AI security threat model: A comprehensive approach

Dr Anton Tkachenko

November 6, 2025

PROMON

AI Security Threat Model: A Comprehensive Approach

Anton Tkachenko

November 6, 2025

1 Introduction

The increasing use of artificial intelligence (AI) in critical domains necessitates a preventative approach to security. The AI development lifecycle encompasses several stages: data collection and preparation, model development and training, model operation, and application integration.

While our expertise may be limited regarding data preparation and model training outside our professional domain, we specialize in the security aspects of model operation and application integration. These latter stages directly intersect with our core work in application protection.

The initial phases of AI development—data collection, preparation, and training—are typically internal processes where teams must anticipate threats like data poisoning. However, our focus is specifically on the security challenges that emerge during model deployment and integration with mobile and desktop applications, which is why we’ve developed this comprehensive threat model for these environments.

2 Document Structure and Purpose

This document systematizes specific threats related to model operation and application integration. For each threat, we provide:

- Detailed threat description
- Potential consequences
- Compromised information properties:
  - Confidentiality
  - Integrity
  - Availability
  - Accuracy
- Vulnerable attack surfaces

This threat model serves as a critical tool for all stakeholders involved in AI implementation:

- Developers responsible for model development and training
- Cybersecurity specialists protecting data and IT infrastructure
- Solution architects integrating AI into business processes
- Executives evaluating AI implementation risks in mission-critical operations

Our systematic approach covers the entire AI lifecycle—from initial data preparation through model operation.

This methodology helps organizations identify vulnerabilities and implement preventive measures based on Promon’s specialized expertise and leading cybersecurity frameworks including OWASP, MITRE, and NIST. Here you can find: General scheme of the object of protection during operation of the model and integration with applications on scheme i DRAW(Figure 1: General scheme of the object of protection during operation of the model and integration with applications.)

3 Regulatory Compliance Considerations

AI security threats must be viewed through the lens of regulatory requirements:

- **EU AI Act:** Risk-based classification system requiring different levels of controls based on model risk categorization.
- **DataProtectionRegulations:** GDPR and similar frameworks requiring protection of personal data processed by AI systems.
- **Sector-Specific Requirements:** Additional regulations for AI in finance, healthcare, critical infrastructure, and other sensitive domains.
- **TransparencyRequirements:** Obligations to document AI systems, their training data, and decision-making processes.
- **Security Standards:** Emerging frameworks for AI security from standards bodies like ISO, NIST, and industry consortia.

4 Device AI Security Threats

4.1 Dev01 Model Substitution or Modification

Description: Unauthorized modification or substitution of a model due to weak access control

Consequences: Biased model results, reduced accuracy, or exploitation of backdoors in the model

Target Object: Mode˙l inference deployment and hosting infrastructure on devices

Violated Property: Confidentiality, integrity, availability, accuracy

Persons ResponsibleforThreatMitigation: Application security team; device security manager

4.2 Dev02 Model Theft

Description: Model theft due to weak access control or insufficient obfuscation

Consequences: Creation of shadow models or recovery of training data due to white-box access to the stolen model

Target Object: Model inference deployment infrastructure on devices

Violated Property: Confidentiality

Persons ResponsibleforThreatMitigation: Application security team; device security manager

4.3 Dev03 Model Availability Disruption

Description: Attacks targeting API endpoints, resource exhaustion, or exploitation of vulnerabilities in the on-device AI framework

Consequences: Termination or reduction in the quality of AI service provision to end users due to disruption of model availability

Target Object: Model inference deployment and hosting infrastructure on devices

Violated Property: Availability

Persons Responsible for Threat Mitigation: Application security team; device security manager

4.4 Dev04 Leaks of Confidential Information from Logging Systems

Description: Information leaks from logs (journals) of requests containing personal data and other confidential information, including details about the implementation of AI agents and multi-agent systems

Consequences: Theft of confidential information, privacy violations

Target Object: Application infrastructure, logging mechanisms, AI agents

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; privacy officer

4.5 Dev05 Inability to Detect Security Incidents Due to Insufficient Logging

Description: Absence or incompleteness of interaction logging data (including model requests and responses, telemetry data, function calls, etc.) between the model and app components

Consequences: Increased time or inability to detect, respond to, and investigate security events and incidents

Target Object: Device instrumentation, application infrastructure

Violated Property: Integrity, availability

Persons Responsible for Threat Mitigation: Application security team; security operations center

4.6 Dev06 Interception or Substitution of Model Requests/Responses

Description: Interception or substitution of model requests or responses through man-in-the-middle (MiTM) attacks, tampering with communication between app components

Consequences: Interception of sensitive data, modification of requests to obtain incorrect predictions, or substitution of model responses to mislead users

Target Object: Application infrastructure, communication channels, AI agents, local databases

Violated Property: Confidentiality, integrity, availability, accuracy

Persons Responsible for Threat Mitigation: Application security team; network security team

4.7 Dev07 Unauthorized Disabling of Input/Output Filtering Mechanisms

Description: Disabling or changing systems designed to verify and clean data entering the model or returned to the user

