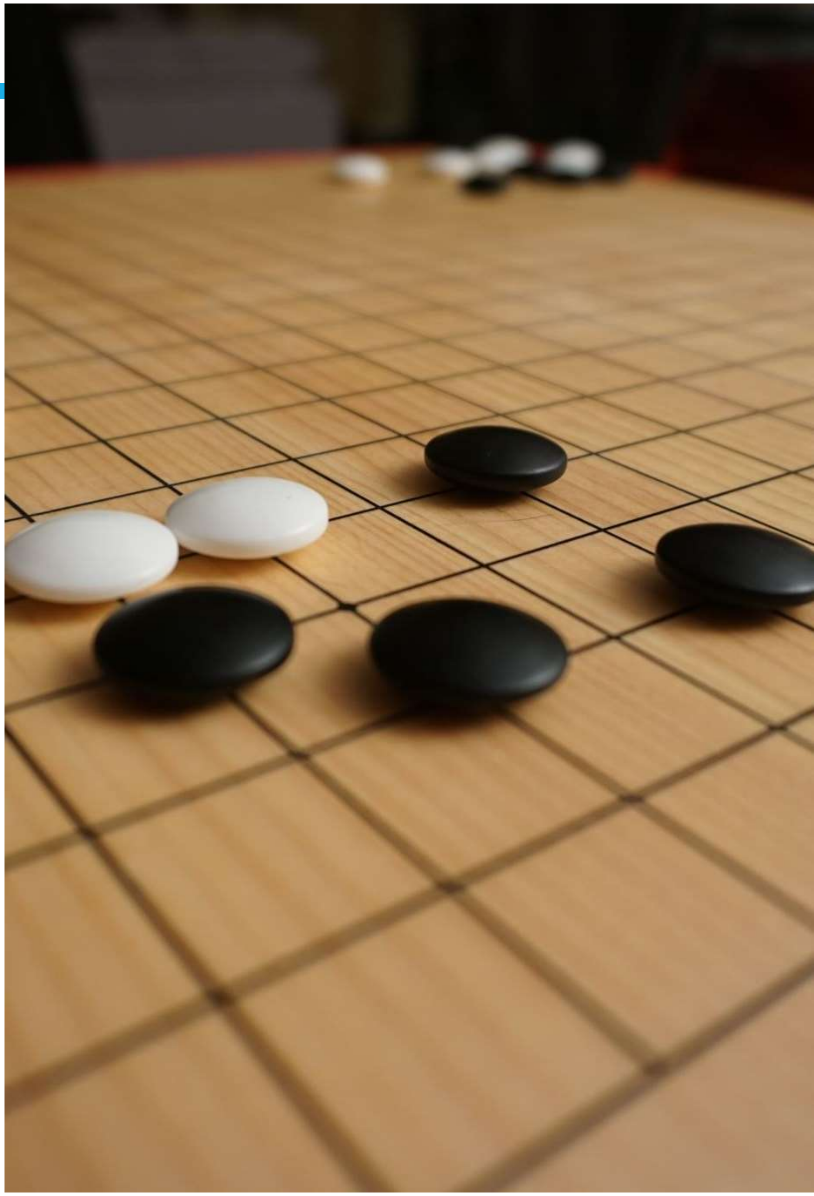
SAIF'S AI SYSTEM MITIGATIONS

- SAIF is designed to help mitigate risks specific to AI systems like stealing the model, data poisoning of the training data, injecting malicious inputs through prompt injection, and extracting confidential information in the training data.

- As AI capabilities become increasingly integrated into products across the world, adhering to a bold and responsible framework will be even more critical.

STEPS PUTTING SAIF INTO PRACTICE

- Step 1 - Understand the use
- Step 2 - Assemble the team
- Step 3 - Level set with an AI primer
- Step 4 - Apply the six core elements of SAIF



SAIF INDUSTRY ENGAGEMENTS

- NIST AI Risk Management Framework
  - Used to improve the ability to incorporate trustworthiness considerations into the design, development, use, and evaluation of AI products, services, and systems.

- ISO/IEC 42001 AI Management System Standard (the industry's first AI certification standard).
  - ISO/IEC 42001:2023 Information technology Artificial intelligence Management system
  - ISO/IEC 42001 is an international standard that specifies requirements for establishing, implementing, maintaining, and continually improving an Artificial Intelligence Management System (AIMS) within organizations. It is designed for entities providing or utilizing AI-based products or services, ensuring responsible development and use of AI systems.

- NIST Cybersecurity Framework and ISO/IEC 27001 Security Management System
  - Alongside the increase in security risks comes the corresponding laws and regulations necessary to keep your organization’s data safe. Two common safeguards today are ISO 27001 and NIST CSF.
  - ISO 27001 is an international standard to improve an organization’s information security management systems, while NIST CSF helps manage and reduce cybersecurity risks to their networks and data.
  - Both ISO 27001 and NIST CSF effectively contribute to a stronger security posture. However, the way they go about data protection is distinct to each framework.

THREAT MODELING EXAMPLE

- A code-generating LLM is trained on a more specialized dataset that includes code repositories, technical forums, coding platforms, documentation of various products and general web data that’s useful for this purpose. Because code-generating LLMs are integrated with your IDE, they fully grasp the context of your code (comments, function names, and variable names), and use this contextual information to further improve the suggestions they make. Assessing a third-party tool involves a thorough examination of its security, functionality, compliance, and overall performance. Here's a structured approach to evaluate such a tool:

- **1. Security Assessment**
  - **Vulnerability Scanning**: Conduct automated scans to identify known vulnerabilities in the tool.
  - **Penetration Testing**: Perform ethical hacking to test the tool’s defenses against simulated attacks.
  - **Code Review**: Analyze the source code for security weaknesses, such as hardcoded credentials or insecure libraries.
  - **Dependency Checking**: Examine third-party libraries and dependencies for vulnerabilities.
- **2. Functionality and Performance Testing**
  - **Unit Testing**: Test individual components for expected behavior.
  - **Integration Testing**: Ensure that different parts of the tool work together correctly.
  - **Performance Testing**: Evaluate the tool’s efficiency, speed, and resource consumption under different conditions.
  - **User Acceptance Testing (UAT)**: Verify the tool meets the end-users' requirements and expectations.
- **3. Compliance and Standards**
  - **Regulatory Compliance**: Check if the tool complies with relevant legal and regulatory requirements, such as GDPR for data protection.
  - **Industry Standards**: Assess alignment with industry standards like ISO/IEC 27001 for information security management.
- **5. User Feedback and Reviews**
  - **Customer Feedback**: Collect and analyze feedback from users who have interacted with the tool.
  - **Online Reviews and Ratings**: Check third-party websites and forums for reviews and discussions about the tool.
- **7. Continual Monitoring and Review**
  - **Monitoring Tools**: Implement monitoring tools to track the tool’s performance and usage over time.
  - **Regular Review Meetings**: Schedule periodic meetings to discuss the tool’s performance, issues, and improvement areas.

ASSESSMENT TIPS

- ## 1. Define Assessment Criteria
  - **Functionality**: Does the tool meet the operational requirements it was intended for?
  - **Performance**: How well does the tool perform under various conditions?
  - **Security**: What are the security measures in place, and how resilient is the tool against potential threats?
- ## 2. Conduct Technical Evaluation
  - **Code Review**: If access to the source code is available, conduct a thorough review to identify any potential security or performance issues.
  - **Penetration Testing**: Perform or commission penetration testing to identify vulnerabilities.
  - **Compatibility Testing**: Ensure the tool is compatible with your current systems and workflows.
- ## 3. Risk Management
  - **Risk Assessment**: Identify and evaluate any risks associated with using the tool, including dependency risks, support risks, and any operational risks.
  - **Mitigation Strategies**: Develop strategies to mitigate identified risks.
- ## 4. Security Assessment
  - **Vulnerability Assessment**: Use automated tools to scan for known vulnerabilities.
  - **Compliance Check**: Ensure the tool complies with relevant legal and regulatory requirements.
  - **Data Handling**: Review how the tool handles and protects sensitive data.
- ## 5. Obtain User Feedback
  - **Surveys and Interviews**: Gather feedback from end-users and stakeholders to assess the tool's effectiveness and user satisfaction.
  - **Usage Analytics**: Review usage data to understand how the tool is being used and its impact on operations.

HOW DO YOU PERFORM THREAT MODELING FOR A 3RD PARTY AI SOFTWARE

1. Define and Scope the 3rd party AI System: Clearly understand the functionality, components, and boundaries of the AI system. Determine what the AI system does, its inputs and outputs, the data it processes, and its interaction with other systems.

2. Identify and Prioritize 3rd party AI System Assets: Identify the assets that are valuable within the system, such as proprietary algorithms, sensitive data, and infrastructure. Prioritize these assets based on their importance and the impact of their compromise.

3. Create a Trust Model: Define which entities (users, other systems) can interact with the 3rd party AI system and to what extent. Establish trust levels for each entity to understand potential insider threats and the trust boundaries.

4. Identify Potential Threats: Use frameworks like ATLAS to identify potential threats to the system. For AI-specific concerns, consider threats like data poisoning, model theft, and adversarial attacks.

5. Assess Vulnerabilities and Risks: Evaluate the AI system and its components for vulnerabilities. Consider using tools for static and dynamic analysis and penetration testing. Assess the risk for each identified threat by considering the likelihood of occurrence and the potential impact.

6. Investigate with the Third-Party Software: Engage with the third-party AI software users to understand their security practices and controls.

7. Mitigate Risks: Develop strategies to mitigate or manage the identified risks. This could involve implementing security controls, designing robust algorithms resistant to attacks, data encryption, access controls, regular security audits, and incident response planning.

