University of Bahrain

College of Information Technology

Department of Information Systems

Developing an Agentic AI Vulnerabilities Framework

Prepared by

Mohamed Husam Mohamed Darwish 202208375

Kaltham Abdulla Mohamed Basalar 202210258

For

ITCY 499

Senior Project

Academic Year 2025-2026-Semester 2

Project Supervisor: Dr.Yaqoob Salman Mohamed AlSlais

12-5-2026

Abstract

The emergence of agentic artificial intelligence (AI) presents the next iteration of intelligent systems, characterized by their ability to think independently, make use of memories, make decisions and execute external tools. As much as the benefits associated with these qualities include better automation and increased effectiveness, these agentic systems also present unique cybersecurity vulnerabilities that are different from what is experienced in regular AI systems or software in general. In this paper, a framework for examining the vulnerabilities in agentic AI systems is proposed, where security threats are analyzed using five levels of operation namely Interaction, Context & Memory, Reasoning, Tool & Execution and Governance & Trust. Moreover, this research analyzes the concept of cross-level vulnerability propagation that occurs at more than one stage within an agentic system as opposed to just being contained within one part. For purposes of evaluating the suggested framework, qualitative feedback was collected using an interview aimed at cybersecurity professionals and those in the AI industry. Findings have shown that the proposed framework and concepts such as vulnerability propagation, prompt injection and others are highly relevant and supported by respondents.

Acknowledgments

First and foremost, we are immensely grateful to Allah Subhanahotu Wala Ta'ala for blessing us with the strength, dedication, knowledge, and determination to undertake this research project successfully. This could not be achieved without his blessings.

Secondly, we would also like to thank our mentor, Dr. Yaqoob Salman Mohamed Alslais, for his unwavering support and advice throughout the course of undertaking this research project. His guidance was invaluable for creating this research in the best possible manner.

In addition to this, we are immensely grateful to those experts and individuals who spared their valuable time to analyze and provide their expert opinion regarding our framework design through the interview.

Lastly, we would like to extend our sincere gratitude to our families, friends, and colleagues who motivated us throughout the course of carrying out this research research.

Table of Contents

List of Figures

Chapter 1
Introduction

As time passes, Artificial Intelligence (AI) evolves, and what we have used to think about is impossible to happen, it started to happen or even already happened with nowadays technology. As AI agents started to appear, it left a remarkable mark on the technology field. AI agents are autonomous systems that can perform tasks beyond what is required to do with the least human involvement. It can perform planning, decision-making, and even act all on its own by integrating Large Language Models (LLMs), which LLMs are trained on a huge amount of data and built on machine learning. Despite that AI agents took a large load of problems and solved them, such as digital assistants that have access to emails and calendars for scheduling or even lands a hand in coding environments and scrips, it still has its own vulnerabilities that are being faced every now and then that needs to be considered.

1.1 Problem Statement

As already noted above, autonomous AI agents rely on substantial amounts of data, which brings not only high efficiency as one of their advantages but, at the same time, creates some additional security threats that push traditional cybersecurity systems beyond their limits.

The primary issue related to AI agents is that they are largely based on instructions written in natural language, use externally sourced data, and have a reasoning algorithm embedded within the system. Such characteristics make AI agents vulnerable to various malicious activities since the attackers might manipulate an AI system making it perform incorrectly.

In addition, current measures of ensuring the safety of AI technology concentrate mostly on securing machine learning-based algorithms without paying any special attention to autonomous AI agents that interact with the external world. Due to this fact, there is little information available about vulnerabilities associated with agentic AI agents.

The lack of proper AI agents' security frameworks and threat detection methods emphasizes the necessity of carrying out further research on the matter. In particular, the main aim of the project

under discussion will be to investigate vulnerabilities of agentic AI agents and examine possible threats caused by their autonomy.

