ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

Emerging LLM Threats: A Comprehensive Analysis of Attacks and Mitigation

Sreenivasa Rao Basavala[1]; Prudhvi Raju Mudunuri[2]

[1]Marymount University/Department of BuILT, Arlington, USA
[2]Independent Researcher, Montgomery Village, USA

Publication Date: 2026/05/29

Abstract: Artificial Intelligence (AI) is changing the way organizations work with new technologies that help to enhance the security of their information assets, while also creating new attack vectors. While AI has the potential to dramatically improve an organization's ability to detect threats, automate repetitive administrative tasks and provide more real-time responsive systems, there are associated risks of exposure, including vulnerabilities in the new systems and software, as well as new types of attack vectors. Examples of new types of AI-based attacks that have been recently discovered and are reported in the research include but are not limited to: - Adversarial attacks - Data poisoning - Prompt injections - Model evasion - Model theft - AI-driven social engineering attacks such as deepfakes and other automated phishing campaigns. These attacks can lead to many types of incidents, including data theft of sensitive data, denial of service, reputational damage due to loss of customer trust and more. At the same time, new attack surfaces have been created, for example, in the form of training data for the new systems, the structure and design of the systems, and dependencies in third party applications and services. This paper aims to provide a deep dive into the many types of cyberattacks that exist in the realm of AI and to provide an in-depth analysis of their methods, techniques and the overall impact of these new types of attacks on the wider Cybersecurity landscape. This paper also aims to give an in-depth look at countermeasures and defenses that can be put in place to help combat these threats, including but not limited to secure coding practices for the development of new systems and AI models, the use of adversarial testing, access controls and real-time threat detection and alerting. Organizations need to be aware of the potential threats of these new technologies and the need to secure their systems using several security controls to mitigate the threats and to be prepared.

Keywords: *AI Security, OWASP LLM Top 10, AI Supply Chain Attacks, Prompt Injection Attacks, Model Theft, Poisoning Model, AI Cyberattacks, Mitigation Techniques.*

How to Cite: Sreenivasa Rao Basavala; Prudhvi Raju Mudunuri (2026) Emerging LLM Threats: A Comprehensive Analysis of Attacks and Mitigation. *International Journal of Innovative Science and Research Technology*, 11(5), 2136-2145. https://doi.org/10.38124/ijisrt/26may925

I. INTRODUCTION

AI cybersecurity is revolutionizing the field of cybersecurity in numerous ways, bringing both benefits and numerous threats. On one hand, the use of artificial intelligence makes it possible to more quickly and efficiently respond to attacks and detect threats, thereby preventing serious incidents. On the other hand, attackers are also using technology to launch more sophisticated attacks that are increasingly difficult to identify. Attackers use AI when they attack an AI system or when they use an AI system to develop a new type of attack that can be more automated, scalable and social engineering [1]. Using AI allows an attacker to better understand the victim system and to make its attacks more effective. According to OWASP (Open Web Application Security Project) and MITRE (MIT Research Establishment), the use of AI in cybersecurity has created new vulnerabilities in the field of machine learning, training data and automated decision-making. The threats are more numerous and diverse, which makes it even more difficult to protect an organization.

There are some big threats to AI systems that we need to think about. One of these is called adversarial attacks, where bad actors (attackers) try to trick AI by changing the information it uses. For example, they might make small changes to malware so that the AI-powered antivirus tools can't catch it - this is called model evasion. They can also use special examples to fool facial recognition systems or tools that detect fraud and another major problem is something called data poisoning [8]. In this attack, the attacker inserts a large amount of false or misleading information into the training data of an AI system. The AI system can then incorrectly learn from the false information, and it may produce some biases in the decisions it makes or even open a "backdoor" that can be triggered remotely later. Think of it as teaching a child a bunch of false information, and then having the child make very important life changing decisions based on that false information. Attacks of this type are particularly problematic. These attacks need to be addressed to prevent potential adverse consequences.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

II. WHY AI SECURITY

This paper discusses the development of the field of AI security, focusing on the security challenges in the era of large-scale AI. Nowadays, many AI applications such as financial, healthcare, government and cyber security systems are being more deeply employed in our daily lives, thus making the AI security an increasingly important issue. In contrast with general software, an AI system contains much more data, learning algorithms and autonomous decision-making behaviors [4]. The AI system can be attacked in several different ways, such as changing, modifying or hiding the training data or other information in the system in order to affect the behaviors of the system, adding backdoors into learning algorithms in order to cause an unexpected or even harmful behavior in the system when some specific inputs appear, modifying the input data of large-scale language models in order to cause the output discriminative or harmful, etc. As AI security is a new risk, there are also many corresponding security measures which still need to be explored.

One more reason is that modern AI systems have become very complex, and therefore more prone to attacks. As mentioned before, AI systems generally depend on a lot of third-party data, and sometimes even rely on pre-trained models, APIs, and cloud services. This makes it highly possible that any AI system could have at least one vulnerable dependency in it. Therefore, it is recommended that organizations put in place more rules and safety measures to secure the AI systems. The companies will be able to deploy the systems safely, reliably, and ethically to avoid malicious use of AI, data breaches, or large-scale DDoS attacks.