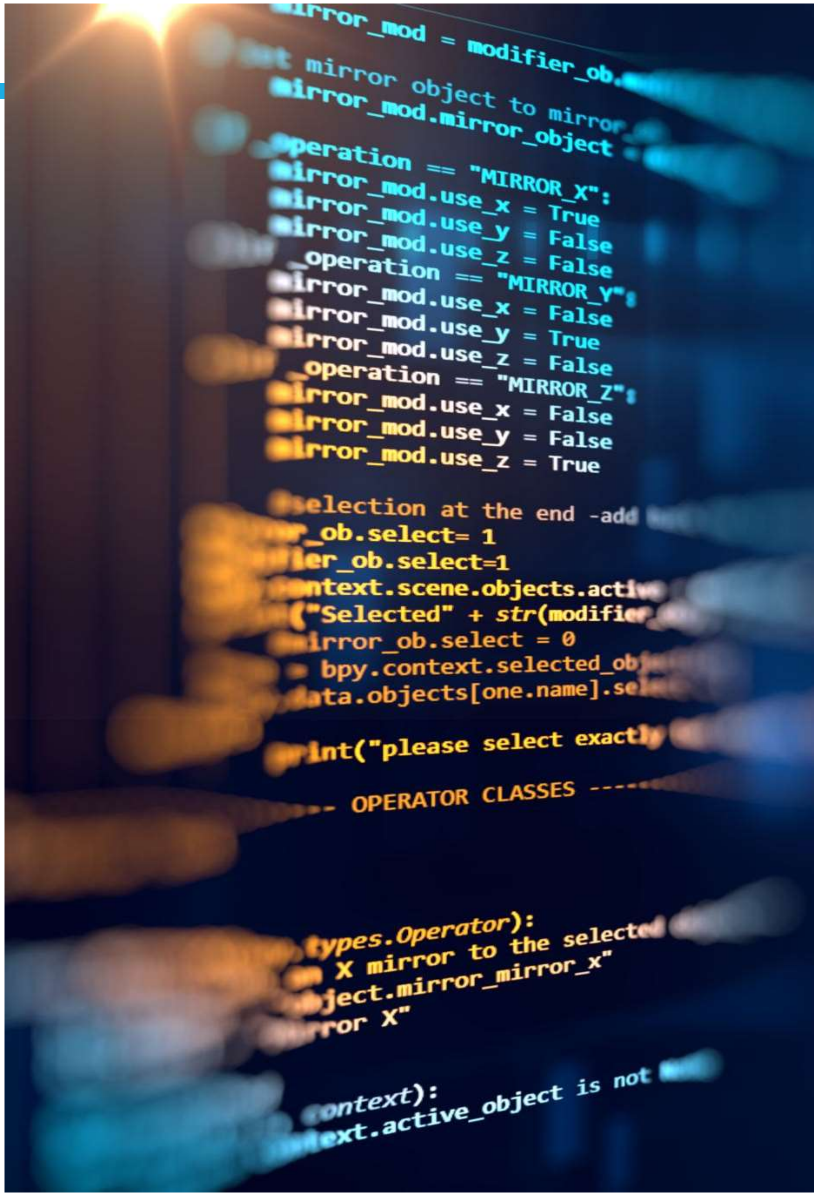
8. Document and Update: Document the threat model, including the identified threats, vulnerabilities, and mitigation strategies. Regularly update the threat model to reflect changes in the system, emerging threats, and technological advancements.

USE CASES FOR CODE-BASED GENERATIVE AI

- Generative AI used in helping and automating certain elements of software development, particularly in the form of generative adversarial networks (GANs) and language models like GPT-3/4:
- 1 Code-Generation
  - Code Generation: Developers can benefit from generative AI by proposing and finalizing code snippets based on the context of what they are currently coding. This has the potential to greatly accelerate the development process.
  - Automatic Code Completion: Natural language descriptions can be used to produce code by AI models. Developers can specify the functionality they require, and the AI will generate the necessary code.
- 2 Testing and Debugging:
  - Automated Testing: Generative AI can be used to autonomously produce test cases and scenarios, assisting in the testing phase of software development.
  - Bug Detection: AI models may examine code for possible bugs or dangers, making improvement suggestions and detecting errors early in the development cycle.
- 3 Natural Language Interfaces:
  - Conversational Interfaces: Non-technical individuals may be able to communicate with AI systems by describing the capabilities they require using natural language. The AI can then generate the necessary code without the user having to know how to program.
- 4 Design Prototyping:
  - UI/UX Prototyping: Based on high-level descriptions or wireframes provided by designers or product managers, AI may build basic user interface layouts and design prototypes.
- 5 Code Refactoring:
  - Automated Refactoring: Generative AI can aid in code refactoring by suggesting changes, maximizing efficiency, and adhering to coding standards.
- 6 Domain-Specific Code Generation:
  - AI for Specific Frameworks or Libraries: AI models can be trained on certain programming languages, frameworks, or libraries, allowing them to write code that adheres to those technologies’ conventions and best practices.

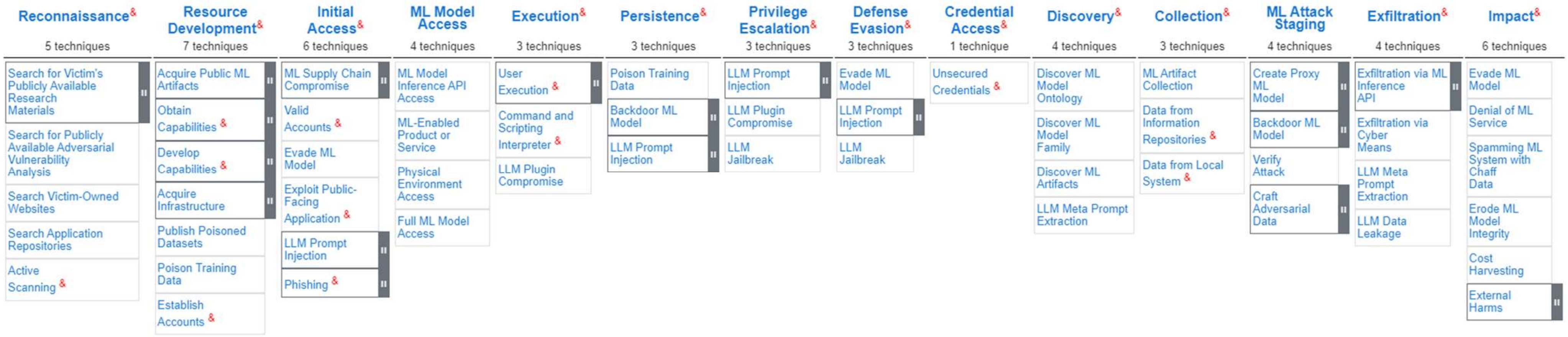
IDENTIFYING POTENTIAL THREATS

- Code Generation and Refactoring: Threats can include malicious code snippets being suggested by the AI, incorrect code generation leading to vulnerabilities, and improper refactoring.
- Testing and Debugging: False negatives in bug detection or the AI missing critical bugs can be a concern.
- Natural Language Interfaces: Injection attacks via conversational interfaces could manipulate the AI.
- UI/UX Prototyping: The generated prototypes may include vulnerabilities or expose sensitive data.



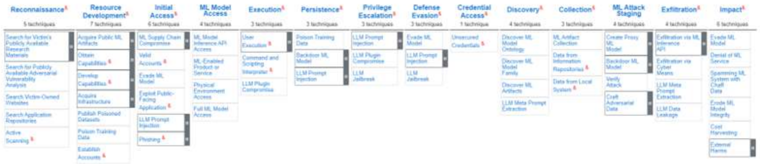
EXAMPLE OF MITRE ATLAS TACTICS AND TECHNIQUES USED

- **Initial Access**: Tactics like Phishing (T1566) to manipulate developers or AI interactions.
- **Execution**: Techniques such as Command and Scripting Interpreter (T1064) for executing unauthorized commands through AI-generated code.
- **Persistence**: Techniques like Create or Modify System Process (T1543) where AI-generated code could introduce persistent threats.
- **Privilege Escalation**: Techniques like Exploit Public-Facing Application (T1190) to escalate privileges through AI interactions.
- **Defense Evasion**: Techniques such as Obfuscated Files or Information (T1027) where generated code might include obfuscated malicious logic.
- **Credential Access**: Techniques like Brute Force (T1110) that could be suggested through AI-generated code snippets for authentication systems.
- **Discovery**: Techniques such as File and Directory Discovery (T1083) which could be facilitated by AI in understanding the environment.
- **Collection**: Techniques such as Data from Information Repositories (T1213) where AI could inadvertently collect and expose sensitive information.



Reconnaissance&	Resource Development&	Initial Access&	ML Model Access	Execution&	Persistence&	Privilege Escalation&	Defense Evasion&	Credential Access&	Discovery&	Collection&	ML Attack Staging	Exfiltration&	Impact&
5 techniques	7 techniques	6 techniques	4 techniques	3 techniques	3 techniques	3 techniques	3 techniques	1 technique	4 techniques	3 techniques	4 techniques	4 techniques	6 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution &	Poison Training Data	LLM Prompt Injection	Evade ML Model	Unsecured Credentials &	Discover ML Model Ontology	ML Artifact Collection	Create Proxy ML Model	Exfiltration via ML Inference API	Evade ML Model
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities &	Valid Accounts &	ML-Enabled Product or Service	Command and Scripting Interpreter &	Backdoor ML Model	LLM Plugin Compromise	LLM Prompt Injection		Discover ML Model Family	Data from Information Repositories &	Backdoor ML Model	Exfiltration via Cyber Means	Denial of ML Service
Search Victim-Owned Websites	Develop Capabilities &	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection	LLM Jailbreak	LLM Jailbreak		Discover ML Artifacts	Data from Local System &	Verify Attack	LLM Meta Prompt Extraction	Spamming ML System with Chaff Data
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application &	Full ML Model Access						LLM Meta Prompt Extraction		Craft Adversarial Data	LLM Data Leakage	Erode ML Model Integrity
Active Scanning &	Publish Poisoned Datasets	LLM Prompt Injection											Cost Harvesting
	Poison Training Data	Phishing &											External Harms
	Establish Accounts &												