1.2 Project Objectives

The ultimate goal of this research is to analyse security flaws in AI agents along with the dangers related to them in actual world scenarios. To achieve this goal, this research aims to comprehend the architecture and working principles of AI agents, gain knowledge about various vulnerabilities that can pose security threats to AI agents, and learn about the most prevalent attack types that can occur against AI agents, including prompt injection attacks, manipulation attacks, and data poisoning attacks. Furthermore, the research seeks to examine the already existing research works and frameworks related to the security aspect of AI agents and to identify the loopholes in the current research scenario along with potential ways to mitigate these vulnerabilities.

1.3 Relevance/Significance of the project

The fact that AI agents are needed everywhere has led to a greater need to ensure their security and trustworthiness among all fields. This is because the more the use of AI agents increases, in such needs as financial services, cybersecurity tools, and enterprise automation tools, the more it leads to real security issues in these systems, exposing it all to some serious consequences that needs to be handled.

As we have mentioned earlier, most studies on AI security have focused on attacks on machine learning models, such as adversarial examples and training data poisoning. However, AI agents are complex systems that not only depends on machine learning models but also uses tools and interact with their different environments.

Thus, it is very important to understand the security issues related to AI agents, and this research is a contribution to the growing body of knowledge in AI security by pointing out the possible security issues in agent-based systems and areas where further research is required to be taken, to ensure the security of AI agents under different circumstances.

1.4 Research Methodology

This research utilizes a qualitative methodology by reviewing the related literature and seeking expert evaluation. The activities performed in this research involve a literature review of previous works about the architecture of AI agents, their vulnerabilities, and possible attack scenarios, as well as the current state of research with regard to security frameworks and guidelines applicable to such systems. On the basis of the gaps in existing studies, a five-tiered framework of evaluation of agentic AI system vulnerabilities was proposed, which includes the Interaction Layer, the Context and Memory Layer, the Reasoning Layer, the Tool and Execution Layer, and the Governance and Trust Layer. After the framework design, seven research propositions to validate its practical relevance, structural consistency, and applicability were formed. For that purpose, a qualitative interview with open questions was created and distributed to four deliberately chosen participants with relevant professional backgrounds in cybersecurity, artificial intelligence, cloud technology, and software engineering. Their responses were analyzed through thematic analysis to find common themes and inconsistencies.

1.5 Scope and Limitations

The research addresses security vulnerabilities in agentic AI systems incorporating large language models and decision-making skills. Several attack techniques are reviewed, such as prompt injection attacks, tool-based attacks, memory poisoning attacks, and decision manipulation attacks.

It should be emphasized that the development of a prototype of an AI agent or the creation of a working security tool is not within the scope of the current study. Rather, the main attention is paid to a systematic literature review, the analysis of security risks in agentic AI systems, and the introduction of a theoretical framework for their evaluation consisting of five layers.

Finally, it should be pointed out that advances in AI technology occur regularly, and thus new vulnerabilities and attacks on AI agents can be anticipated in the future. This fact means that the results and recommendations outlined above should be treated as relevant only in the present context and might need updating over time.

1.6 Report Outline

The report is divided into six chapters:

- Chapter 1 is the introduction of the research topic, the problem statement, the objectives of the project, and the importance of researching the vulnerability of AI agents.
- Chapter 2 is the literature review of the research that has already been conducted on the topic of AI agents and the security issues that surround them.
- Chapter 3 is the explanation of the research model and the propositions that were developed as part of the research.
- Chapter 4 is the explanation of the methodology that was used in the research.
- Chapter 5 is the explanation of the results that were obtained from the research.
- Lastly, Chapter 6 is the summary of the conclusions that were drawn from the research and the recommendations that can be made on the topic.

Chapter 2
Literature Review

2.1 Introduction

Rapid development of the technology of AI has led to the emergence of a need for the development of agentic AI systems that possess the ability to plan, reason, and perform in constantly changing environments. In contrast to typical LLMs, agentic AI systems integrate their reasoning abilities with the use of utility from the external world, memory, and the interaction with the environment. The agentic AI system has the potential to obtain information from external resources, perform tasks, give commands, and interact with digital devices without constant human supervision (Deng, 2025).