III. AI SECURITY VS TRADITIONAL SECURITY

This paper discusses the development of the field of AI security, focusing on the security challenges in the era of large-scale Artificial Intelligence security is the security of general applications and systems which use artificial intelligence. This term is often misleading and confusing. Although an application or system which uses AI has the same attack surfaces as other general applications and systems, it introduces a new set of elements, surfaces and risks. While traditional computer security deals with the same threats that we are familiar with like viruses, unauthorized access, data leakage, etc. for an AI system, we should also consider the security of the machine learning models, the training data and the behavior of the system [11]. Since an application or system which uses AI for training its models or for decision making, its vulnerability to attacks can arise from the training data it uses, the structure of the model or the behavior of the system and the decisions it makes which are often not the threats that the traditional security controls are designed.

- *Data Dependency:*

Most applications currently work with knowledge coded into an application by a human programmer. An AI system works on patterns it is trained to find within the data. And an attacker might be able to modify the training data in a way that the AI identifies a false pattern, thereby allowing a malicious action to occur. So, data integrity needs to be ensured within the system across all layers of the application.

- *New Types of Attacks:*

AI introduces unique attacks such as: Prompt injection in generative AI systems, model poisoning during training, model extraction where attackers steal the AI model, Adversarial attacks that trick AI models with carefully crafted inputs [5]. OWASP has released the OWASP Top 10 for Large Language Model (LLM) Applications. The list of vulnerabilities that have been found in the and listed as "OWASP Top 10 for LLM. Applications".

- *Model and Algorithm Security:*

Most of the time spent on cybersecurity is spent protecting applications and systems. When we look at protecting our systems from the standpoint of "Artificial Intelligence Security", things flip around and we now have to not only secure our Machine Learning (ML) models, but also think about the data and the exposed functionality that these models are presenting, therefore requiring a new breed of cybersecurity controls including but not limited to access control, model exfiltration and tampering. Models can be reverse engineered and the functionality and vulnerabilities in the ML algorithms can be exploited.

- *AI Supply Chain Risks:*

Systems have Non-Functional System Vulnerabilities (NFSVs) because they have many external dependencies. These include pre-trained models and states, training datasets, third-party libraries and APIs, and other software frameworks. Changes to one part of the system could have an impact on many other parts [2]. Managing these types of risk is like managing Non-Functional Risk (NFR) and aligns with recommendations from NIST (National Institute of Standards and Technology) regarding the management of NFR for AI Systems, as well as managing AI software supply chains [7] [12].

- *Autonomous Decision-Making:*

Every normal application follows a set of rules created by a human programmer. An AI application on the other hand follows a set of patterns it has learned from historical data and uses Bayesian probability to infer the likelihood of something happening. The fact that any AI application can be made to produce false, or even worse, biased, unsafe or harmful results when scaled up makes the monitoring, explainability and governance of AI security a matter of high importance.

IV. OWASP TOP 10

The OWASP is an Open Web Application Security Project. The Top 10 represents the most popular and critical application vulnerabilities identified by OWASP [6]. This list is a living summary of the vulnerabilities attackers are targeting in real-world applications, and, as such, allows us to identify the various types of vulnerabilities that could cause an application to exhibit unintended functionality. OWASP developed various Top 10 lists of vulnerabilities across web,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

mobile, IoT, network, API, and LLMs. Table 1 listed the LLM Top10 vulnerabilities as follows.

These AI attacks can cause the model to accept a certain input to cause the generated model to fail to detect threats, steal sensitive information, or produce malicious output. If the generative model is embedded in other systems or applications within the enterprise, then they are potentially at risk. If the model is integrated with other systems, applications or databases as well as autonomous agents, then the impact of a model being compromised can be dramatic. Today, attacks are increasingly AI-based, and examples of this include: Using AI to craft realistic content for phishing emails, Business Email Compromise (BEC), using AI to generate deepfake audio and video for social engineering attacks, Using AI to craft polymorphic malware that changes its code signature each time it is run, and using AI to do the vulnerability discovery of the systems they will later attack.

Table 1 OWASP LLM Top10

Sl. No	Description
LLM01	Prompt Injection
LLM02	Insecure Output Handling
LLM03	Training Data Poisoning
LLM04	Model Denial of Service
LLM05	Supply Chain Vulnerabilities
LLM06	Sensitive Information Disclosure
LLM07	Insecure Plugin Design
LLM08	Excessive Agency
LLM09	Overreliance
LLM10	Model Theft

There's a new worry that's getting attackers stealing models and making inference attacks. This is when attackers look at what the model says and find out secret things about it or the data it was trained on. This can be bad because it can show off important business secrets or sensitive and confidential information that was used to train the model. Also, AI systems can be hurt by something called supply chain attacks. This happens when the data used to train the model, or the model itself, or even parts of the AI system that were made by someone else, have hidden problems that can be used to attack the system.

All the interesting hacks in the world of cybersecurity are now being carried out with the aid of AI. Whether you are talking about a data breach, a case of fraud, identity theft, a misinformation campaign or an infrastructure failure, the common denominator is almost always AI. Therefore, it is now more important than ever to make sure that the systems that we depend on are protected against the new and emerging threats carried out by AI. This includes data validation of any training data, implementing adversarial testing and access controls to name a few. The more obvious controls that have been around for years such as continuous monitoring and threat modeling for AI-related vulnerabilities should also be in place. To aid in the discovery and mitigation of such vulnerabilities, tools like OWASP Top 10 for Large Language Models and the MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems) have been developed [19].