STEPS TO ASSESS AN AI TOOL USING THE MITRE ATLAS FRAMEWORK



Reconnaissance	Resource Development	Initial Access	ML Model Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Collection	ML Attack Staging	Exfiltration	Impact
5 techniques	7 techniques	6 techniques	4 techniques	3 techniques	3 techniques	3 techniques	3 techniques	1 technique	4 techniques	3 techniques	4 techniques	4 techniques	6 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution	Poison Training Data	LLM Prompt Injection	Evade ML Model	Unsecured Credentials	Discover ML Model Ontology	ML Artifact Collection	Create Proxy ML Model	Exfiltration via ML Inference API	Evade ML Model
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities	Valid Accounts	ML-Enabled Product or Service	Command and Scripting Interpreter	Backdoor ML Model	LLM Plugin Compromise	LLM Prompt Injection		Discover ML Model Family	Data from Information Repositories	Backdoor ML Model	Exfiltration via Cyber Means	Denial of ML Service
Search Victim-Owned Websites	Develop Capabilities	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection	LLM Jailbreak	LLM Jailbreak		Discover ML Artifacts	Data from Local System	Verify Attack	LLM Meta Prompt Extraction	Spamming ML System with Chaff Data
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application	Full ML Model Access						LLM Meta Prompt Extraction		Craft Adversarial Data	LLM Data Leakage	Erode ML Model Integrity
Active Scanning	Publish Poisoned Datasets	LLM Prompt Injection											Cost Harvesting
	Poison Training Data	Phishing											External Harms
	Establish Accounts												

1. **Evaluate the Tool's Capabilities and Limitations**: Using the ATLAS techniques as a guide, examine the tool's capabilities to understand where it may be susceptible to the techniques adversaries could use. For instance, does the tool have a way to detect data poisoning or adversarial attacks?
2. **Develop Use Cases**: Create specific use cases or scenarios based on the ATLAS techniques to assess how the tool performs under conditions that simulate an attack or manipulation.
3. **Perform Threat Modeling**: Conduct a threat modeling exercise, incorporating ATLAS techniques to identify how an adversary could target the AI system.
4. **Map ATLAS Tactics to AI Tool Functions**: Identify which parts of the tool correspond to tactics described in the ATLAS framework. For instance, if the tool interacts with public data sources for training, map this to the "Collection" tactic.
5. **Identify Relevant ATLAS Techniques**: For each tactic, look at the associated techniques in ATLAS and determine which ones could apply to the tool's functions.
6. **Run Simulations and Drills**: Use the identified techniques to run simulations of potential adversarial attacks on the AI system to see how it responds and to identify vulnerabilities.
7. **Assess Risk Based on Findings**: Based on the results from the simulations and threat modeling, assess the level of risk associated with each technique and determine the likelihood and impact of exploitation.
8. **Mitigate Identified Risks**: Develop strategies to mitigate the risks identified. This could include improving data validation, enhancing model training procedures, or introducing anomaly detection systems.
9. **Document and Communicate Findings**: Ensure that findings, risk assessments, and mitigation strategies are documented clearly and communicated to relevant stakeholders within the organization.
10. **Create a Continuous Improvement Plan**: Develop a plan for continuous assessment and improvement based on the ATLAS framework to keep pace with evolving threats.

USING ATLAS FRAMEWORK SECURITY EVALUATION

1. **Analyze Adversarial Goals**: Understand the potential goals of adversaries targeting your AI system. ATLAS can help you identify why an adversary might target your system (e.g., to steal intellectual property, to disrupt operations, or to manipulate decision-making).
2. **Map AI System Components to ATLAS Tactics**: Identify which components of your third-party AI system could be affected by the tactics described in ATLAS. For example, consider how data poisoning (a technique under the tactic of "Training Data Manipulation") could impact your AI system's training datasets.
3. **Identify Relevant Techniques and Mitigations**: For each tactic, look at the corresponding techniques in ATLAS and determine which ones are relevant to your system. Assess how an adversary could exploit these techniques and what mitigations can be put in place to prevent or reduce the impact of such attacks.
4. **Prioritize Threats and Mitigations**: Based on the analysis, prioritize the threats that are most likely to occur and have the highest impact. Allocate resources to implement the necessary mitigations, focusing on high-priority threats first.
5. **Integrate with Risk Management**: Incorporate the findings from using the ATLAS framework into your overall risk management and threat modeling process. This should include regular updates and revisions as new threats are identified and as the AI system evolves.
6. **Collaboration and Reporting**: Engage to discuss the findings from the ATLAS analysis and collaborate on mitigation strategies. Reporting and documentation should include detailed accounts of the identified threats, analysis, and proposed mitigation strategies.

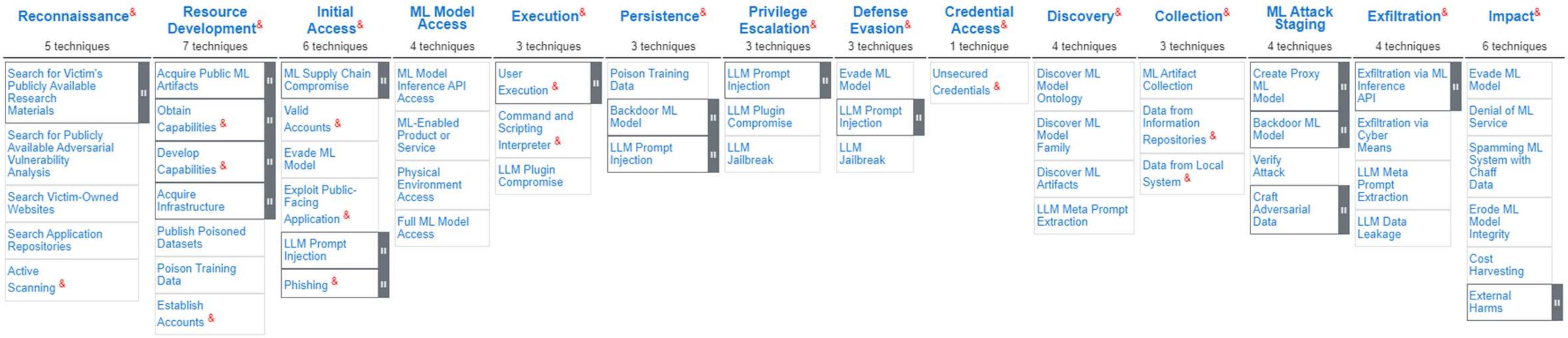
EXAMPLE THREAT MODEL FOR OPEN-SOURCE LLM/GENAI MODEL USAGE

- Step 1: Define and Scope
  - The AI system includes the LLM/GenAI model, development environment, deployment pipeline, and generated code repositories.
- Step 2: Identify and Prioritize Assets
  - High-priority assets include the LLM/GenAI model, the generated Java code, and the data used for training.
- Step 3: Create a Trust Model
  - Trust boundaries are established between internal development teams, the LLM/GenAI model, and third-party libraries.
- Step 4: Identify Potential Threats
  - Possible threats could include code manipulation, leaking sensitive information through generated code, or injecting malicious code via the LLM/GenAI model.
- Step 5: Assess Vulnerabilities and Risks
  - An attacker might use social engineering to gain access to the development environment or exploit vulnerabilities in the open-source components.
- Step 6: Mitigate Risks
  - Implement strong access controls, regular code reviews, automated security scanning of generated code, and monitor the usage of the LLM/GenAI model.
- Step 7: Document and Update
  - Maintain a threat intelligence platform to keep track of new threats and vulnerabilities, ensuring the threat model is current and actionable.

AN ATLAS-BASED THREAT MODEL FOR AN OPEN SOURCE LLM/GENAI MODELS USING GPT-4, DEEPSEEK CODER, CLAUDE 2, CODE LLAMA, GEMINI PRO TO GENERATE JAVA CODE

- **1. Define and Scope the AI System**
  - **Objective**: Understand how the LLM/GenAI model is used within the organization to generate Java code.
  - **ATLAS Tactics**: Reconnaissance, Resource Development.
- **2. Identify and Prioritize Assets**
  - **Objective**: Determine the critical components like proprietary algorithms, training data, generated code repositories, and deployment mechanisms.
  - **ATLAS Tactics**: Resource Development, Initial Access.
- **3. Create a Trust Model**
  - **Objective**: Establish which entities are trusted within the system and their level of access.
  - **ATLAS Tactics**: Initial Access, Persistence.
- **4. Identify Potential Threats**
  - **Objective**: Enumerate threats such as unauthorized access, model theft, data poisoning, adversarial attacks, and misuse of generated code.
  - **ATLAS Tactics**: Execution, Persistence, Privilege Escalation, ML Model Access
- **5. Assess Vulnerabilities and Risks**
  - **Objective**: Evaluate how an attacker could exploit these threats and the potential impact.
  - **ATLAS Tactics**: Credential Access, Discovery, ML Attack Staging.
- **6. Mitigate Risks**
  - **Objective**: Develop strategies to mitigate identified risks.
  - **ATLAS Tactics**: Collection, Exfiltration, ML Attack Staging
- **7. Document and Update**
  - **Objective**: Keep a detailed account of the threat model and update it regularly.
  - **ATLAS Tactics**: Impact, Inhibiting Response.