Consequences: Processing of malicious or incorrect input data, model malfunctions, returning unwanted, dangerous, or confidential content to the user

Target Object: Input validation components, output filtering mechanisms

Violated Property: Integrity, confidentiality, accuracy

Persons Responsible for Threat Mitigation: Application security team; AI safety team

4.8 Dev08 System Prompt Theft

Description: Gaining access to system prompts that control the behavior of a model, application, or AI agent, with the purpose of stealing them for reverse engineering

Consequences: Disclosure of intellectual property, facilitation of prompt attacks, loss of competitive advantages

Target Object: Application infrastructure, AI agents, prompt storage

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; intellectual property protection team

4.9 Dev09 Unauthorized Modification of System Prompt

Description: Gaining access to system prompts that control the behavior of a model to manipulate the output or functionality

Consequences: Disruption of model or application functionality, including incorrect, malicious, or unwanted generations

Target Object: Application infrastructure, AI agents, prompt storage

Violated Property: Integrity, accuracy

Persons Responsible for Threat Mitigation: Application security team; AI safety team

4.10 Dev10 Unauthorized Modification of Data in Internal Data Sources

Description: Access to internal storage and modification of data used in model operation, to inject malicious information or unethical content

Consequences: Distortion of model output, violation of response integrity, generation of incorrect instructions

Target Object: Application infrastructure, local databases, caches

Violated Property: Confidentiality, integrity, availability, accuracy

Persons Responsible for Threat Mitigation: Application security team; data security team

4.11 Dev11 Information Leaks from Internal Data Sources

Description: Unauthorized copying, transfer, or disclosure of confidential information or personal data stored in local databases or files

Consequences: Leakage of confidential information, privacy violations, regulatory non-compliance

Target Object: Application infrastructure, local databases

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; privacy officer

4.12 Dev12 Unauthorized Connections to the Model

Description: Connection and access to the model using compromised credentials or through intermediary systems **Consequences:** Inability to detect security events, resource exhaustion, unauthorized access **Target Object:** Model inference deployment infrastructure on devices **Violated Property:** Availability, confidentiality **Persons Responsible for Threat Mitigation:** Application security team; authentication system owners

4.13 Dev13 AI Agent Data Leaks or Implementation Details

Description: Unauthorized copying, transfer, or disclosure of information related to the AI agent, including its purpose, function descriptions, or memory contents **Consequences:** Violation of confidentiality of the AI agent's proprietary implementation **Target Object:** Application infrastructure, AI agents **Violated Property:** Confidentiality **Persons Responsible for Threat Mitigation:** Application security team; intellectual property protection team

4.14 Dev14 Unauthorized Modification of AI Agent

Description: Unauthorized modification of an AI agent through adding malicious commands or altering available functions **Consequences:** Disruption of AI agent functionality or the application implementing it **Target Object:** Application infrastructure, AI agents **Violated Property:** Confidentiality, integrity, availability, accuracy

Persons Responsible for Threat Mitigation: Application security team; AI safety team

4.15 Dev15 Information Leakage About Multi-Agent System Architecture

Description: Extraction of information about a multi-agent system (including its architecture, composition, interaction rules) from device interfaces or memory **Consequences:** Violation of confidentiality of the multi- agent system design, facilitation of targeted attacks **Target Object:** Application infrastructure, AI agents
Violated Property: Confidentiality **Persons Responsible for Threat Mitigation:** Application security team; system architecture team

4.16 Dev16 Insider Threats

Description: Misuse of authorized access by insiders (e.g., developers, administrators) to compromise the AI system, such as modifying models, stealing data, or disabling security controls.
Consequences: Unauthorized access, data theft, system compromise.
Target Object: Entire AI system on the device.
Violated Property: Confidentiality, integrity, availability.
Persons Responsible for Threat Mitigation: Application security team, security operations center.

5 Model-Related AI Security Threats

5.1 Mod01 Bypassing Input/Output Processing Mechanisms

Description: Discovering methods to bypass or disrupt the operation of input or output processing mechanisms, including sanitization, validation, and filtering systems in AI models

Consequences: Injection of malicious data, manipulation of results, or unauthorized access to information through compromised security controls
Target Object: Input and output processing components of AI models
Violated Property: Confidentiality, integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; AI safety team

5.2 Mod02 Model Availability Disruption (DoS) Through Request Manipulation

Description: Using specially crafted inputs (e.g., computationally intensive) or sending a large number of requests specifically designed to maximize load on the model inference environment
Consequences: Termination or degradation of AI service performance, device battery drain, or processing capacity exhaustion
Target Object: Input processing mechanisms, model inference endpoints
Violated Property: Availability
Persons Responsible for Threat Mitigation: Application security team; system performance team

5.3 Mod03 Resource Quota Exhaustion (DoW) From Uncontrolled Requests

Description: Sending excessive requests to AI models, resulting in increased token usage and quota exhaustion
Consequences: Service interruption due to quota limits or unexpected financial costs for API usage
Target Object: Input processing mechanisms, API request management
Violated Property: Availability
Persons ResponsibleforThreatMitigation: Application security team; cloud resource management team