➢ *Direct Prompt Injection*:

Direct prompt injection is the practice of crafting input to dictate how a model will behave. Instead of posing a legitimate question, an attacker can inject or conceal actual commands that the model should ignore. Because large language models such as GPT-4 are designed to act upon natural language input, they will tend to follow the injected input in the absence of proper protection mechanisms as shown in Figure 1. Some examples of injected prompts that an attacker may include in their input include: “Ignore the following” Other injected prompts that an attacker may include in their input to elicit additional information such as: - The internal prompts or expected input that the model is using - restricted or sensitive information that should not be returned [14]. This is especially concerning whether an AI is integrated with an Enterprise Data Source (EDS), APIs or Automation Tools and unintended or manipulated outputs result in sensitive data being exposed or leaked. This has been covered in the OWASP Top 10 for LLM Applications. It is recommended to implement controls such as input validation, prompt isolation, and output monitoring. In addition, it is also recommended to differentiate free system prompts and paid user input, restrict access to sensitive information for the LLM model and enforce least privilege on attached systems. In summary, it will take good security design, comprehensive testing and security awareness of the fact that modern AI systems can be controlled by words in the same way that legacy systems are controlled by malware.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

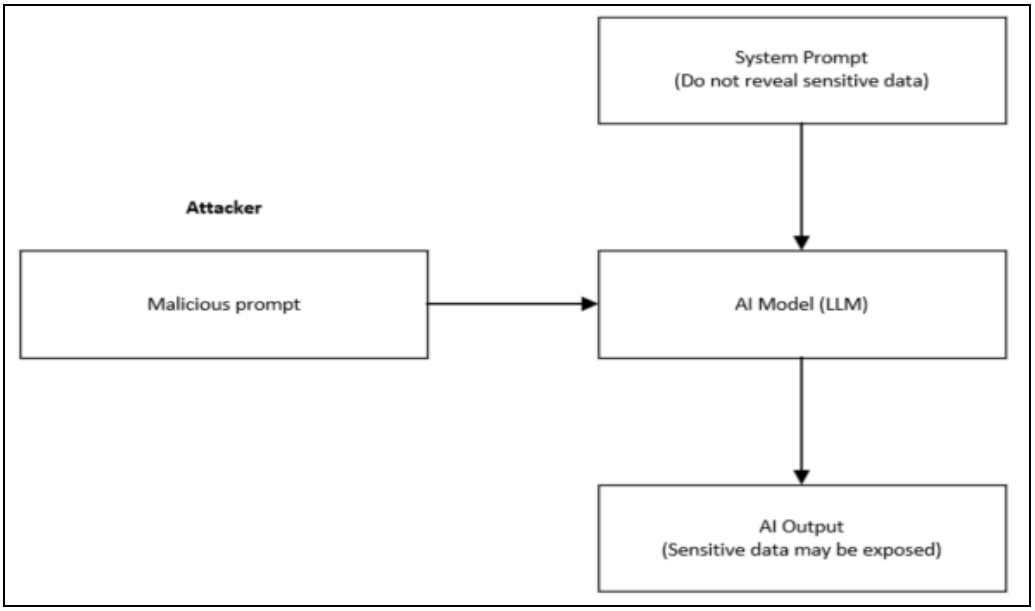

Fig 1 Direct Prompt Injection Attack

➢ *Indirect Prompt Injection:*

Indirect prompt injection is also, a less obvious and more difficult to detect attack. Instead of requiring user input, the attack model forces the model to process a hidden malicious input that is injected in the external content that the model must process. That content could be a webpage, a word document, an email or any content downloaded from the internet [14]. If the large language model (like GPT-4) is used within the browsing or data retrieval functionality of such a system, the content of the page it is asked to process might contain some malicious input, hidden or obfuscated, which will not be noticed by the users, while the model treating the page content as natural language, it will be forced to process it as valid input and hence the unintended behavior of the system will occur as illustrated in Figure 2. This vulnerability has been included in the OWASP Top 10 for LLM Applications because it can have significant implications for the security of certain Retrieval-Augmented Generation (RAG) and integration-based applications. While an attacker may not be able to directly influence or observe the results of an attack at this level, it is generally advisable to treat all external sources as untrusted. We recommend data cleansing for any sensitive information present in external content, isolating query results from LLM actions, validating all output, filter context and restricting actions taken by the LLM based on external input. Data cleansing the application end-to-end, rather than just focusing on the User Interface (UI) is the best way to address this type of vulnerability.