USING ATLAS FRAMEWORK FOR EVALUATION OF CONTINUE.DEV

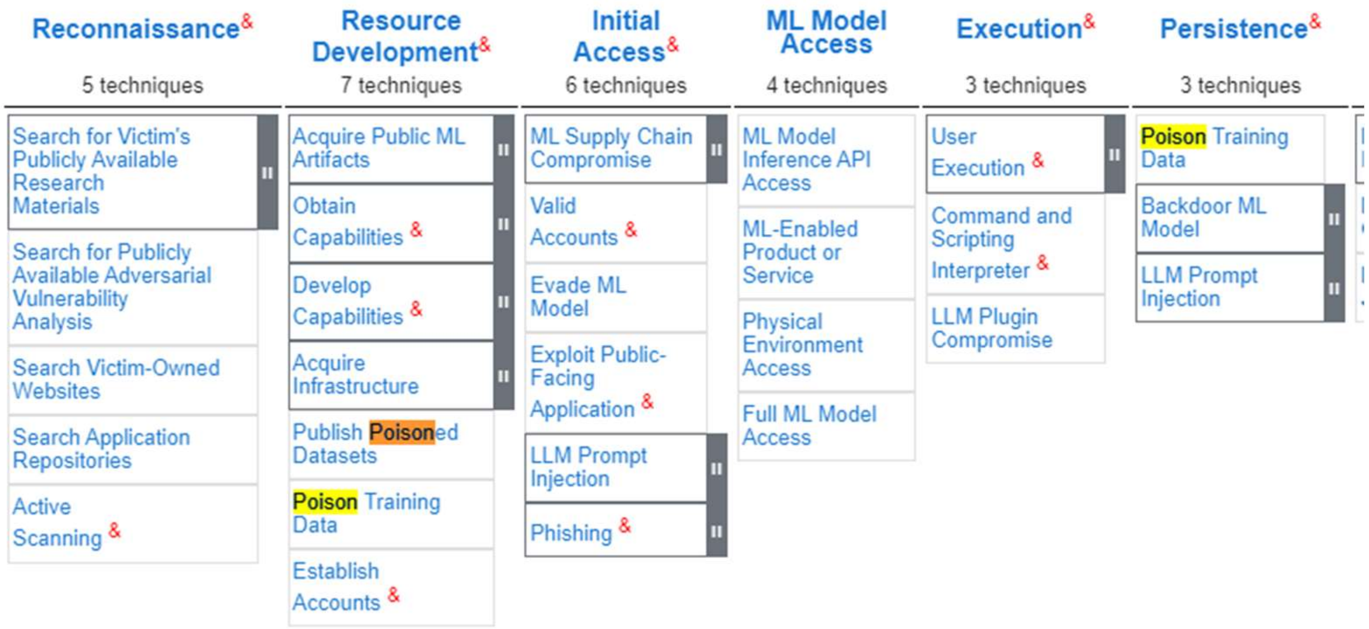


Reconnaissance&	Resource Development&	Initial Access&	ML Model Access	Execution&	Persistence&	Privilege Escalation&	Defense Evasion&	Credential Access&	Discovery&	Collection&	ML Attack Staging	Exfiltration&	Impact&
5 techniques	7 techniques	6 techniques	4 techniques	3 techniques	3 techniques	3 techniques	3 techniques	1 technique	4 techniques	3 techniques	4 techniques	4 techniques	6 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution &	Poison Training Data	LLM Prompt Injection	Evade ML Model	Unsecured Credentials &	Discover ML Model Ontology	ML Artifact Collection	Create Proxy ML Model	Exfiltration via ML Inference API	Evade ML Model
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities &	Valid Accounts &	ML-Enabled Product or Service	Command and Scripting Interpreter &	Backdoor ML Model	LLM Plugin Compromise	LLM Prompt Injection		Discover ML Model Family	Data from Information Repositories &	Backdoor ML Model	Exfiltration via Cyber Means	Denial of ML Service
Search Victim-Owned Websites	Develop Capabilities &	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection	LLM Jailbreak	LLM Jailbreak		Discover ML Artifacts	Data from Local System &	Verify Attack	LLM Meta Prompt Extraction	Spamming ML System with Chaff Data
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application &	Full ML Model Access						LLM Meta Prompt Extraction		Craft Adversarial Data	LLM Data Leakage	Erode ML Model Integrity
Active Scanning &	Publish Poisoned Datasets	LLM Prompt Injection											Cost Harvesting
	Poison Training Data	Phishing &											External Harms
	Establish Accounts &												

EXAMPLE OF MAPPING CONTINUE.DEV COMPONENTS TO ATLAS TACTICS

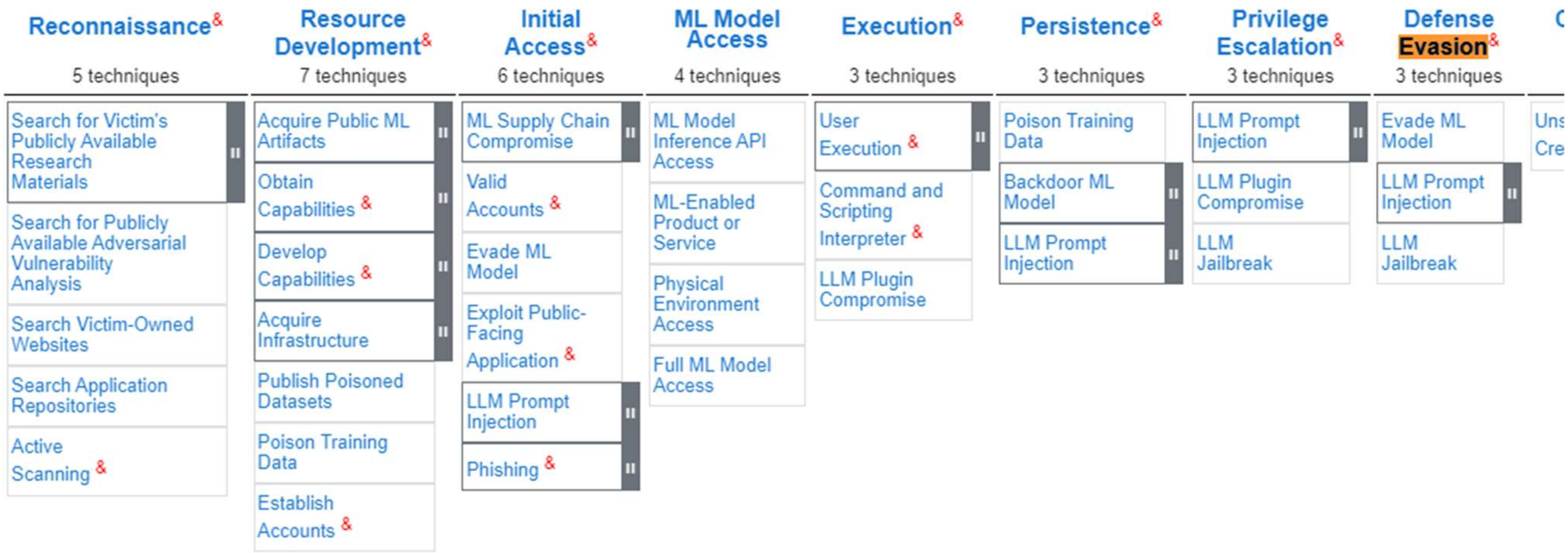
- **1. Data Collection**
  - **ATLAS : Initial Access**
  - **Rationale**: In the initial access phase, adversaries might seek to gain entry into the system to manipulate or steal the data collected for training the AI model. For example, they could introduce misleading data to skew fraud detection results.
  - **Technique Example**: Adversaries might use phishing attacks to gain access to the data collection infrastructure, allowing them to insert or alter data.
- **2. Data Storage and Management**
  - **ATLAS : Data Manipulation**
  - **Rationale**: Once data is stored, adversaries might manipulate it to affect the training process. By corrupting stored data, they can influence the model's behavior.
  - **Technique Example**: Database injection attacks to alter or corrupt the training data.
- **3. Model Training**
  - **ATLAS : Model Poisoning**
  - **Rationale**: During model training, adversaries might introduce subtle manipulations to the training data or process, aiming to poison the model. This could lead to incorrect or biased fraud detection outcomes.
  - **Technique Example**: Injection of carefully crafted adversarial examples into the training dataset to mislead the model.
- **4. Model Deployment**
  - **ATLAS : Defense Evasion**
  - **Rationale**: Once the model is deployed, adversaries might attempt to evade detection by crafting input data in a way that is wrongly classified by the AI system.
  - **Technique Example**: Adversaries constructing financial transactions that are structured to avoid triggering fraud alerts by the deployed model.
- **5. Model Monitoring and Maintenance**
  - **ATLAS : Impact**
  - **Rationale**: Adversaries may seek to disrupt the AI system’s operation or degrade its performance, affecting the integrity of ongoing monitoring and maintenance.
  - **Technique Example**: DDoS attacks on the model's serving infrastructure to reduce availability or tampering with model update mechanisms to degrade performance over time.

USING ATLAS FRAMEWORK FOR EVALUTAION OF CONTINUE.DEV



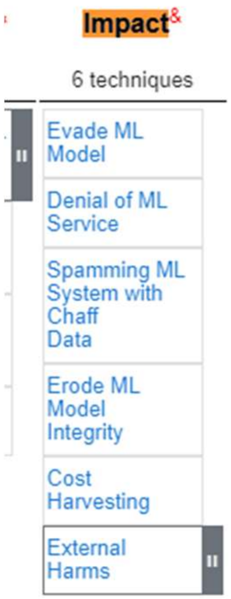
Reconnaissance& 5 techniques	Resource Development& 7 techniques	Initial Access& 6 techniques	ML Model Access 4 techniques	Execution& 3 techniques	Persistence& 3 techniques
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution &	Poison Training Data
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities &	Valid Accounts &	ML-Enabled Product or Service	Command and Scripting Interpreter &	Backdoor ML Model
Search Victim-Owned Websites	Develop Capabilities &	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application &	Full ML Model Access		
Active Scanning &	Publish Poisoned Datasets	LLM Prompt Injection			
	Poison Training Data	Phishing &			
	Establish Accounts &				

USING ATLAS FRAMEWORK FOR EVALUTAION OF CONTINUE.DEV



Reconnaissance& 5 techniques	Resource Development& 7 techniques	Initial Access& 6 techniques	ML Model Access 4 techniques	Execution& 3 techniques	Persistence& 3 techniques	Privilege Escalation& 3 techniques	Defense Evasion& 3 techniques	C
Search for Victim's Publicly Available Research Materials	Acquire Public ML Artifacts	ML Supply Chain Compromise	ML Model Inference API Access	User Execution &	Poison Training Data	LLM Prompt Injection	Evade ML Model	Uns Cre
Search for Publicly Available Adversarial Vulnerability Analysis	Obtain Capabilities &	Valid Accounts &	ML-Enabled Product or Service	Command and Scripting Interpreter &	Backdoor ML Model	LLM Plugin Compromise	LLM Prompt Injection	
Search Victim-Owned Websites	Develop Capabilities &	Evade ML Model	Physical Environment Access	LLM Plugin Compromise	LLM Prompt Injection	LLM Jailbreak	LLM Jailbreak	
Search Application Repositories	Acquire Infrastructure	Exploit Public-Facing Application &	Full ML Model Access					
Active Scanning &	Publish Poisoned Datasets	LLM Prompt Injection						
	Poison Training Data	Phishing &						
	Establish Accounts &							

USING ATLAS FRAMEWORK FOR EVALUTAION OF CONTINUE.DEV



Impact

6 techniques

- Evade ML Model
- Denial of ML Service
- Spamming ML System with Chaff Data
- Erode ML Model Integrity
- Cost Harvesting
- External Harms

SECURITY ASSESSMENT OF LARGE LANGUAGE MODELS (LLMS) AND GENERAL AI FOR SOFTWARE CODE GENERATION

Security assessment of Large Language Models (LLMs) and General AI for software code generation involves identifying potential threats and implementing mitigation strategies.