5.4 Mod04 Bypassing Built-in AI Safety Mechanisms

Description: Bypassing built-in protective mechanisms of the model including through adversarial attacks and prompt-engineering techniques

Consequences: Incorrect or unauthorized model behavior, including safety violations or content policy bypasses

Target Object: Input processing mechanisms, model security controls

Violated Property: Confidentiality, integrity, availability, accuracy

Persons Responsible for Threat Mitigation: Application security team; AI safety team

5.5 Mod05 Model Information Leakage

Description: Using various methods (e.g., repeated queries or analysis of model behavior) to extract information about the model, its architecture, decision boundaries, and related artifacts

Consequences: Leakage of intellectual property, facilitation of adversarial and prompt attacks

Target Object: Output processing mechanisms, model files

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; intellectual property protection team

5.6 Mod06 Confidential Information Leakage From Fine-tuned Models

Description: Using specially crafted inputs (e.g., with jailbreaking techniques) to extract confidential information from fine-tuned models or LoRA adaptations

Consequences: Leakage of confidential information, including potentially sensitive training data **Target Object:** Output processing mechanisms, fine-tuned model components **Violated Property:** Confidential- ity **Persons Responsible for Threat Mitigation:** Application security team; data protection officer

5.7 Mod07 Model Extraction, Inversion, or Reverse Engineering

Description: Sending multiple queries to the model, analyzing responses to create a functional copy of the model, recreating its behavior without direct access to its architecture **Consequences:** Model theft, creation of unauthorized duplicates, intellectual property loss **Target Object:** Output processing mechanisms, model inference endpoints **Violated Property:** Confidentiality **Persons Responsible for Threat Mitigation:** Application security team; intellectual property protection team

5.8 Mod08 Training Data Exfiltration and Model Inversion

Description: Using specially crafted inputs or analyzing model outputs to extract confidential information, including training data fragments or reconstructed inputs (e.g., through model inversion attacks). This can involve techniques like membership inference or direct inversion of model predictions. **Consequences:** Recovery of sensitive training data or other confidential information, leading to privacy violations and potential regulatory non-compliance. **Target Object:** Output processing mechanisms, model memory.

Violated Property: Confidentiality.

Persons Responsible for Threat Mitigation: Application security team, data protection officer.

5.9 Mod09 Adversarial Attacks

Description: Crafting inputs designed to cause the AI model to make incorrect predictions or classifications, exploiting the model's vulnerabilities.
Consequences: Incorrect model outputs, which can lead to wrong decisions, security bypasses, or degraded performance.
Target Object: Model inference process.
Violated Property: Accuracy, integrity.
Persons Responsible for Threat Mitigation: Application security team, AI safety team.

5.10 Mod10 Backdoor Attacks

Description: Implanting hidden triggers in the model during training or deployment that can be activated to cause malicious behavior.
Consequences: Unauthorized control over model behavior, potential for data exfiltration or sabotage.
Target Object: Model internals.
Violated Property: Integrity, confidentiality, availability.
Persons Responsible for Threat Mitigation: Application security team, AI safety team.

6 Application-Related AI Security Threats

6.1 App01 Insecure Component Integration

Description: Using unsafe integrations of functional components (AI agents, functions, plugins) in applications, including lack of input/output validation, use of insecure APIs, absence of data encryption during transmission, and incorrect access rights configuration

Consequences: Confidential information leaks, availability disruption, response integrity violations, incorrect or unauthorized model behavior
Target Object: Mobile application with AI capabilities
Violated Property: Confidentiality, integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.2 App02 Bypassing Application-Level Input/Output Controls

Description: Discovering methods to bypass or disrupt application-level input/output processing mechanisms, including sanitization, validation, and filtering systems in mobile applications with AI features
Consequences: Injection of malicious data, manipulation of results, unauthorized access to information
Target Object: Input/output processing mechanisms within mobile applications
Violated Property: Confidentiality, integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.3 App03 Malicious Code Loading from External Sources

Description: Using specially crafted data sources (including databases, code repositories, or websites) hosting malicious code that can be executed by mobile applications with AI capabilities that have code execution features

Consequences: Execution of malicious payloads leading to further security breaches, including data theft, credential harvesting, or device compromise
Target Object: External data sources, AI agents, functions within mobile applications
Violated Property: Confidentiality, integrity, availability

Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.4 App04 Loading Poisoned Data from External Sources

Description: Using specially crafted data sources containing files, text, or multimedia with indirect prompt injections that manipulate AI components in mobile applications
Consequences: Confidential information leaks, availability disruption, integrity violations, incorrect model behavior
Target Object: External data sources, AI agents, functions within mobile applications
Violated Property: Confidentiality, integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.5 App05 Injecting Indirect Prompt Attacks into Internal Data Sources