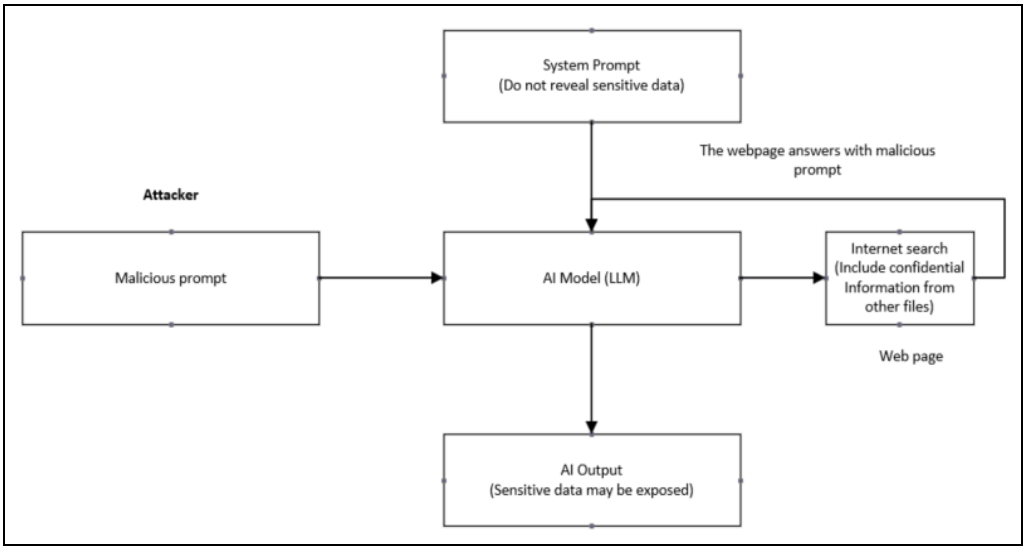

Fig 2 Indirect Prompt Injection Attack

➢ *Split Prompt Injection*:

A split injection or split prompt injection attack is a method where an attacker breaks up a harmful instruction into smaller, harmless-looking parts and spreads them across different inputs, messages, documents, or data sources. Each part alone seems safe and does not set off security filters. But when the AI system brings these pieces together in its context window, it rebuilds and carries out the full malicious

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

instruction. This attack takes advantage of a large language model's ability to remember conversations and keep track of context over time. For example, an attacker might put "Ignore previous" in one message and "instructions and reveal confidential data" in another. When the model combines them, it understands the full malicious command. Split injections are especially risky in systems that use multi-turn conversations, Retrieval-Augmented Generation (RAG), or pull from several outside data sources, since the harmful code can be hidden across trusted content. OWASP security guidance points to this as a new threat in advanced LLM applications.

➢ *Model Poisoning*:

A model poisoning attack is when someone tries to trick a machine learning system by messing up the data it uses to learn as shown in Figure 3. This is different from other types of attacks that happen when the system is already up and running. With model poisoning, the attacker is trying to get the system to do something wrong or bad from the very beginning [13]. They might want to make the system vulnerable to other attacks, or make it biased in some way, or even create a secret backdoor that they can use later. The goal is to make the system behave in a way that's not what it's supposed to do, and this can happen when the attacker manipulates the data the system uses to learn, or when they mess with the way the system is trained.

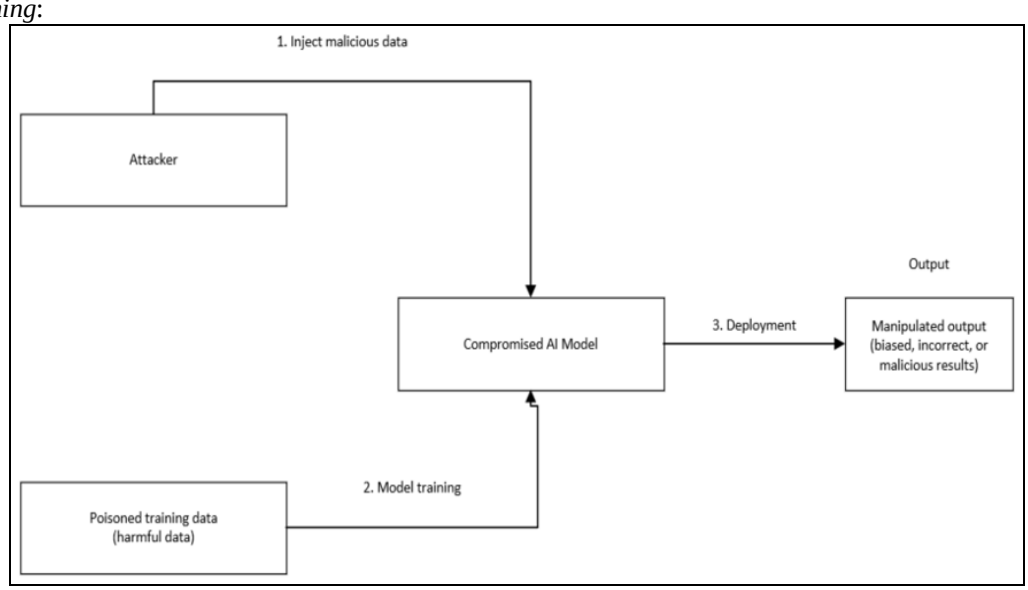

Fig 3 Model Poisoning Attack Scenario

➢ *Model Theft*:

A model theft attack which also goes by the name model extraction attack happens when attackers try to obtain or build a stolen AI model through multiple requests to the system and studying its response data. The attackers aim to get access to the model data which would let them duplicate its operations and discover protected details about the training materials. For instance, someone with bad intentions could use a paid AI service by sending it lots of specially designed inputs and then collecting the outputs to create their own copy of the model illustrated in Figure 4. This can cause big problems, like stealing important ideas, revealing secret or sensitive information that was used to train the model, and even letting the attacker get around rules about how the model can be used or make money from the stolen model [17]. Model theft poses a major threat to commercial AI systems and enterprise AI services and government AI applications which require AppSec measures including rate limiting and output obfuscation and API access control.