Despite the added advantages brought about by the increased autonomy of agent-based AI applications in different fields such as cyber security, software engineering, and automation, among others, they are currently facing new challenges in terms of security. The operating environment for agent technology is dynamic and includes handling of malicious input and interaction with external systems; therefore, it exposes it to being exploited. Some of the current major security threats include prompt injection, tools abuse, memory poisoning, and privilege escalation (Khan, 2024) (Chiang, 2025) (OWASP, 2025).

The key point of carrying out the literature review is the analysis of the existing knowledge concerning the vulnerabilities of agentic AI and the identification of the significant risks found by other researchers during their investigations. The analysis will also cover the methodologies used for measuring the level of security of the autonomous artificial intelligence agents. This chapter will provide an opportunity to summarize the existing state of knowledge, identify its shortcomings, and lay the foundation for further investigation.

2.2 Evolution of Artificial Intelligence Agents

AI agent development may therefore be described in terms of the transition from static systems that possessed specialized computing powers to self-contained systems that reason and have the ability to plan, communicate, and act within an environment that is always changing. As far as early ideas concerning AI agents go, intelligence was frequently perceived as task-specific, with particular tasks such as classifying, predicting, optimizing, and automating according to certain rules. They were highly effective when put to work for their designed purposes, but they were unable to comprehend ambiguous goals, devise a strategy for achieving them, and react to a constantly shifting setting. Modern studies on agentic AI indicate that today's AI agents have advanced considerably past previous AI agents due to their inclusion of linguistic understanding, decision-making capabilities, memory retention, and action (Flehmig, 2025) (Sapkota, 2025).

An additional factor, which is of importance when assessing the significance of the above-mentioned evolution, is that it represents a change in the expectations for the AI system. As compared to the previous versions of AI, the newer systems are supposed to not only respond to the goals and carry out the actions depending on the data provided, but to be capable of setting themselves high-level goals, to decompose them into sub-goals, to identify necessary preconditions and to take a series of steps to accomplish all of those things. This way, an AI system becomes active rather than passive in the work flow process. The concept taxonomy article on AI agents versus agentic AI makes the same statement about the changes (Sapkota, 2025).

2.2.1 From traditional AI systems to early agents

In this context, the first phase of evolution can be interpreted, in my view, as non-agentic AI, since at this point in time AI had already been designed to solve specific problems, instead of being conceived as an independent agent. Such systems were able to perform well in pattern recognition, prediction, and making decisions based on a particular rule. However, they still did not possess any form of a persistent state, capacity for interactive task selection, or independence while exploring the environment. As per the literature supplied by you, this very era is highlighted as a historical predecessor to the present-day one, known as the age of agentic AI. The significance of such a comparison is clear: in addition to better models, it implies something novel for AI in general (Sapkota, 2025) (Flehmig, 2025).

The next step brought about agents in a somewhat classical understanding of the term, with the capacity for environmental perception, decision making based on internal logic, and goal achievement. At this level, the agent was perceived not only as a prediction mechanism but also as a being that can perceive, decide, and act. Nevertheless, most of the previous generation of agents could work in relatively structured settings and depended on specific mechanisms of decision making or logic. While the survey acknowledges the importance of the previous step in the development of agents, it notes that those agents have little to do with the current generation of agents, who are capable of text-based reasoning, communication through language, and flexible action under various circumstances (Sapkota, 2025) (Abou Ali, 2025).

The main feature that sets the early computational agents apart from current agents is, in fact, greater freedom to interact and reason about tasks. In fact, early agents were typically designed on the basis of pre-set decision flows. Modern-day agents, on the other hand, have proven their ability to comprehend natural-language requests, reason over large texts, use external utilities, and adjust their behavior accordingly. This means that the evolution is no longer solely technological; there is also an evolution of the architecture itself (Sapkota, 2025) (Flehmig, 2025).