Description: Using specially crafted inputs to inject indirect prompt attacks into internal data sources within mobile applications, particularly when model processing results are stored locally
Consequences: Persistent manipulation of AI behavior through poisoned local data
TargetObject: Internal data sources within mobile applications
ViolatedProperty: Confidentiality, integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; data security team

6.6 App06 Information Leaks from Internal Data Sources

Description: Using specially crafted inputs (e.g., with jailbreaking techniques) to extract confidential information from internal databases and storage within mobile applications
Consequences: Theft of confidential information stored locally on devices
Target Object: Internal data sources within mobile applications
Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; data protection officer

6.7 App07 Execution of Malicious Instructions Generated by AI Models

Description: Using specially crafted inputs to trigger generation of harmful instructions that are then executed within the mobile application environment
Consequences: Unauthorized modification or destruction of information in internal sources, confidential information leaks
Target Object: Internal data sources within mobile applications
Violated Property: Confidentiality, integrity

Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.8 App08 Direct Prompt Injection Due to Inadequate Input Validation

Description: Using specially crafted inputs to conduct prompt attacks against AI models integrated into mobile applications
Consequences: Incorrect or unauthorized model behavior, confidential information leaks
Target Object: Input processing mechanisms within mobile applications
Violated Property: Confidentiality, integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.9 App09 Integration Availability Disruption (DoS/DoW)

Description: Using specially crafted inputs or excessive request volume to cause DoS of the integration or excessive token usage and quota exhaustion (DoW)

Consequences: Service interruption or increased operational costs for AI-enabled mobile applications
Target Object: Input processing mechanisms within mobile applications
Violated Property: Availability
Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.10 App10 Task Execution Logic Disruption

Description: Using specially crafted inputs (adversarial attacks or prompt-attacks) to circumvent instructions defined in the system prompt, limitations, or other system settings

Consequences: Altered operational logic, performance of undeclared actions that were explicitly prohibited or not intended during development

Target Object: Input processing mechanisms within mobile applications
Violated Property: Integrity, availability, accuracy
Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.11 App11 System Prompt Information Leakage

Description: Using specially crafted inputs to obtain information about the system prompt used in mobile applications with AI capabilities

Consequences: Confidential information leaks, facilitation of prompt attacks against the application
Target Object: Output processing mechanisms within mobile applications
Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.12 App12 Toxic or Malicious Content Generation

Description: Using specially crafted inputs to obtain toxic content (violating legal or ethical norms) or malicious content (dangerous instructions, cybercrime guidance, malicious code, vulnerable code)
Consequences: Incorrect or unauthorized application behavior due to model-generated content
Target Object: Output processing mechanisms within mobile applications
Violated Property: Integrity, accuracy
Persons Responsible for Threat Mitigation: Application security team; content safety team

6.13 App13 Environment Information Leakage

Description: Using specially crafted inputs to obtain information about the system configuration, internal processes, API keys, software versions or other details that could be used for further attacks

Consequences: Confidential information leaks, facilitation of cyberattacks against the mobile device or application

Target Object: AI agents, functions within mobile applications
Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; mobile development team

6.14 App14 Automated Propagation of Malicious Instructions

Description: Using specially crafted inputs to implement self-replicating prompt attacks that spread to other applications on the device or through shared data stores

Consequences: Cyberattacks against other applications, execution of malicious payloads that spread beyond the original target

Target Object: AI agents, functions within mobile applications

Violated Property: Integrity, availability

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7 AI Agent Security Threats

7.1 Agt01 Data Exfiltration from AI Agent Runtime Environment

Description: Using specially crafted inputs to trigger functions that send confidential information (including environment variables) from the mobile application to external resources

Consequences: Leakage of sensitive data contained in the runtime environment, facilitation of further attacks

Target Object: AI agents within mobile applications

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7.2 Agt02 File Deletion or Modification in AI Agent Runtime

Description: Using specially crafted inputs to trigger functions that delete or modify files accessible in the mobile application's execution environment

Consequences: Data integrity violations within the mobile execution environment

Target Object: AI agents within mobile applications

Violated Property: Integrity

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7.3 Agt03 Malware Deployment in AI Agent Runtime

Description: Using specially crafted inputs to trigger functions that download malicious files from external sources and execute them within the mobile application environment

Consequences: Execution of malicious payloads leading to further security breaches on the mobile device

Target Object: AI agents within mobile applications

Violated Property: Confidentiality, integrity, availability

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7.4 Agt04 AI Agent Runtime Availability Disruption

Description: Using specially crafted inputs to cause resource exhaustion in the mobile AI agent runtime environment through computational complexity attacks or infinite loops

Consequences: Application unresponsiveness, battery drain, device overheating, or termination of services

Target Object: AI agents within mobile applications

Violated Property: Availability

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7.5 Agt05 Multi-Agent System Architecture Information Leakage

Description: Using specially crafted inputs to extract information about a multi-agent system (including its architecture, composition, interaction rules) from AI agent responses in mobile applications

Consequences: Violation of confidentiality regarding the multi-agent system design, facilitation of targeted attacks