- Security assessment of Large Language Models (LLMs) and General AI for software code generation involves identifying potential threats and implementing mitigation strategies. These can be mapped to specific MITRE ATLAS tactics and techniques, along with corresponding mitigations.
- **1. Threat: Code Injection and Execution**
  - **MITRE ATLAS Tactic**: Execution
  - **Technique**: Command and Scripting Interpreter
  - **Mitigation**: Implement input validation, conduct regular code reviews, and use sandbox environments to test generated code before deployment.
- **2. Threat: Data Poisoning**
  - **MITRE ATLAS Tactic**: Initial Access
  - **Technique**: Supply Chain Compromise
  - **Mitigation**: Validate and sanitize training data, monitor for anomalies in data sources, and secure the data supply chain to prevent tampering.
- **3. Threat: Model Stealing or Inversion**
  - **MITRE ATLAS Tactic**: Collection
  - **Technique**: Data from Information Repositories
  - **Mitigation**: Use techniques like differential privacy, rate limiting API calls, and encrypting model outputs to protect against unauthorized model access or inference.
- **4. Threat: Adversarial Attacks**
  - **MITRE ATLAS Tactic**: Impact
  - **Technique**: Manipulate Model Behavior
  - **Mitigation**: Regularly test models against adversarial examples, update models with adversarial training, and deploy anomaly detection systems.

SECURITY ASSESSMENT OF LARGE LANGUAGE MODELS (LLMS) AND GENERAL AI FOR SOFTWARE CODE GENERATION

- **5. Threat: Unauthorized Access**
  - **MITRE ATLAS Tactic**: Credential Access
  - **Technique**: Brute Force
  - **Mitigation**: Implement strong authentication mechanisms, regular password audits, and multi-factor authentication (MFA) to secure access to the AI system.
- **6. Threat: Misinformation and Biased Outputs**
  - **MITRE ATLAS Tactic**: Influence Operation
  - **Technique**: Spread Misinformation
  - **Mitigation**: Monitor and validate the AI’s outputs, implement fairness checks, and ensure diversity in training datasets to reduce bias.
  - **Mapping to MITRE ATLAS for Mitigation**

1. **Regular Security Assessments**: Conduct vulnerability scans and penetration tests to identify and remediate potential security issues.
   - **ATLAS Tactic**: Discovery
   - **Technique**: System Network Configuration Discovery
2. **Secure Development Lifecycle (SDLC)**: Integrate security practices throughout the development and deployment phases of the AI software.
   - **ATLAS Tactic**: Defense Evasion
   - **Technique**: Obfuscated Files or Information
3. **Incident Response and Recovery Plans**: Develop and implement incident response strategies to quickly address and mitigate any security breaches or issues.
   - **ATLAS Tactic**: Response
   - **Technique**: Incident Response Process
4. **User Training and Awareness**: Educate developers and users about potential security threats and safe practices when using AI systems.
   - **ATLAS Tactic**: Resource Development
   - **Technique**: Establish Accounts

INTEGRATING AI-BASED CODE GENERATION INTO AN ONLINE WEDDING MATCHING PORTAL

- **1. AI Component: Code Generation for Matchmaking Algorithms**
  - **ATLAS Tactic: Model Poisoning**
  - **Threat**: Adversaries could attempt to manipulate the matchmaking algorithm to produce biased or undesirable matches.
  - **Mitigation**: Implement robust data validation and anomaly detection to identify and filter out malicious inputs during the training phase. Regularly audit and validate the model's output against expected patterns and historical data.
- **2. AI Component: User Data Processing**
  - **ATLAS Tactic: Data Manipulation**
  - **Threat**: An attacker might try to alter user data to influence match results or steal sensitive information.
  - **Mitigation**: Use encryption for data at rest and in transit, employ access controls to limit data exposure, and implement data integrity checks to ensure that user data has not been tampered with.
- **3. AI Component: Automated Content Generation**
  - **ATLAS Tactic: Evasion**
  - **Threat**: Adversaries may craft input data to evade detection or filtering mechanisms, generating inappropriate or harmful content.
  - **Mitigation**: Develop and regularly update content filtering algorithms to detect and prevent the generation of inappropriate content. Implement rate limiting and behavioral analysis to identify and block suspicious activities.
- **4. AI Component: User Interaction and Feedback Analysis**
  - **ATLAS Tactic: Impact**
  - **Threat**: Threat actors could aim to disrupt the service or manipulate feedback mechanisms to degrade the system's effectiveness or reputation.
  - **Mitigation**: Establish monitoring and alerting for unusual patterns of user interaction that could indicate a coordinated attack or system malfunction. Use feedback analysis with manual oversight to detect and correct abnormal behavior patterns.

IMPLEMENTING THE MITIGATIONS

- **Continuous Monitoring**: Implement continuous monitoring of the AI components to detect and respond to anomalies in real-time, using security information and event management (SIEM) systems.
- **Regular Security Assessments**: Conduct regular security assessments, including penetration testing and vulnerability scanning, to identify and mitigate new threats.
- **User Education and Awareness**: Educate users about potential security threats and encourage secure practices, such as using strong, unique passwords and recognizing phishing attempts.
- **Collaboration with Security Researchers**: Engage with the security community to stay informed about new threats and vulnerabilities and obtain insights into advanced mitigation strategies.

BUILD TRUST WITH AI-BASED CODE GENERATION FOR A MARRIAGE PORTAL

Focus Area	Strategy
Transparency	Provide clear explanations of AI processes, data usage, and decision-making logic.
Accuracy & Reliability	Test and validate AI-generated code to ensure it meets quality standards; employ code reviews and testing.
Security & Privacy	Implement strong security measures, comply with privacy laws, and encrypt sensitive data.
Ethical Considerations	Adhere to ethical guidelines, ensure non-discriminatory practices, and maintain transparency.
User Control & Feedback	Offer users control over AI interactions and incorporate their feedback to improve the system.
Continuous Improvement	Use performance data and user feedback for ongoing refinement and enhancement of the AI system.
Stakeholder Engagement	Involve users, developers, and experts in the AI system’s development and evaluation process.
Regulatory Compliance	Ensure the AI system adheres to applicable laws and industry regulations.
Education & Training	Provide training to stakeholders about the AI’s functionality, benefits, and risks.

MAPPING THE TRUST-BUILDING STRATEGIES FOR AI-BASED CODE GENERATION IN A MARRIAGE PORTAL TO SPECIFIC MITRE ATLAS TACTICS AND TECHNIQUES

Strategy	ATLAS Tactic	ATLAS Technique	Description
Transparency	Reconnaissance	Collect System Information	Being open about how the AI system operates can deter malicious actors by demonstrating system knowledge and preparedness.
Accuracy & Reliability	Model Poisoning	Modify Model	Regular testing and validation can prevent the incorporation of biased or malicious inputs that could poison the model.
Security & Privacy	Defense Evasion	Obfuscate Data/Information	Implementing encryption and secure data practices makes it harder for attackers to access or decipher valuable information.
Ethical Considerations	Impact	Manipulate System Resources	Adhering to ethical guidelines helps prevent exploitation of the system's resources for unethical purposes.
User Control & Feedback	Execution	User Execution	Allowing user control and feedback can identify and rectify unintended AI actions or biases, preventing misuse or exploitation of AI capabilities.
Continuous Improvement	Persistence	Establish Foothold	Continual updates and improvements make the system more resilient to attacks and ensure long-term defense mechanisms are in place.
Stakeholder Engagement	Discovery	Understand Operational Patterns	Involving stakeholders in the AI process helps in understanding and anticipating potential threats, leading to better preparedness and response strategies.
Regulatory Compliance	Credential Access	Steal or Forge Kerberos Tickets (Golden Ticket)	Compliance with legal standards ensures that the system maintains secure authentication and access controls, preventing unauthorized access.
Education & Training	Resource Development	Develop Capabilities	Training stakeholders increases awareness and capability to identify and mitigate AI-specific threats effectively.

MAPPING AI-BASED LARGE LANGUAGE MODELS (LLMS) AND GENERAL AI CODE GENERATION TO MITRE ATLAS

AI Component	ATLAS Tactic	ATLAS Technique	Description
Code Generation Process	Execution	Malicious Code Execution	Threat actors might exploit vulnerabilities in the code generation process to execute malicious code.
Data Input for Training	Initial Access	Supply Chain Compromise	Adversaries could compromise the supply chain to poison the training data, affecting the AI's output.
Model Training Environment	Persistence	Establish Foothold	Attackers may try to establish a persistent presence in the model training environment to manipulate outcomes.
Output Validation	Defense Evasion	Modify System Properties	Threat actors might bypass output validation mechanisms to inject malicious code or outputs into the system.
Deployment Infrastructure	Impact	Service Stop	Adversaries could target the deployment infrastructure to disrupt the availability of the AI service.
User Interaction with AI	Privilege Escalation	Exploit Public-Facing Application	Attackers may exploit vulnerabilities in applications interfacing with the AI to escalate privileges.
Feedback Loop for Improvement	Discovery	System Network Configuration Discovery	Malicious actors could exploit the feedback loop to discover network configurations and vulnerabilities.