2.2.2 The role of large language models in this evolution

The development of large language models became a significant event in the development of AI agents since it greatly improved the ability of AI agents to analyze inputs, understand context, and reason on numerous types of texts. In contrast to previous generations of AI, large language models have significantly boosted the abilities of understanding and generating natural language commands, performing summaries, reasoning, and other similar functions. The role of large language models in this regard is undoubtedly mentioned in the literature you provided as one of the pivotal enablers of the era of agentic AI agents (Flehmig, 2025) (Sapkota, 2025).

What is important to remember from the literature review is the statement that even if LLM technology can be part of some software, it does not necessarily mean that it will be called an agent because of this. By itself, the technology of language generation is not associated with any type of autonomy at all. For LLM to evolve further and turn into an agent, it should be put into a larger system, which is able to perform actions and plan actions related to particular goals. Thus, it can be said that in this case, the goal of LLM was to provide cognitive capabilities for an agentic agent (Sapkota, 2025).

The significance of this distinction arises from the reason for which the development of AI agents is more than the development of models. The literature indicates that this evolution involves many layers, where language understanding and general cognition emerge on one side while decision-making and actions occur on another side. All these together lead to the creation of present-day agents. As described by Nisa et al., this evolution leads to the entry into the "age of reasoning" due to the emergence of planning-based agentic agents (Flehmig, 2025) (Hamilton, 2026).

2.2.3 From AI agents to agentic AI

One of the most crucial distinctions drawn in recent writings is between AI agents and agentic AI, which are somewhat similar but distinct notions. In the article entitled “AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges,” it is said that “the use of the expression ‘AI agent’ is often generalized to include any technology that has the ability to carry out some actions on the user’s behalf,” while “the term ‘agentic AI’ implies a greater autonomy, iterative decisions, and proactive execution of tasks.” This distinction is helpful since it makes clear that the transition is not an either-or development but rather a gradual one (Sapkota, 2025).

Based on this strand of research, even the present-day AI applications referred to as “agents” have limited scope for autonomous decision-making, as the human user directly controls the actions at almost all stages of the operation. On the other hand, agentic AI is a step up from this, since it will be able to understand its objectives, develop a strategy, adjust its approach depending on any new data, and execute a sequence of actions without human intervention. Hence, the shift towards agentic AI is a shift towards proactive action (Sapkota, 2025) (Flehmig, 2025).

The increased level of initiative will also alter the way the system operates in practice. A typical AI agent can summarize content or answer queries. An agentic system will be able to research, invoke external services, manage several tasks, and deliver the final output. In the literature, the development of AI to agentic AI is thus seen as moving from performing tasks under strict supervision to executing actions under loose supervision. Indeed, this is exactly the step that leads to the security issues that arise in your thesis statement (Deng, 2025).

2.2.4 Core architectural developments in modern agents

The development of artificial intelligence agents has been accompanied by architectural evolution as well. Contemporary artificial intelligence agents are distinguished from those created in earlier times not only due to the advancement of their models, but also thanks to architectural improvements. Some of the most essential elements that contemporary artificial intelligence agent architectures include have been discussed frequently in the existing literature; those include planning, memory, tool usage, perceptual abilities and iterative reasoning. All these elements enable agents to engage in multi-stage problem-solving (Flehmig, 2025) (Abou Ali, 2025).

Another significant architectural innovation in this area is planning. Planning enables the system to break down its goal into lower-level actions that are executable. Rather than simply reacting once to the environment, planning enables the system to determine what needs to be done first, what information needs to be collected, what tools need to be applied, and what comes next. In the surveys, this represents another fundamental breakthrough because the concept of planning adds a temporal dimension to the AI system (Hamilton, 2026) (Flehmig, 2025).