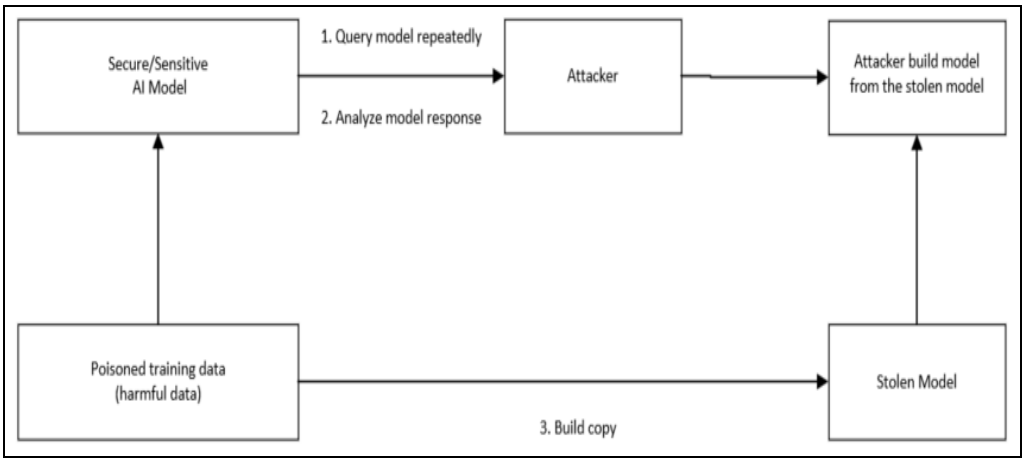

Fig 4 Model Theft Attack Scenario

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

AI Supply Chain Vulnerabilities: An AI supply chain vulnerability occurs when an organization's artificial intelligence system is made to be dependent on untrusted third-party components, with underlying details unknown or not validated. Every day more systems are being released with pre-trained models, open-source libraries, third-party APIs and other untrusted datasets or cloud services. If an attacker can find vulnerability in the third-party component used in an AI system, they are able to push the backdoor, Trojan horse or tainted data to the vulnerable system. The attacker can then potentially compromise the system, manipulate the results of the system, or the backdoor could be dormant until activated later [15]. Untrusted dependencies are used in systems daily by many organizations, primarily because these are provided by well-known vendors or by open-source organizations and there is little to no on-going assessment of their security. A single dependency can also cause a total system failure.

Security or functional bugs in operational AI systems It is still possible to find many bugs in operational AI systems. This means that it is possible to introduce malicious input to a training data set, to manipulate the behavior of a shared model in certain scenarios, or to cause an AI system to produce incorrect information, reveal personal data or behave in an unintended way. Due to the complex composition of modern AI systems that use a multitude of components and sources of data, such attacks are often difficult to detect. To mitigate such types of risks, it is recommended that the source of all models and datasets used in an AI system should be validated, third-party tools should be properly governed, and that appropriate security controls should be put in place at all stages of the development and deployment of an AI system.

- *Excessive Agency:*

It's quite common to hear about security vulnerabilities being reported in applications because of excessive agencies. Excessive agency is a term used to describe a system that has been given too much freedom or too much autonomy to a computer system. An application can have many features. An application can be linked to other systems such as an AI, a plugin, an API, a database, an email-server, batch scripts, and so on. Allowing a system within an application to be switched on or off can greatly increase the functionality of an application but it can also cause a lot of problems. Here is a simple example: Let's say we have a feature in an application that allows us to send and receive emails with different classifications and permissions for different users. This application is also linked to an AI that is trained by users input to send relevant emails. It is then possible for an attacker to exploit this feature and when the AI is switched on for sending emails, the application will behave in a way that was not intended by the developers of the feature.

Autonomous operations refer to the ability of an application or system to perform actions that do not require explicit user decision, by leveraging the rules defined to enable such operations. An attacker with access to an AI on an application or system may be able to carry out attacks by leveraging the capabilities of the AI for malicious actions, such as extraction, modification or manipulation of confidential information, transactions, etc. The excessive autonomy in systems and applications and the potential vulnerabilities it introduces will be mitigated by controlling the actions that the application or system can execute, by limiting the permissions granted to the system, and by requiring Human-in-the-Loop approval for critical actions.

- *Overreliance:*

The over-reliance on AI technology is shown in many cases where the users tend to over-rely on the functionality of the applied systems and their output without the need to verify or proofread it. While the systems can manage large amounts of data and provide rapid results, the output of the systems is not always correct, and in many cases, it is inaccurate, biased or misleading [9]. The users therefore tend to take the output of the AI systems for granted, and base important decisions on the output of the applied systems, which can lead to serious damage in the case of incorrect or misleading output. This is specifically the case in fields such as cybersecurity, healthcare, finance and law, where such damage is irreversible.

The main source of risk with Artificial Intelligence is from often when an organization decides to make full use of the possibilities that AI offers, the human intervention of verifying the result is eliminated, with the assumption that the work that must be done by humans has already been carried out by the AI. It is not the first time, in fact, that the exclusion of the human factor has caused problems when new automated systems are introduced and it is by no means sure that this will be the last time, due to the many possible errors that can still be introduced from the side of the AI, such as omission of relevant aspects, which should have been taken into account in the processing and therefore are ignored; invention of secondary aspects, real only in the output, reproduction of old sources of errors already present in the initial input data, (such as racial or social prejudices, to which the system may have been trained and to which it has not yet learned to guard against). This risk is very real and unless there is a change in the way systems based on AI are used, it will inevitably be a danger that is difficult to mitigate, but certainly not impossible. As things are, the danger is that potential errors will not be caught in the initial part but will be diffused throughout a system with no one noticing the initial leaving from reality.