Target Object: AI agents within mobile applications

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; system architecture team

7.6 Agt06 False Information Transmission Between AI Agents

Description: Using specially crafted inputs to modify and distort semantic information transmitted between AI agents in a multi-agent mobile application

Consequences: Disruption of agent interaction and cooperation, compromised application workflow

Target Object: AI agents within mobile applications

Violated Property: Integrity, accuracy

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7.7 Agt07 Goal Manipulation of Cooperative AI Agents

Description: Using specially crafted inputs to modify the goal of another AI agent in a multi-agent mobile system by transmitting a request with a prompt attack

Consequences: Disruption of agent interaction and cooperation, compromised application workflow

Target Object: AI agents within mobile applications

Violated Property: Integrity, accuracy

Persons Responsible for Threat Mitigation: Application security team; mobile development team

7.8 Agt08 Application Workflow Disruption via AI Agent

Description: Using specially crafted inputs to modify and distort the semantic content, structure, or format of information presented by an AI agent in a mobile application

Consequences: Corrupted output results, application availability disruption, functional compromise **Target Object:** AI agents within mobile applications **Violated Property:** Integrity, availability, accuracy **Persons Responsible for Threat Mitigation:** Application security team; mobile development team

7.9 Agt09 AI Agent Configuration Information Leakage

Description: Using specially crafted inputs to extract information related to the AI agent, including its goals, function descriptions, planning mechanism instructions, or memory contents from mobile applications

Consequences: Violation of confidentiality regarding the AI agent's configuration, intellectual property theft

Target Object: AI agents within mobile applications

Violated Property: Confidentiality

Persons Responsible for Threat Mitigation: Application security team; intellectual property protection team

8 Threat Analysis and Promon Protection Solutions

8.1 Critical AI Security Threats for Mobile and Desktop Applications

After systematic analysis of industry frameworks including MITRE ATLAS, OWASP Top-10 LLM 2025, and the OWASP Agentic Threats Taxonomy, we have identified several threat categories of particular significance for organizations deploying AI systems in mobile and desktop environments:

- **Runtime Model Tampering** (Dev01, Dev02, Dev09) - Unauthorized modification, substitution, or theft of deployed models represents a critical risk to system reliability and intellectual property protection, affecting all model types from predictive to generative AI.
- **Prompt Injection Attacks** (App08, Mod04) - Direct and indirect input manipulation designed to bypass AI safety mechanisms and alter model behavior represents one of the most prevalent attack vectors, particularly for generative AI systems.
- **Local Data Store Compromise** (Dev10, Dev11, App05, App06) - Attacks targeting the integrity and confidentiality of local data sources used by AI systems can create persistent vulnerabilities with regulatory implications, especially for systems processing personal data.
- **AI Agent Runtime Exploitation** (Agt01, Agt03, Agt07) - As AI agents gain prominence in application architectures, their runtime environments present elevated risks for unauthorized code execution, data exfiltration, and manipulation of agent behavior.

As summarized in Table 1, we've categorized these threats based on both their potential impact and relevance to Promon's security capabilities.
This analysis highlights the critical threat areas where Promon's solutions provide immediate impact (high relevance), areas for strategic development (medium relevance), and specialized threats that may require complementary technologies (low relevance).

8.2 Promon Protection Matrix

Promon's application protection platform addresses these critical threat categories through a comprehensive security approach as detailed in Table 2. This matrix maps specific Promon protection capabilities to the identified high-relevance threats.

8.3 Implementation Strategy

Effective protection against AI security threats in mobile and desktop applications requires a layered approach that addresses vulnerabilities across the entire application stack, as outlined in our protection matrix (Table 2):

1. **Model-Level Protection**: Securing the AI model itself against extraction, manipulation, and unauthorized access through runtime protection mechanisms that monitor for suspicious interactions with model files.
2. **Communication-Level Protection**: Ensuring the integrity and confidentiality of data flowing between application components and AI systems through strong encryption, certificate validation, and traffic analysis.
3. **Input/Output Security**: Implementing robust validation mechanisms that detect and block malicious inputs before they reach AI components while also filtering potentially harmful outputs generated by the model.
4. **Data Store Security**: Protecting local databases and files that store AI-related data, including model parameters, cached results, and system prompts through encryption and access controls.

5. **Runtime Environment Hardening**: Securing the execution environment for AI agents through memory protection, anti-debugging features, and integrity checks.

These layers combine to create a defense-in-depth strategy that addresses the unique challenges of securing AI components within mobile and desktop applications. By implementing Promon's protection capabilities, organizations can significantly reduce the attack surface available to adversaries targeting their AI-enabled applications.

8.4 Regulatory Alignment

The protection mechanisms outlined above directly support compliance with emerging AI regulations:

- The input/output validation controls align with EU AI Act requirements for high-risk AI systems regarding technical robustness and safety.
- The local data protection capabilities support GDPR compliance for AI systems processing personal data on mobile devices.
- The comprehensive logging and monitoring features enable the transparency and accountability required by various regulatory frameworks.