EXAMPLE OF BLUE TEAM/RED TEAM ACTIVITIES ON AI SECURITY EVALUATION

- **Blue Team (Defensive) for GenAI/LLM:**

1. **Secure Development Lifecycle**: Integrate security into the AI development process, ensuring that secure coding practices and code reviews are a standard part of the lifecycle.
2. **Data Protection**: Enforce strict controls over the datasets used for training the AI models to prevent sensitive information leakage and ensure data privacy.
3. **Model Robustness Testing**: Regularly test AI models against various attacks, including adversarial examples that could mislead the AI's decision-making processes.
4. **Access Control**: Implement robust authentication and authorization mechanisms to control access to AI models and their APIs.
5. **Monitoring and Anomaly Detection**: Continuously monitor the AI model's operations for signs of tampering or abnormal behavior, and have automated responses in place.
6. **Incident Response**: Develop an AI-specific incident response plan, which includes scenarios like model corruption, data poisoning, or unexpected model behavior.

- **Red Team (Offensive) for GenAI/LLM:**

1. **Penetration Testing**: Probe the AI system’s infrastructure for vulnerabilities that could be exploited, including the APIs and endpoints that interact with the AI model.
2. **Adversarial ML Attacks**: Craft adversarial inputs to challenge the model's ability to make accurate predictions, aiming to exploit weaknesses in the model’s processing.
3. **Social Engineering**: Test the human elements involved in operating and interacting with the AI system to assess the risk of phishing or other social-based attacks.
4. **Supply Chain Analysis**: Investigate third-party plugins, libraries, and data sources used in the AI model for vulnerabilities or malicious code that could compromise the system.
5. **Model Theft and Intellectual Property Risks**: Attempt to reverse-engineer or extract proprietary AI models to assess the risk of intellectual property theft.

LLM/GENAI BLUE TEAM RED TEAM ACTIVITIES

Team	Activities	Objectives
Blue Team (Defensive)	Monitor AI system performance for anomalies.	Ensure the integrity and availability of AI services.
	Conduct regular security audits and compliance checks.	Maintain security standards and regulatory compliance.
	Implement access controls and authentication measures.	Prevent unauthorized access to AI systems.
	Develop and test incident response plans.	Prepare for and mitigate the impact of security incidents.
	Employ encryption and data masking techniques.	Protect the confidentiality of data used by AI systems.
	Update AI models and systems with security patches.	Address vulnerabilities and strengthen AI system security.
	Educate users and developers about AI security.	Raise awareness and reduce the risk of human error.
Red Team (Offensive)	Simulate adversarial attacks on AI systems.	Identify vulnerabilities before a real attacker does.
	Attempt to bypass security controls and gain access.	Test the effectiveness of security measures in place.
	Conduct social engineering campaigns.	Assess the human factor in AI system security.
	Test the robustness of AI systems against data poisoning and adversarial inputs.	Evaluate how AI systems handle malicious or manipulated data.
	Try to exfiltrate data or disrupt AI services.	Expose potential consequences of successful breaches.
	Provide feedback and recommendations for improvements.	Help the Blue Team improve defensive strategies.

LLM/GENAI VULNERABILITIES CAN BE ADDRESSED THROUGH RED AND BLUE TEAM ACTIVITIES AND MAPPED TO OWASP TOP 10 FOR LLM APPLICATIONS

OWASP LLM Top 10 Vulnerability	Blue Team Activity	Red Team Activity
LLM01: Prompt Injection	Implement input validation, monitoring, and anomaly detection to prevent unauthorized model manipulation.	Simulate prompt injection attacks to find ways the LLM can be manipulated.
LLM02: Insecure Output Handling	Harden output handling with encoding and sanitation practices to prevent exploitation like XSS or RCE.	Test for vulnerabilities by exploiting the output handling processes.
LLM03: Training Data Poisoning	Perform data source verification and apply robust data sanitization to maintain model integrity.	Attempt to inject biased or harmful content to test model resilience.
LLM04: Model Denial of Service	Ensure efficient resource management and scaling to maintain service availability.	Stress test the system resources to evaluate the service's capacity and robustness.
LLM05: Supply Chain Vulnerabilities	Use trusted datasets, models, and plugins. Maintain attestations via ML-BOM.	Probe for weak points in the supply chain to identify potential entry points.
LLM06: Sensitive Information Disclosure	Implement data leakage prevention mechanisms and strict data handling policies.	Seek to reveal confidential data through crafted queries to the LLM.
LLM07: Insecure Plugin Design	Apply secure design principles to LLM plugins, including strong access controls.	Target plugin vulnerabilities to assess their potential for exploitation.
LLM08: Excessive Agency	Limit the functionality, permissions, or autonomy granted to the LLM-based systems.	Test the bounds of the system's permissions and functionalities.
LLM09: Overreliance	Promote human oversight and validation in the use of LLMs to avoid misjudgments.	Test the system’s dependency on the LLM by presenting complex tasks.
LLM10: Model Theft	Protect proprietary LLM models with robust access control and monitoring.	Attempt to access, copy, or exfiltrate proprietary LLM models.

LLM/GENAI VULNERABILITIES CAN BE ADDRESSED THROUGH RED AND BLUE TEAM ACTIVITIES AND MAPPED TO ATLAS

Blue Team Activity	Red Team Activity	MITRE ATLAS Tactics
Implement input validation, regular expressions, and behavior monitoring for anomaly detection.	Craft and inject prompts to attempt to control or influence the LLM's output.	Initial Access
Sanitize and encode LLM outputs, use output handling libraries that have security measures.	Inject scripts or commands into LLM to see if it can trigger unexpected actions in downstream systems.	Execution
Regularly audit and verify the integrity of training data, employ robust data sanitization protocols.	Attempt to introduce biased, incorrect, or malicious data into the training set to test detection capabilities.	Persistence
Deploy rate-limiting and resource quotas, monitor for unusual levels of resource utilization.	Execute operations that are resource-intensive to test the system's ability to handle high loads.	Impact
Implement a secure software supply chain management process, including the use of digital signatures and integrity checks.	Seek to compromise the supply chain by introducing vulnerabilities or malicious payloads.	Initial Access
Apply data masking and access control measures, and use differential privacy where appropriate.	Craft queries that could potentially lead to the disclosure of sensitive information from the LLM.	Collection
Apply the principle of least privilege to plugin functionalities and conduct security assessments on third-party plugins.	Exploit vulnerabilities in plugin design to achieve privilege escalation or unauthorized access.	Privilege Escalation
Limit LLM functionality and permissions, employ strict role-based access controls.	Test the LLM's capabilities to identify any functions that can be exploited beyond their intended use.	Defense Evasion
Conduct thorough testing and validation, ensure human-in-the-loop checkpoints for critical decisions.	Analyze decision points where the system relies heavily on LLM and evaluate the potential impact.	Decision Support
Secure model storage and APIs with authentication, encryption, and logging.	Attempt to exfiltrate AI models to evaluate the effectiveness of protective measures.	Exfiltration

THIRD-PARTY SOFTWARE: RED TEAM TESTING USING THE MITRE ATLAS FRAMEWORK

- **Red Team (Offensive)**
- 1. Reconnaissance
  - **ATLAS Tactic & Technique**: Collect System Information
  - **Objective**: Gather as much information as possible about the third-party software, including its architecture, data flow, and external dependencies.
- 2. Initial Access
  - **ATLAS Tactic & Technique**: Supply Chain Compromise
  - **Objective**: Attempt to exploit vulnerabilities in the software supply chain to gain initial access to the system.
- 3. Execution
  - **ATLAS Tactic & Technique**: Command and Scripting Interpreter
  - **Objective**: Execute arbitrary commands or scripts to probe further into the system's defenses.
- 4. Persistence
  - **ATLAS Tactic & Technique**: Create or Modify System Process
  - **Objective**: Establish a way to maintain access within the third-party software for continued exploitation.
- 5. Privilege Escalation
  - **ATLAS Tactic & Technique**: Exploit Public-Facing Application
  - **Objective**: Elevate privileges to gain more control over the system and access sensitive information.

THIRD-PARTY SOFTWARE: BLUE TEAM TESTING USING THE MITRE ATLAS FRAMEWORK

- **Blue Team (Defensive)**
- 1. Detection and Monitoring
  - **ATLAS Tactic & Technique**: Network Traffic Analysis
  - **Objective**: Continuously monitor network traffic for unusual patterns that could indicate a breach or attempted breach.
- 2. Incident Response
  - **ATLAS Tactic & Technique**: Incident Response Process
  - **Objective**: Develop and refine incident response protocols to quickly and effectively respond to any identified threats.
- 3. System Hardening
  - **ATLAS Tactic & Technique**: Update Software
  - **Objective**: Regularly update and patch the third-party software to fix known vulnerabilities and reduce attack surfaces.
- 4. User Training
  - **ATLAS Tactic & Technique**: Security Awareness Training
  - **Objective**: Educate users on the potential security threats and safe practices to prevent social engineering and phishing attacks.
- 5. Threat Hunting
  - **ATLAS Tactic & Technique**: Analyze Data/Information
  - **Objective**: Proactively search for hidden threats within the system that may have bypassed initial defenses.

PRE AND POST PRODUCTION DESIGN THREAT ANALYSIS AND MODELING ACTIVITIES

- **Pre-Production Design Threat Analysis**

1. **Threat Identification**
   - **Activity**: Define threat actors, potential targets, and attack vectors.
   - **MITRE ATLAS Tactics**: Reconnaissance.
2. **Risk Analysis**
   - **Activity**: Assess the impact and likelihood of identified threats materializing.
   - **MITRE ATLAS Tactics**: Resource Development.
3. **Security Requirements**
   - **Activity**: Establish security controls based on risk analysis.
   - **MITRE ATLAS Tactics**: Initial Access.
4. **Privacy and Compliance Review**
   - **Activity**: Ensure data protection laws and ethical AI principles are addressed.
   - **MITRE ATLAS Tactics**: N/A (Specific to compliance, not a direct correspondence in MITRE ATLAS).
5. **Secure Development Lifecycle Integration**
   - **Activity**: Integrate security testing and review into the development lifecycle.
   - **MITRE ATLAS Tactics**: Defense Evasion.