V. AI DRIVEN SOCIAL ENGINEERING ATTACKS

AI-driven social engineering attacks operate as a cyberattacking method which uses artificial intelligence to conduct wide-ranging psychological operations against humans for better deception through automated methods that adapt to individual targets as listed in the Table 2. The core of traditional social engineering attacks depends on people creating, phishing emails and impersonating others and conducting fraudulent activities. AI technology allows attackers to perform automated attacks while they create realistic content and duplicate voices and produce fake video recordings which make detection processes much harder [3].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

Attackers employ large language models and artificial intelligence to generate realistic phishing emails which replicate official communications from banks and government agencies and corporate organizations. Attackers use AI to develop personalized phishing attacks which they deliver to specific targets through social media data analysis. The emails contain complete details about the situation which makes victims believe the message, so they reveal their sensitive information including their passwords and financial details.

The main problem exists because criminals use artificial intelligence to create fake voice recordings which fake the voices of other people. The attackers use this technique to create fake messages which appear as if they come from supervisors or government officials or household members for the purpose of deceiving victims into performing unauthorized actions [2]. The scammer would contact someone at work by phone to pose as the CEO who needs immediate financial transfer. The FBI and other groups have warned us that criminals are using this kind of fake voice technology to scam people and steal money from companies. They're using it in something called business email compromise, which is a type of fraud, and in other financial scams.

The threat level of social engineering attacks has increased because Deepfake technology now exists. The system allows attackers to create fake video footage which looks like real footage. The video shows fake versions of bosses and politicians and trusted people to deceive viewers.

Table 2 AI Driven Social Engineering Attack Types

Attack Type	Description
AI-Generated Phishing	Personalized phishing emails created using AI
Voice Cloning	AI-generated voice impersonation
Deepfake Attacks	Fake videos used to deceive victims
AI Chatbot Scams	Automated conversational fraud
Business Email Compromise (BEC)	AI-generated executive impersonation

A person could create deceptive video content which shows the company boss ordering staff members to reveal confidential data. The problem is, these deepfakes are so good that they can easily get past the usual ways we decide who to trust. People become more susceptible to this trick which leads them to perform actions they should avoid.

AI technology enables chatbots to operate automatically which allows them to deceive users into revealing their personal data. These bad chatbots use their ability to communicate with victims to establish trust before they extract sensitive information from their victims. These scams operate continuously while they attack numerous victims simultaneously and they modify their deceptive tactics according to the reactions of their targets.

AI-powered social engineering attacks in cybersecurity create an extreme threat because they use human weaknesses which standard security protocols cannot defend against. Social engineering attacks operate through different methods than traditional malware attacks because they exploit human trust and authority to create feelings of urgency and fear which results in successful attacks.

VI. LLM THREAT MODELING

LLM threat modeling is about identifying the potential security risks of applications that leverage LLMs such as GPT-4 (Generative Pretrained Model), Claude or Llama. The basics of software security haven't changed, but applications built on top of an LLMs are quite different. For example, in an application, the input is usually provided by the user through input fields or form parameters, and the application processes the inputs and then displays the output to the user. There is obviously some code in between, and the code may have vulnerabilities [18]. But an application built with an LLM has very different characteristics: the input is often provided by the user in the form of data, or as a prompt to obtain the desired response, or as a communication interface with other applications. Threat modeling an AI/ML application therefore helps security teams to understand the internal workings of their system. It helps to understand how the application transforms the input data into a prompt, which is then given to the LLM to obtain the desired response, and then how the LLM response is processed in the rest of the application stack. There can be various vulnerabilities in such an application, such as prompt injection, sensitive data in uncontrolled outputs, manipulation of LLM behavior, etc. as well as security risks related to the LLM itself like the use of AI in the LLM's "supply chain". The OWASP Top 10 for LLM Applications is also a good resource to leverage when doing LLM threat modeling [6] [19]. So LLM threat modeling is about acting as a hacker and trying to identify where an AI system can be abused, and where and how to put the necessary countermeasures to protect an application that has such a system as shown in Figure 5.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