By implementing these protections, organizations can not only secure their AI deployments but also demonstrate due diligence in addressing the regulatory requirements increasingly applied to AI systems.

9 Promon Solution Relevance Analysis

9.1 Threat Relevance Classification

We have categorized all identified threats based on their current addressability with Promon's application protection technology:

- **High Relevance** - Threats that can be effectively mitigated with Promon's current solution portfolio
- **MediumRelevance** - Threats that can be partially addressed but require additional capability development
- **Low Relevance** - Threats that would require significant extension of Promon's current capabilities

This classification helps organizations prioritize their security investments and understand where Promon's solutions provide immediate value versus areas where complementary technologies may be needed.

As detailed in Table 3, our analysis shows that Promon's current solutions are particularly effective against device-level threats and application security concerns.

Figure 2 provides a visual representation of this threat distribution across different categories, highlighting Promon's current strengths in device-level security.

9.2 Strategic Implications

The relevance analysis presented in Table 3 and Figure 2 reveals important insights for Promon's product strategy and customer guidance:

1. **Current Strength Areas:** Promon's solutions excel at addressing device-level threats to AI systems, particularly those involving runtime integrity, code protection, and secure communications. These capabilities provide immediate value for organizations deploying AI models on mobile and desktop applications.

2. **Near-Term Development Priorities:** Medium-relevance threats represent strategic opportunities for capability extension, particularly in the areas of AI agent protection and enhanced input/output validation for GenAI systems.

3. **Partnership Opportunities:** Low-relevance threats, while important for comprehensive security, may be better addressed through partnerships with specialized AI security vendors rather than direct capability development.

As shown in Figure 2, Promon currently offers strong protection against 14 high-relevance threats, with particular strength in device-level protections. This aligns with Promon's core expertise in application security while highlighting strategic areas for focused innovation.

10 Comprehensive Threat Mitigation Strategies for Promon

Building upon the threat prioritization in Table 1 and the solution mapping in Table 2, this section out- lines concrete mitigation strategies for the 14 high-relevance threats that Promon's solutions are specifically designed to address. These practical implementations enable organizations to effectively protect their AI deployments on mobile and desktop environments.

10.1 Dev01: Model Substitution or Modification

Description: This threat involves an attacker replacing or altering a machine learning model to manipulate its behavior or outputs.

Mitigation Strategies:

- Implement cryptographic integrity checks (e.g., SHA-256 hashes) to verify model authenticity at runtime.
- Deploy Promon SHIELD's runtime integrity verification to detect unauthorized model changes during execution.
- Use anti-tampering controls including repackaging detection and hooking framework detection.
- Implement continuous monitoring of model files and related assets.

10.2 Dev02: Model Theft

Description: This threat involves unauthorized access to and exfiltration of AI model inference code, potentially leading to intellectual property theft or reverse engineering attempts.

Mitigation Strategies:

- Deploy Promon IP Protection Pro to obfuscate model inference code.
- Use advanced code obfuscation techniques to make reverse engineering significantly more difficult.
- Implement anti-reverse engineering protections that detect and respond to analysis attempts.
- Note: While obfuscation cannot prevent file copying, it dramatically reduces the value of stolen models by making them difficult to understand or replicate.

10.3 Dev06: Interception or Substitution of Model Requests/Responses

Description: This threat involves man-in-the-middle attacks attempting to intercept or modify communications between AI components.

Mitigation Strategies:

- Protect existing certificate pinning implementations using Promon SHIELD.
- Deploy anti-tampering controls to prevent bypass of certificate validation.
- Implement runtime integrity checks to detect manipulation of network communication code.
- Monitor for hooking frameworks attempting to intercept SSL/TLS traffic.

10.4 Dev07: Unauthorized Disabling of Input/Output Filtering

Description: This threat involves tampering with systems designed to validate and sanitize data entering or leaving the AI model.

Mitigation Strategies:

- Deploy Promon SHIELD's anti-tampering mechanisms to protect filter infrastructure.
- Implement repackaging detection to identify modified applications with disabled filters.
- Use hooking framework detection to prevent runtime manipulation of filtering logic.
- Apply runtime integrity verification to ensure filtering mechanisms remain operational.

10.5 Dev09: Unauthorized Modification of System Prompt

Description: This threat involves altering the system prompt that controls AI model behavior, which can lead to incorrect or malicious outputs.

Mitigation Strategies:

- Store system prompts in protected memory regions monitored by Promon SHIELD.
- Implement runtime integrity checks specifically for prompt storage and loading mechanisms.
- Use anti-tampering controls to detect modifications to prompt files or in-memory representations.
- Deploy code obfuscation to obscure prompt handling logic.

10.6 Dev10: Unauthorized Modification of Data in Internal Sources

Description: This threat involves tampering with internal data sources used by the AI model, leading to data poisoning or incorrect model behavior.