Another critical feature is the enhancement of memory capacities. Traditional systems used to consider each engagement separately and/or maintained limited information on prior sessions. As far as agentic systems have been concerned, memory features become an indispensable part allowing previous experiences, task histories, and contextual factors to affect subsequent engagements. While making agents smarter and increasing their capabilities in performing complex and lengthy operations, memory capacities can result in accumulating stateful data, retaining errors, and transferring contextual influences between periods. Therefore, the issue of memory is discussed not only as a means of increasing abilities but also as a potential security problem (Flehmig, 2025) (DENG, 2025).

Integration of tools is also vital for agent development. When a language model has the capability to search for information from the Internet, call an API, access a document, retrieve data from a database, or do something else, then the model becomes much more useful in practical applications. The two surveys about agentic systems and distributed autonomous agents highlight that the ability to use tools turns the artificial intelligence system from a mere interpreter into an executor. One of the best indicators of contemporary agents is that they are not merely language generators anymore (Abou Ali, 2025) (Deng, 2025).

The role of environmental perception and interactions is another crucial factor. Many of the contemporary applications of agents, particularly those that are based on the internet and other types of distributed systems, require the system to detect changes in state, receive feedback from outside, and respond accordingly. Thus, the agent is not just a computational entity but an interactive one, which is an important aspect of the transition from software automation to agentic AI (Abou Ali, 2025) (Sapkota, 2025).

2.2.5 The evolution of autonomy

One might say that autonomy is the most important element in the development of AI agents. In academic literature, autonomy is not regarded as a binary trait, but rather as an evolutionary scale. On one side of the scale are agents that depend almost exclusively on human input and follow narrowly defined tasks without much room for independence. On the other side are those who are able to understand general goals, come up with tactics, adjust to new data, and operate independently without too much involvement from humans (Flehmig, 2025) (Sapkota, 2025).

Autonomy also has a lot to do with the interplay between planning, memory, and instrumental action. When a system gains autonomy, it is capable of determining what information it requires, obtaining that information, remembering its previous context, and making choices from several available options. In terms of autonomy, the literature makes it clear that the issue does not only have to do with independence from humans but with the ability of systems to sustain their own goal-directed activity. Hence, the concept of agentic AI is viewed as a new level of development (Flehmig, 2025) (DENG, 2025).

On the other hand, as indicated by previous literature, increased autonomy brings about uncertainty and control problems. An autonomous system can become more efficient and useful, but it also becomes less predictable and difficult to control. This will become an important problem for your thesis since the same evolutionary traits that enable agents to become strong also make them weak. According to Deng et al. in AI Agents Under Threat, the introduction of autonomy, environment interaction, and actions through tools increases the attack surface from that of standalone systems (Deng, 2025).

2.2.6 Multi-agent and distributed evolution

Another crucial stage of the formation of AI agents is the shift from autonomous AI agents to multi-agent systems. As the term implies, at this stage, agents are expected not to be regarded as autonomous agents only but as elements of a group where they act and interact together. The application of the concept of AgentAI in this regard is especially appropriate since the survey of AgentAI shows that autonomous agents have become widely discussed in connection with distributed AI and Industry 4.0 (Abou Ali, 2025) (Saleh, 2025).

It should be stated that the evolutionary process affects not only the capabilities but also risks associated with the use of AI. To put it differently, an agent can perform the functions on behalf of one person while the group or collection of agents can perform them more effectively because of the possibility to distribute responsibilities, share context, and implement more complicated algorithms. However, at the same time, this process introduces some additional features, making an agent more sophisticated and intricate (Abou Ali, 2025) (Giusti, 2025).

As far as conducting a literature review is concerned, its advantages include the fact that it proves that the development of agents is not related to their intelligence alone but is associated with the system in which they exist. Today, agents are designed and implemented so that they can be used in a hierarchical environment, infrastructure, and cooperation (Giusti, 2025) (Deng, 2025).