PRE AND POST PRODUCTION DESIGN THREAT ANALYSIS AND MODELING ACTIVITIES

Post-Production Threat Modeling

1. **Security Monitoring and Logging**
   - **Activity**: Implement monitoring to detect anomalies and security breaches.
   - **MITRE ATLAS Tactics**: ML Model Access.
2. **Incident Response Planning**
   - **Activity**: Develop plans for responding to security incidents.
   - **MITRE ATLAS Tactics**: Impact.
3. **Regular Security Audits**
   - **Activity**: Conduct periodic security audits to identify new vulnerabilities.
   - **MITRE ATLAS Tactics**: Collection
4. **Continuous Compliance Checks**
   - **Activity**: Continuously monitor for compliance with security requirements.
   - **MITRE ATLAS Tactics**: Discovery.
5. **User and Entity Behavior Analytics (UEBA)**
   - **Activity**: Utilize advanced analytics to detect insider threats or compromised accounts.
   - **MITRE ATLAS Tactics**: Exfiltration.

ENTERPRISE INTERNAL CONTROL FOR LLM/GENAI

- Establishing enterprise internal controls for Large Language Models (LLMs) and General Artificial Intelligence (GenAI) systems is essential for managing risks and ensuring responsible AI usage. Here are examples of some key internal controls:

1. **Access Controls**: Limit who can interact with the LLM/GenAI, including which systems can send inputs and receive outputs.
2. **Data Governance**: Control and monitor the data used for training and operating the AI to ensure its quality, relevance, and security.
3. **Model Validation and Testing**: Regularly test AI models for accuracy, fairness, and reliability across different scenarios and datasets.
4. **Change Management**: Implement strict change management procedures for updates to AI models and their operating environments.
5. **Audit Trails**: Keep detailed logs of AI system activities to support audits and forensics.
6. **Incident Response**: Develop and maintain an incident response plan specifically for AI-related incidents.
7. **Ethical Standards Compliance**: Ensure that AI applications comply with organizational ethical standards and external regulations.
8. **Training and Awareness**: Conduct regular training for staff who develop, manage, or interact with AI systems.
9. **Third-party Risk Management**: Monitor and manage the risks associated with third-party components in AI systems, such as open-source libraries or external data sources.
10. **Continuous Monitoring**: Employ tools and processes to continuously monitor the performance and behavior of AI systems.

QUESTION AND ANSWER

WORKING WITH SECURITY CONTROL FOR THREATS ASSOCIATED WITH LLMS/GENAI

- Key Considerations in Threat Modeling for AI/ML Systems
- Identifying Malicious Behavior in LLMs/GenAI
- Signals of Malicious Behavior: Overview of what signals and patterns to look for that may indicate malicious behavior within AI systems.
- Detecting Data Poisoning: Specific techniques and tools for detecting data poisoning and other forms of adversarial attacks on AI models.
- Implementing Security Controls: Detailed session on implementing the discussed security controls in real-world AI applications.

WORKING WITH SECURITY CONTROL FOR THREATS ASSOCIATED WITH LLMS/GENAI

- Understanding Threat Landscape: Before implementing security controls, it's crucial to comprehend the potential threats associated with LLMs/Generative AI. These threats may include misinformation dissemination, generation of harmful content (such as fake news or malicious code), privacy breaches, manipulation of financial markets, and more.
- Access Control: Limiting access to LLMs and their training data can help mitigate security risks. Implement strict access controls to prevent unauthorized individuals or entities from tampering with the model or its training data.
- Data Sanitization and Validation: Ensure that the training data fed into the LLMs are thoroughly sanitized and validated to remove any malicious or sensitive content. This can help prevent the model from generating harmful or inappropriate outputs.
- Anomaly Detection: Implement anomaly detection mechanisms to identify and flag unusual or potentially malicious activities involving LLMs. This may involve monitoring the model's behavior and output for deviations from expected norms.
- Model Verification and Validation: Regularly verify and validate LLMs to ensure their integrity and reliability. This may include conducting rigorous testing, peer reviews, and audits to identify and address any vulnerabilities or weaknesses in the model.

WORKING WITH SECURITY CONTROL FOR THREATS ASSOCIATED WITH LLMS/GENAI

Column Name	Description
Anomaly Type	Types of anomalies that can occur in LLMs/GenAI systems, such as data anomalies, model anomalies, or behavior anomalies.
Detection Methods	Techniques and algorithms used for detecting anomalies in LLMs/GenAI systems, such as statistical methods, machine learning models, or rule-based systems.
Challenges	Challenges associated with anomaly detection in LLMs/GenAI systems, such as high-dimensional data, concept drift, or interpretability of anomalies.
Applications	Applications and use cases of anomaly detection in LLMs/GenAI systems, such as fraud detection, cybersecurity, or quality assurance.

WORKING WITH SECURITY CONTROL FOR THREATS ASSOCIATED WITH LLMS/GENAI

- Adversarial Testing: Perform adversarial testing to evaluate the robustness of LLMs against various attack vectors, including adversarial inputs designed to exploit vulnerabilities in the model.
- Ethical Guidelines and Governance: Establish clear ethical guidelines and governance frameworks for the development and deployment of LLMs. Ensure that these guidelines address security concerns and promote responsible AI usage.
- Transparency and Explainability: Promote transparency and explainability in LLMs to facilitate understanding of their decision-making processes and outputs. This can help identify and address potential security issues more effectively.
- Continuous Monitoring and Updates: Implement continuous monitoring mechanisms to track the performance and behavior of LLMs in real-time. Regularly update the models with security patches and enhancements to address emerging threats and vulnerabilities.
- Collaboration and Information Sharing: Foster collaboration and information sharing within the AI research community to collectively address security challenges associated with LLMs. This can involve sharing best practices, threat intelligence, and lessons learned from security incidents.

KEY CONSIDERATIONS IN THREAT MODELING FOR AI/ML SYSTEMS

- Threat modeling for AI/ML systems involves identifying potential security threats and vulnerabilities specific to artificial intelligence and machine learning technologies. Here are some key considerations:
- Data Security and Privacy: AI/ML systems heavily rely on data for training and inference. Therefore, ensuring the security and privacy of data is paramount. Threats may include unauthorized access, data breaches, data poisoning, and data manipulation.
- Model Security: The integrity and security of the AI/ML model itself must be ensured. Threats may include model inversion attacks, model stealing, adversarial attacks, and backdoor attacks.
- Adversarial Attacks: Adversarial attacks are a significant concern in AI/ML systems, where malicious actors can manipulate inputs to deceive the system. Threat modeling should consider various types of adversarial attacks such as evasion attacks, poisoning attacks, and model extraction attacks.
- Bias and Fairness: AI/ML models can inherit biases from the data they are trained on, leading to unfair or discriminatory outcomes. Threat modeling should consider the potential implications of biased decision-making and methods to mitigate bias in the system.
- Explainability and Transparency: Understanding how AI/ML models make decisions is crucial for detecting and mitigating potential threats. Threat modeling should incorporate mechanisms for model explainability and transparency to enable stakeholders to understand and validate model behavior.

KEY CONSIDERATIONS IN THREAT MODELING FOR AI/ML SYSTEMS

- Robustness to Distribution Shifts: AI/ML models may encounter data distributions in the real world that differ from their training data, leading to degraded performance or vulnerabilities. Threat modeling should account for robustness to distribution shifts and incorporate strategies such as domain adaptation and continual learning.
- Operational Security: Considerations for operational security include securing access to AI/ML systems, implementing proper authentication and authorization mechanisms, and monitoring for unauthorized activities or anomalies.
- Regulatory Compliance: Compliance with regulations such as GDPR, CCPA, HIPAA, or industry-specific standards is essential for AI/ML systems, particularly concerning data privacy and security. Threat modeling should align with relevant regulatory requirements.
- Supply Chain Risks: AI/ML systems often rely on components and libraries from various sources, introducing supply chain risks. Threat modeling should assess the security of third-party components and consider the potential impact of supply chain attacks.
- Resilience to Attacks: In addition to preventing attacks, AI/ML systems should be resilient to attacks, meaning they can detect, mitigate, and recover from security incidents effectively. Threat modeling should include strategies for intrusion detection, anomaly detection, and incident response.
- Ethical Considerations: Threat modeling for AI/ML systems should also consider ethical implications, ensuring that the system's deployment aligns with ethical principles and societal values.

IDENTIFYING MALICIOUS BEHAVIOR IN LLMS/GENAI

- Here are some key aspects of identifying malicious behavior in LLMs/GenAI:
- Contextual Understanding: To identify malicious behavior, it's essential to have a deep understanding of the context in which the model is operating. This involves analyzing the input prompts given to the model and the generated outputs in various contexts to determine whether they align with ethical and legal standards.
- Pattern Recognition: Researchers and developers work on developing algorithms that can recognize patterns associated with malicious behavior. This might include identifying language patterns commonly found in spam, phishing attempts, hate speech, misinformation, or other forms of harmful content.
- Anomaly Detection: Anomaly detection techniques are employed to identify deviations from expected behavior. These techniques involve monitoring the behavior of the model during inference and flagging any unusual or suspicious activities that might indicate malicious intent.
- Fine-tuning and Control: Model fine-tuning and control mechanisms are implemented to steer the behavior of LLMs away from generating harmful content. This involves training the model on datasets that specifically include examples of malicious behavior and reinforcing ethical guidelines during training.
- Human Oversight: Human oversight plays a crucial role in identifying and mitigating malicious behavior. Humans can review the outputs of LLMs to ensure that they comply with ethical standards and intervene when necessary to prevent the dissemination of harmful content.

IDENTIFYING MALICIOUS BEHAVIOR IN LLMS/GENAI

Indicator Type	Description	Example	Mitigation Strategy
Unusual Input Patterns	Detecting abnormal input patterns that deviate significantly from normal usage or expected data distributions.	Sudden influx of offensive language or hate speech in generated text.	Implement anomaly detection algorithms to flag unusual input patterns for further investigation and filtering.
Output Discrepancies	Identifying inconsistencies or discrepancies between generated outputs and expected results, indicating potential malicious tampering or model manipulation.	AI-generated content deviates significantly from training data or exhibits biased or harmful language.	Conduct regular auditing and validation of AI model outputs using human reviewers or automated verification tools.
Anomalous Resource Usage	Monitoring resource utilization metrics, such as CPU, memory, or network bandwidth, to detect abnormal consumption patterns that may indicate malicious activities or denial-of-service attacks.	Unusually high computational resources consumed during model inference, exceeding typical usage patterns.	Implement resource usage monitoring tools and thresholds to detect and mitigate anomalous resource consumption by LLMs/GenAI systems.
Adversarial Attack Signatures	Identifying known signatures or patterns associated with adversarial attacks, such as gradient-based perturbations or input manipulation techniques, in generated outputs or model behavior.	AI-generated content exhibits subtle perturbations or alterations designed to deceive AI detection systems or mislead human observers.	Employ adversarial robustness testing and detection techniques to identify and mitigate potential adversarial attacks targeting LLMs/GenAI systems.