Mitigation Strategies:

- Deploy Promon Asset Protection to encrypt all sensitive data stored locally.
- Implement secure key management practices to protect encryption keys.
- Use integrity verification mechanisms to detect unauthorized data modifications.
- Apply access controls enforced by SHIELD to restrict data access to authorized components only.

10.7 Dev11: Information Leaks from Internal Data Sources

Description: This threat involves unauthorized copying, transfer, or disclosure of confidential information or personal data stored in local databases or files.

Mitigation Strategies:

- Deploy Promon Asset Protection to encrypt sensitive data at rest.
- Implement secure storage mechanisms for API keys, credentials, and other secrets.
- Use SHIELD's runtime protection to prevent unauthorized access to protected data stores.
- Apply secure key management with hardware-backed keystores where available.

10.8 Dev14: Unauthorized Modification of AI Agent

Description: This threat involves unauthorized modification of an AI agent through adding malicious commands or altering available functions.

Mitigation Strategies:

- Deploy Promon SHIELD's anti-tampering controls to protect agent code and configuration.
- Implement runtime integrity verification for agent components and function definitions.
- Use repackaging detection to identify modified applications with altered agent behavior.
- Apply code obfuscation to agent implementation to hinder reverse engineering.

10.9 App01: Insecure Component Integration

Description: This threat involves unsafe integrations of functional components including lack of input/output validation, use of insecure APIs, absence of data encryption during transmission, and incorrect access rights configuration.

Mitigation Strategies:

- Deploy Promon SHIELD to protect certificate pinning implementations from bypass.
- Implement runtime protection for API integration code.
- Use anti-tampering mechanisms to prevent modification of security controls.
- Apply integrity verification to component integration points.

10.10 App02: Bypassing Application-Level Input/Output Controls

Description: This threat involves discovering methods to bypass or disrupt application-level input/output processing mechanisms, including sanitization, validation, and filtering systems.

Mitigation Strategies:

- Deploy Promon SHIELD's anti-tampering controls to protect input/output validation infrastructure.
- Implement runtime integrity checks for sanitization and filtering code.
- Use repackaging detection to identify applications with modified validation logic.
- Apply hooking framework detection to prevent runtime bypass of control mechanisms.

10.11 App03: Malicious Code Loading from External Sources

Description: This threat involves using specially crafted data sources hosting malicious code that can be executed by applications with AI capabilities that have code execution features.

Mitigation Strategies:

- Deploy Promon SHIELD's application hardening to prevent unauthorized code execution.
- Implement control flow integrity mechanisms to detect and block unexpected code paths.
- Use runtime environment isolation to contain AI components.
- Apply strict validation of any externally loaded code or data.

10.12 App05: Injecting Indirect Prompt Attacks into Internal Data Sources

Description: This threat involves using specially crafted inputs to inject indirect prompt attacks into internal data sources, particularly when model processing results are stored locally.

Mitigation Strategies:

- Deploy Promon Asset Protection to encrypt and protect internal data stores.
- Implement integrity verification for data before it is used by AI models.
- Use secure storage mechanisms that prevent unauthorized data modification.
- Apply access controls to restrict which components can write to AI-consumed data stores.

10.13 App06: Information Leaks from Internal Data Sources

Description: This threat involves unauthorized copying, transfer, or disclosure of confidential information stored in internal databases and storage within applications.

Mitigation Strategies:

- Deploy Promon Asset Protection to encrypt all sensitive data stored by the application.
- Implement secure key management practices.
- Use SHIELD's runtime protection to prevent unauthorized data access.
- Apply data loss prevention controls at the application level.

10.14 Agt03: Malware Deployment in AI Agent Runtime

Description: This threat involves using specially crafted inputs to trigger functions that download malicious files from external sources and execute them within the AI agent environment.

Mitigation Strategies:

- Deploy Promon SHIELD's application hardening to prevent unauthorized code execution.
- Implement runtime environment isolation for AI agents.
- Use control flow integrity to detect and block malicious execution attempts.
- Apply strict validation and sandboxing of agent function execution.

10.15 Implementation Roadmap

Organizations looking to secure their AI deployments with Promon's solutions should follow this phased implementation approach:

1. **Assessment Phase**: Evaluate the AI application architecture and identify which threat categories are most relevant to your specific implementation. Focus on understanding where models, agents, and sensitive data reside.

2. **Deploy Promon SHIELD (Runtime Protection)**: Implement SHIELD as the foundation layer to protect against tampering, code manipulation, and unauthorized execution. This addresses the majority of HIGH threats:

    - Model integrity protection (Dev01, Dev09)
    - Input/output filter protection (Dev07, App02)
    - Certificate pinning protection (Dev06, App01)
    - Agent infrastructure protection (Dev14)

    - Malicious code prevention (App03, Agt03)

3. **Deploy Promon Asset Protection (Data Security)**: Secure local data stores, secrets, and sensitive information to prevent both unauthorized access and modification (Dev10, Dev11, App05, App06).

4. **Deploy Promon IP Protection Pro (Code Obfuscation)**: Protect model inference code and application logic from reverse engineering (Dev02). Critical for proprietary AI implementations.