2.2.7 Why this evolution matters for the present study

The study of the evolution of agentic AI agents is very critical to our current investigation since vulnerabilities in agentic AI cannot be understood properly unless we understand how agents developed their autonomy and capability of interacting with the environment. Unlike the classic predictive model, an agentic AI is at risk due to its capabilities, which are different from those of other predictive models. From the studies carried out, it is clear that the development of AI from a narrow model to interactive and agentic model came with different security challenges (DENG, 2025).

It follows that the evolution discussed in this section is more than just background information. Instead, it lays the groundwork for the rest of the literature review. The following discussion about prompt injection, interaction attack techniques, memory flaws, and framework design all depend upon the fact that modern AI agents have moved beyond being passive models; they are actors. In doing so, the literature provides a clear justification for our claim: the evolution of artificial intelligence agents necessitated agentic AI security (Flehmig, 2025) (Deng, 2025).

2.3 AI Agents architecture

Before analyzing the weaknesses of AI agents, it is important to consider the real foundation of AI agents, their architecture, composition, memory management, and interaction with the outer world. These factors will define the extent to which the agents can independently make decisions. In this part, all that will be discussed will be taken into account so that we can create an overall picture of the agents' foundation.

AI Agent System represents a complex system design that surpasses the typical language model system as it includes various components that make the system independent. Unlike conventional AI systems that solely operate under the input/output design, AI agents are characterized by features such as reasoning and planning (Zhan, 2024) (Debenedetti, 2024).

An AI agent comprises of different modules, where a critical component is the reasoning module that generally constitutes a giant language model along with others, including the memory module, tools module, and environmental interactions. It is through the combination of these different components that the AI agent will be capable of sensing, analyzing, and acting in a cyclic manner.

2.3.1 Core components of AI agents

Some basic building blocks that are included in the architecture of AI agents include those that help them become autonomous. First of all, there is agent core that can be described as a large language model capable of reasoning, decision-making, and reaction. . In other words, agent core serves as its brain that will process the information received and make decisions about the action to be made. (Zhan, 2024).

Memory units are also important elements of AI agents' architecture. They serve the purpose of storing context information generated because of interactions between agents and users. The use of memory units helps to generate effective decisions based on the agent's past experiences. At the same time, memory units can be subdivided into short-term and long-term ones. (Chiang, 2025).

Tool interface should be implemented by the developer to provide the opportunity for interaction with other sources such as databases and execution environments as well as APIs. With the help of the aforementioned tool interface, the agent will have opportunities to do more than just generate the text. For example, it can be used to retrieve information and run programs.. (Debenedetti, 2024).

Environment interface serves the purpose of enabling access to input information from the external environment. This can be done through such processes as accessing the internet and receiving commands from users. On the one hand, it increases the complexity of the agent as well as risks involved in its functioning. On the other hand, it opens more perspectives for the agent. (Khan, 2024).

2.3.2 Perception-Reasoning-Action cycle

The principle behind the design of AI agents' architecture lies in the perception-reasoning-action paradigm. The principle entails the interaction between the agent and the environment through perception, reasoning, and action (Debenedetti, 2024).

Primarily, the process of perception involves the acquisition of data about the environment of the agent from user interaction, outside sources, or the output of other agents (Debenedetti, 2024). Following this, the gathered data is analyzed by the agent through the reasoning process, and it becomes possible to find the most effective actions. In some cases, the reasoning process involves multiple steps, and an agent needs to perform different actions to achieve the desired goal.

Finally, within the action phase, the agent executes actions selected in the preceding stage. The output information is delivered to the agent and can serve as input for additional processing.

In one sense, the perception-reasoning-action loop plays a vital role in autonomous actions taken by AI agents. In another sense, the repetitive character of this process entails new security threats (Khan, 2024).