IDENTIFYING MALICIOUS BEHAVIOR IN LLMS/GENAI

- Collaborative Efforts: Collaboration between researchers, developers, policymakers, and industry stakeholders is essential for effectively identifying and addressing malicious behavior in LLMs. This collaboration can involve sharing best practices, developing standardized evaluation metrics, and establishing guidelines for responsible AI deployment.
- Ethical Guidelines and Regulations: Ethical guidelines and regulations are developed to set standards for the responsible use of AI technology. These guidelines outline principles for developers and users to follow to ensure that AI systems, including LLMs, are deployed in a manner that minimizes harm and maximizes societal benefit.
- Overall, identifying malicious behavior in LLMs/GenAI requires a multidisciplinary approach that combines technical expertise with ethical considerations and regulatory frameworks to mitigate the risks associated with AI misuse.

SIGNALS OF MALICIOUS BEHAVIOR

- Content Analysis:
  - Unintended Bias: Look for language that reflects biased, prejudiced, or harmful viewpoints. This could include hate speech, discrimination, or promoting violence.
  - Abnormal Language Generation: Identify text that appears nonsensical, inconsistent, or highly improbable based on typical language patterns.
  - Out-of-Domain Responses: Flag responses that are unrelated to the context or prompt given to the LLM.
- Contextual Signals:
  - Response Appropriateness: Evaluate whether the generated response is appropriate and relevant to the input prompt.
  - Semantic Coherence: Assess whether the generated text maintains logical coherence and consistency with the input context.
  - Contextual Understanding: Determine if the LLM demonstrates an understanding of the topic or context presented in the prompt.
- Behavioral Signals:
  - Repetitive Output: Look for instances where the LLM generates repetitive or redundant output.
  - Unnatural Human-Like Behavior: Identify behavior that seems too human-like or deviates from typical human communication patterns.
  - Evasive or Malicious Intent: Detect language suggesting evasion of questions, manipulation, or attempts to deceive.

SIGNALS OF MALICIOUS BEHAVIOR

- Adversarial Inputs:
  - Adversarial Attack Detection: Develop techniques to detect adversarial inputs designed to manipulate or exploit vulnerabilities in LLMs.
  - Perturbation Analysis: Analyze how small changes in input affect the output to detect adversarial attempts.
- User Feedback and Monitoring:
  - Human Oversight: Employ human moderators or reviewers to monitor and evaluate LLM outputs for malicious behavior.
  - User Reporting: Establish mechanisms for users to report potentially malicious content generated by LLMs.
- Ethical and Legal Guidelines:
  - Compliance Checks: Ensure that LLM outputs adhere to ethical guidelines and legal regulations governing content generation, such as avoiding defamation or incitement to violence.
  - Responsible AI Practices: Implement principles of responsible AI development and deployment to mitigate potential risks associated with malicious behavior.

DETECTING DATA POISONING

- Here are some methods and techniques used to detect data poisoning in LLMs/GenAI:
- Anomaly Detection: Anomaly detection techniques can be employed to identify unusual patterns or outliers in the training data that may indicate the presence of poisoned samples. These anomalies could include unexpected language patterns, syntactic errors, or semantic inconsistencies.
- Adversarial Testing: Adversarial testing involves subjecting the model to intentionally crafted inputs designed to expose vulnerabilities or biases. By analyzing the model's responses to these inputs, researchers can identify potential areas of weakness or susceptibility to data poisoning.
- Robustness Evaluation: Robustness evaluation techniques assess the model's performance under various stress conditions, including exposure to poisoned data. By systematically testing the model's behavior in adversarial scenarios, researchers can gauge its resilience to data poisoning attacks.
- Statistical Analysis: Statistical methods can be used to analyze the distribution of features within the training data and identify deviations that may indicate the presence of poisoned samples. Techniques such as hypothesis testing, clustering, and outlier detection can help uncover suspicious patterns in the data.

DETECTING DATA POISONING

Methodology/Technique	Description	Advantages/Strengths
Statistical Analysis	Analyzing statistical properties of training data to detect anomalies or inconsistencies that may indicate data poisoning.	- Can detect subtle changes or anomalies in the data. - Doesn't require access to the entire dataset.
Adversarial Training	Training LLMs/GenAI systems with adversarially crafted data samples designed to expose vulnerabilities to data poisoning attacks.	- Improves the model's resilience against data poisoning attacks. - Provides a proactive defense mechanism.
Model Behavior Monitoring	Monitoring the behavior of the trained model during inference to identify unexpected or malicious outputs that may indicate data poisoning.	- Can detect data poisoning in real-time during model deployment. - Helps in identifying subtle deviations from expected behavior.

DETECTING DATA POISONING

- Input Sanitization: Input sanitization involves preprocessing the training data to remove or mitigate potential sources of bias or poisoning. This may include filtering out irrelevant or malicious content, correcting errors, or augmenting the data with counterexamples to balance out biases.
- Model Monitoring: Continuous monitoring of the model's behavior in production environments can help detect anomalies or deviations from expected performance, which may signal the presence of data poisoning. Real-time monitoring allows for prompt intervention and mitigation of potential threats.
- Human Oversight: Human oversight and review play a critical role in detecting data poisoning, particularly in cases where automated techniques may fall short. Domain experts can provide valuable insights into the quality and integrity of the training data and flag suspicious patterns or anomalies for further investigation.
- Collaborative Defense: Collaborative efforts involving researchers, practitioners, and industry stakeholders can facilitate the exchange of knowledge, techniques, and best practices for detecting and mitigating data poisoning in LLMs/GenAI. Sharing insights and experiences across the community can enhance the collective ability to address this evolving threat.

IMPLEMENTING SECURITY CONTROLS

- Implementing security controls in large language models (LLMs) or general artificial intelligence (GenAI) systems is crucial to ensure the safety and integrity of these powerful technologies. Here are some key considerations and details regarding implementing security controls in LLMs/GenAI:
- Threat Landscape Analysis: Before implementing security controls, it's important to understand the potential threats and risks associated with LLMs/GenAI. This analysis should encompass both internal and external threats, including malicious actors, data breaches, model tampering, and unintended consequences of AI-generated content.
- Data Security: Protecting the data used to train and fine-tune LLMs/GenAI is essential. This involves implementing encryption techniques, access controls, and secure storage mechanisms to prevent unauthorized access, data leaks, or theft.
- Model Security: Safeguarding the integrity of the model itself is critical. Techniques such as model watermarking, integrity checks, and version control can help detect and prevent unauthorized modifications or tampering with the AI model.
- Access Control: Limiting access to LLMs/GenAI systems and resources to authorized personnel or applications can help mitigate the risk of unauthorized usage or malicious activities. Role-based access control (RBAC) and multi-factor authentication (MFA) are commonly used access control mechanisms.

IMPLEMENTING SECURITY CONTROLS

- Adversarial Robustness: LLMs/GenAI systems are vulnerable to adversarial attacks, where subtle modifications to input data can lead to incorrect or malicious outputs. Implementing techniques such as adversarial training, input validation, and robustness testing can help improve the resilience of these systems against such attacks.
- Monitoring and Logging: Continuous monitoring and logging of system activities, including model inference requests, data accesses, and user interactions, are essential for detecting and responding to security incidents in a timely manner. Security information and event management (SIEM) systems can aid in centralizing and analyzing log data for potential security threats.
- Compliance and Standards: Adhering to relevant security standards and regulations, such as GDPR, HIPAA, or ISO/IEC 27001, is crucial for ensuring legal and regulatory compliance in the deployment and operation of LLMs/GenAI systems.
- Ethical Considerations: Security controls should also address ethical concerns surrounding LLMs/GenAI, such as fairness, transparency, and accountability. Implementing mechanisms for bias detection and mitigation, explainability of AI decisions, and ethical review processes can help address these concerns.

IMPLEMENTING SECURITY CONTROLS

Security Control	Description	Implementation Approach	Example Techniques
Data Encryption	Encrypting data used for training and fine-tuning LLMs/GenAI to prevent unauthorized access and data breaches.	Utilize strong encryption algorithms (e.g., AES)	AES-256 encryption, Homomorphic encryption
Access Control	Restricting access to LLMs/GenAI systems and resources to authorized users or applications.	Implement role-based access control (RBAC), Multi-factor authentication (MFA)	Role-based access control (RBAC), OAuth, LDAP authentication
Adversarial Robustness	Enhancing LLMs/GenAI resilience against adversarial attacks by validating and securing input data.	Adversarial training, Input validation, Robustness testing	Adversarial training, Adversarial example detection, Input perturbation
Monitoring and Logging	Continuous monitoring and logging of system activities to detect and respond to security incidents in real-time.	Utilize Security Information and Event Management (SIEM) systems	SIEM integration, Log aggregation and analysis, Real-time alerting

IMPLEMENTING SECURITY CONTROLS

- Incident Response and Recovery: Establishing robust incident response plans and procedures, including containment, investigation, and recovery steps, is essential for minimizing the impact of security breaches or incidents involving LLMs/GenAI systems.
- Security Training and Awareness: Educating stakeholders, including developers, users, and administrators, about security best practices and potential risks associated with LLMs/GenAI is critical for fostering a security-conscious culture and ensuring effective implementation of security controls.
- By addressing these aspects and implementing appropriate security controls, organizations can enhance the resilience and trustworthiness of LLMs/GenAI systems, thereby mitigating potential risks and maximizing their benefits.

QUESTION AND ANSWER

QUESTION AND ANSWER