5. **Validation and Testing**: Verify that all protections are properly configured and not impacting application functionality or AI model performance.

6. **Continuous Monitoring**: Establish ongoing security monitoring specifically focused on AI-related components. Consider Promon Insight for enhanced threat detection as capabilities mature (currently provides basic security event detection).

By following this product-based roadmap, organizations can systematically deploy Promon's security capabilities in a logical order that maximizes protection while minimizing deployment complexity. SHIELD serves as the foundation, with Asset Protection and IP Protection Pro providing specialized protections for data and code respectively.

11 Materials used

1. OWASP Top-10 LLM 2025
2. OWASP Top-10 Machine Learning Security
3. OWASP AI Security Solutions Landscape
4. OWASP Agentic Threats Taxonomy (draft)
5. OWASP AI Exchange 4.5
6. MITRE ATT&CK
7. MITRE ATLAS
8. Google SAIF (Secure AI Framework)
9. NIST Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations
10. AWS Generative AI Security Scoping Matrix

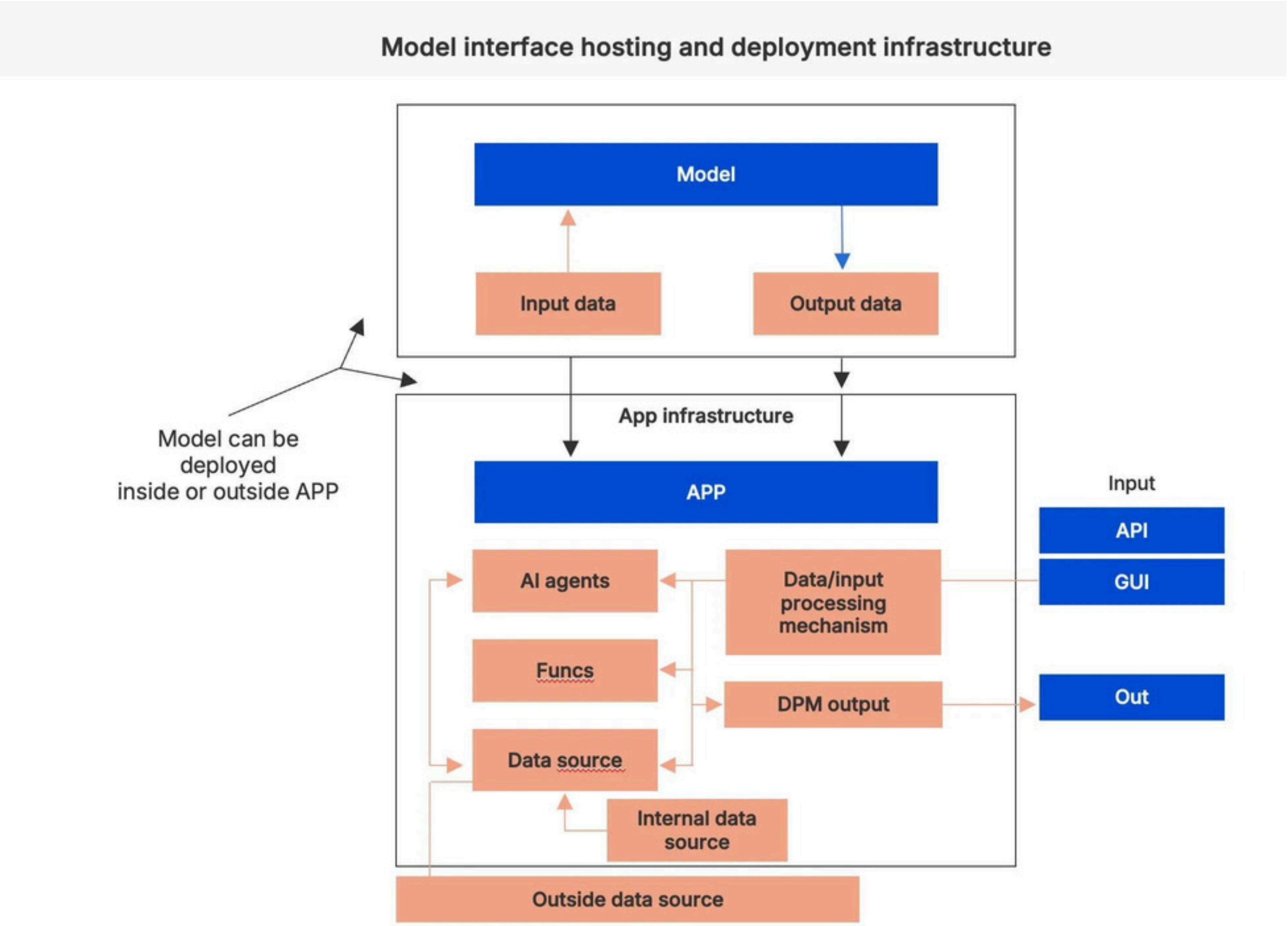

Figure 1: General scheme of the object of protection during operation of the model and integration with applications

PROMON