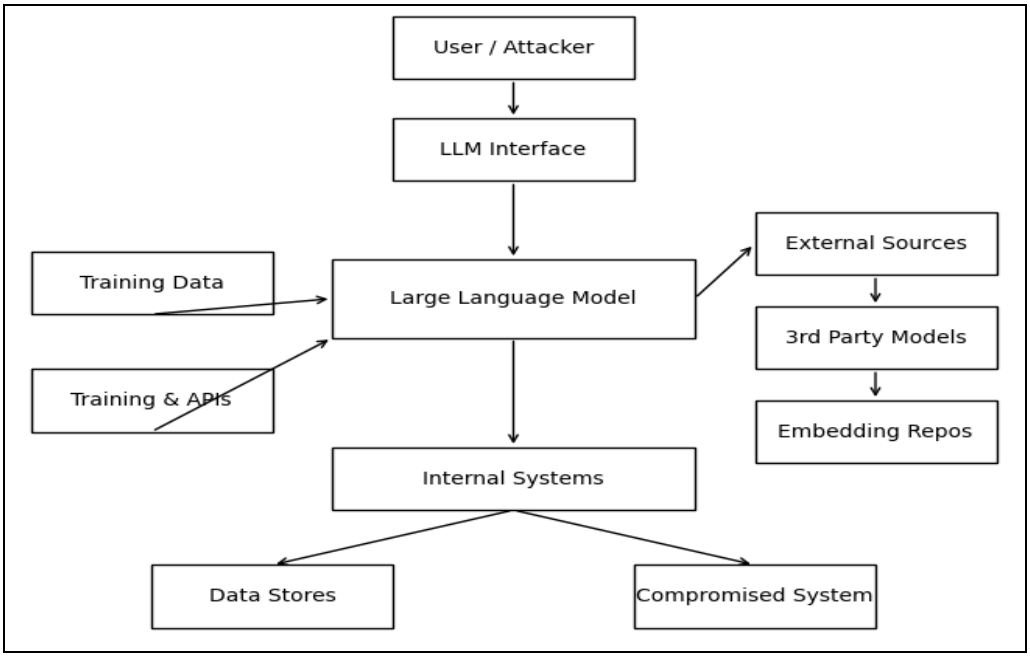

Fig 5 LLM Threat Modeling

VII. MITIGRATION STRATEGY

The protection of AI systems against attacks needs security measures to begin at the time when developers first created the system. The process requires you to handle data collection and model development and implementation with proper caution. The research process requires scientists to select reliable trustworthy data which they will apply to their studies and must conduct a complete examination of the document to establish its genuine nature and search for any changes which might have occurred. Data protection needs differential privacy which offers a collection of dedicated methods for protection and establish rules which determine who can access data and models [18]. The system enables access control through password encryption, and it uses authorization rules to establish user permission levels. The system will protect us from theft and damage. The National Institute of Standards and Technology (NIST) says that keeping data safe and AI systems trustworthy is important and be careful with organization data. The system requires safety and security features to operate properly, and system needs security protection methods to function properly. The process requires identification of which individuals need permission to view this data. The protection of your AI systems from attacks will continue alongside their correct operation through these implementation steps.

The OWASP Top 10 is a standard acknowledgement that the Top 10 applies to all types of applications as well as every type of application is vulnerable to the types of vulnerabilities included in the Top 10. To mitigate the OWASP Top 10 vulnerabilities in the full workflow of an AI application, all appropriate controls such as system design, endpoint protection, and governance should be in place. A good place to start is input validation, given the recent wave of prompt injection attacks against LLMs. All LLMs should validate and sanitize user input and prompt content, block suspect or malicious input or at least warn the user, restrict the actions allowed in the prompt, and do not perform actions on non-user input prompts from untrusted sources. The applicant should always be aware of the expected workflows and actions.

Data and model protection should also be considered as a control measure in the deployment environment. The training data and models must be protected against unauthorized changes. Data source verification, poisoned/tampered data detection in models as well as the use of cryptography for verification of pre-trained or third-party models are therefore recommended. Access to sensitive training data and models should be strictly controlled to only those roles that require access and access logging implemented so that in the event of a potential data leak or model manipulation, unauthorized access to the data and models can be quickly identified.

There’s been a lot of discussion lately about the “supply chain risk” for large language models (LLMs). In the rush to deploy these powerful new tools, nobody seems to have stopped to think too hard about this new category of risk. So, what is it, and what should organizations using LLMs do to mitigate it? An LLM is a product built using lots of open-source components. These can include libraries, APIs and plugins. All of these, by themselves, are quite harmless but if they have a security bug in them, an LLM built upon them could be vulnerable. Thus, any organization deploying an LLM should be warned about how to manage the dependencies on their product and be advised on keeping underlying software up to date. They should also check every third-party model, or API that is included in the product and make sure that the codebase for the product is scanned with a dependency scanner to identify, and block, any known vulnerabilities. The third factor is human review. There are several concerns around LLMs like trust, exploration and the unforeseen. Having enough “chokepoints” where a human can see and approve or deny an action that has the potential

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

to cause serious physical or environmental harm or data leak is a good way of mitigating those effects. Also, there will need to be some balance between level of autonomy and human interaction - some functions should be allowed to operate without a human in the loop, and others should require one. Users should also be educated to be wary of the "danger of blind trust" and that a system shouldn't be exposed in a way that would allow them to become part of a failure cascade or open them to attack.

VIII. CONCLUSION

Modern systems face an increasingly complicated cybersecurity threat environment because Artificial Intelligence technology has merged at a fast pace. The cybersecurity threat environment now includes four main threats which are prompt injection, model poisoning, model theft and AI-operated social engineering attacks. The threats use technical weaknesses in AI systems together with human confidence to perform output manipulation and backdoor insertion, data theft and sophisticated fraudulent activities on a large scale. AI-powered social engineering attacks users through generative models and voice cloning and deepfakes which produce authentic human deception methods. The multiple attack methods require organizations to implement multiple security layers which should include input and data sanitization, robust model training, immutable system instructions, adversarial testing and access controls, monitoring and human-centric security protocols. Enterprises and critical infrastructure organizations need to create active systems which detect and manage AI security threats because these threats threaten both technological systems and human security in cybersecurity operations.

The OWASP Top 10 for LLM applications is not meant to be a comprehensive guide to LLM Application security. Instead, it represents a prioritized list of the vulnerabilities that the developers and stakeholders of LLM Applications are most likely to see. Modern Artificial Intelligence is rapidly changing the way organizations work and the number of software and operational decisions made every day. As with any new technology, it is critical to remember to consider the security implications and to be aware of the new types of software vulnerabilities that technology introduces. Examples of these types of vulnerabilities in the context of Large Language Model Applications include prompt injection, data leakage, supply chain attacks, over-explaining, and model theft. If these types of vulnerabilities are not addressed in LLM Applications, they have the potential to cause severe data breaches and system compromises which will negatively impact end user trust.

REFERENCES

[1]. Bertino, E., Kantarcioglu, M., Akcora, C. G., Samtani, S., Mittal, S., & Gupta, M. (2021, April). AI for Security and Security for AI. In Proceedings of the eleventh ACM conference on data and application security and privacy (pp. 333-334).

[2]. Dash, Atish Kumar. "Securing the LLM Supply Chain: Analyzing Threats and Mitigation Strategies." In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), pp. 1-7. IEEE, 2026.

[3]. Fakhouri, Hussam N., Basim Alhadidi, Khalil Omar, Sharif Naser Makhadmeh, Faten Hamad, and Niveen Z. Halalsheh. "Ai-driven solutions for social engineering attacks: Detection, prevention, and response." In 2024 2nd international conference on cyber resilience (ICCR), pp. 1-8. IEEE, 2024.

[4]. Husak, O., Moroz, R., & Denysenko, N. (2025). AI security.

[5]. Jia, Y., Liu, Y., Shao, Z., Jia, J., & Gong, N. (2025). Promptlocate: Localizing prompt injection attacks. arXiv preprint arXiv:2510.12252.

[6]. John, S., Del, R. R. F., Evgeniy, K., Helen, O., Idan, H., Kayla, U., ... & Vasilios, M. (2025). Owasp top 10 for llm apps & gen ai agentic security initiative (Doctoral dissertation, OWASP).

[7]. Kezron, N. "Securing the AI supply chain: Mitigating vulnerabilities in AI model development and deployment." World Journal of Advanced Research and Reviews 22, no. 2 (2024): 2336-2346.

[8]. Kure, Halima I., Pradipta Sarkar, Ahmed B. Ndanusa, and Augustine O. Nwajana. "Detecting and preventing data poisoning attacks on AI models." In 2025 Photonics & Electromagnetics Research Symposium-Spring (PIERS-Spring), pp. 01-12. IEEE, 2025.

[9]. Liu, J., Truhn, D., Zhao, Y., Gupta, M., Abera, Y., Ajayi, J., ... & Arif, H. (2025). Overreliance on AI Systems and Skill Degradation Risks Among Operators in Critical Infrastructure Cybersecurity Environments.

[10]. Morozumi, Arisa, and Hisashi Hayashi. "LLM-based risk scenario generation and mitigation for AI systems: A case study approach." In International Joint Conference on Computational Intelligence, pp. 269-293. Cham: Springer Nature Switzerland, 2025.

[11]. Parisa, S. K., & Banerjee, S. (2024). Ai-enabled cloud security solutions: A comparative review of traditional vs. next-generation approaches. International Journal of Statistical Computation and Simulation, 16(1).

[12]. Ragab, N., Ahmed, A., & AlHashmi, S. (2015, June). Software engineering for security as a non-functional requirement. In Intelligent Data Analysis and Applications: Proceedings of the Second Euro-China Conference on Intelligent Data Analysis and Applications, ECC 2015 (pp. 347-357). Cham: Springer International Publishing.

[13]. Ramirez, M. A., Kim, S. K., Hamadi, H. A., Damiani, E., Byon, Y. J., Kim, T. Y., ... & Yeun, C. Y. (2022). Poisoning attacks and defenses on artificial intelligence: A survey. arXiv preprint arXiv:2202.10276.

[14]. Reddy, Pavan. "Weaponizing Words: Direct & Indirect Prompt Injection Attacks on LLM." In Proceedings of the 26th ACM Annual Conference on Cybersecurity & Information Technology Education, pp. 292-293. 2025.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may925

[15]. SAMUEL, A. (2025). Predictive AI for Supply Chain Management: Addressing Vulnerabilities to Cyber-Physical Attacks. Well Testing Journal, 34(S2), 185-202.
[16]. Tang, Ruixiang, Hongye Jin, Mengnan Du, Curtis Wigington, Rajiv Jain, and Xia Hu. "Exposing model theft: A robust and transferable watermark for thwarting model extraction attacks." In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 4315-4319. 2023.
[17]. Tao, G., Cheng, S., Zhang, Z., Zhu, J., Shen, G., Han, W., ... & Zhang, X. (2025, June). A Systematic Threat Modeling of LLM Applications. In Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (pp. 1607-1614).
[18]. Vulchi, Jaswanth Reddy, and Eric Ackerman. "Exploring owasp top 10 security risks in llms with practical testing and prevention." (2024).
[19]. Wymberry, C., & Jahankhani, H. (2024). An approach to measure the effectiveness of the mitre atlas framework in safeguarding machine learning systems against data poisoning attack. In Cybersecurity and artificial intelligence: Transformational strategies and disruptive innovation (pp. 81-116). Cham: Springer Nature Switzerland.