2.3.3 Role of external tools and APIs

In this connection, the integration of external tools becomes necessary to enhance the capabilities of AI agents beyond performing linguistic tasks. With the help of these tools, the agents are capable of accessing different external sources such as databases, websites, computational tools, and even software programs. (Debenedetti, 2024).

As a result, not only can AI agents be used to create text but also perform various actions. It means that an AI agent will be able to use the database for gathering relevant information, calculate some things using particular applications, and go online to gather more information.

However, using external programs poses significant risks in terms of security. Each program is a separate entity that requires certain parameters for proper functioning. The issue is that there

are no norms for validating input/output. It means that malicious information can be sent to the system (Chiang, 2025).

2.3.4 Memory and context management

Contextual memory is an important element that should be considered in the AI agent construction because contextual memory can help retain knowledge of previous interactions and use this knowledge while interacting with a user. Unlike traditional models where the history of interactions cannot be saved, artificial intelligence agents can store information about themselves as their memory and make better decisions based on this information (Liu, 2025).

Typically, there are two types of memory that can be identified within the architecture of AI agents. Short-term memory serves to store all required data related to a certain interaction. On the other hand, long-term memory stores all prior experiences of an AI agent, helping the latter learn.

The proper handling of context becomes essential for the efficient operation of an AI agent as it has a direct effect on the actions of an AI agent. However, it also poses substantial threats, such as the accumulation of erroneous information generated by the agent before.

2.3.5 Interaction with external environment

AI agents perform in an ever-changing setting whereby they are continually communicating with programs and sources of data external to themselves. This communication may entail obtaining information from the Internet, processing user instructions, and exchanging data with other programs (Khan, 2024).

Communication skills also allow AI entities to perform more complicated tasks, particularly those demanding current information. However, there will always be a certain amount of risk regarding information gathered from outside sources because it might be erroneous and even false.

In contrast to closed conventional systems working with prearranged input data, AI entities operate with input data coming from different sources that are difficult to verify.

2.3.6 Autonomous planning and decision-making

The key factor of agentic artificial intelligence is that it can plan autonomously. This means that the agent can break down the task, choose the right approach, and perform the action to achieve its objective (Zhan, 2024). This implies that such kinds of agents can act independently without human intervention at all.

During the planning process, the agent makes a plan, evaluates different possible actions, and selects an optimal one. The fact that decisions can be made dynamically makes actions more flexible in relation to the environment.

At the same time, this feature brings about some difficulties too. Firstly, when actions and decisions are made dynamically, it becomes difficult to predict future actions of the agent (Khan, 2024). For that reason, there may be some negative implications from autonomous planning.

2.4 Security Challenges in Agentic AI Systems

The development of agentic artificial intelligence systems has created a new set of problems that cannot be considered similar to problems of regular software systems at all. The fact is that, unlike regular software applications, which work in predetermined environments, agentic AI systems are developed with regard to their ability to work with external tools, data, and digital environment. As a result, a completely new set of threats has emerged related to the interactions between language, reasoning, and actions taken based on such interactions (Deng, 2025).

Prompt injection attacks are among the various threats posed to agentic artificial intelligence (AI). According to the existing literature, prompt injection attacks are not simply one type of attack among others, but rather an essential attack that leads to many other downstream attacks such as data exfiltration, action hijacking, and workflow interference. Prompt injection attacks target the primary mechanism by which agentic AI systems comprehend instructions and make decisions (Yorke, 2024).

2.4.1 Expansion of the Attack Surface in Agentic AI

The shift from traditional AI to agentic AI has drastically altered the way in which the attack surface operates. Traditional AI would take inputs that were always in a structured form and produce outputs that would always be in the same format. On the other hand, agentic AI takes in natural language inputs that are not structured, retrieves data from external sources, interacts with APIs, and executes workflows (DENG, 2025).

One of the main components of this increased attack surface is the dependence on unverified data. The model presented in AgentDojo includes an agent acting on unverified data acquired via tools, proving that contemporary agents are supposed to deal with data coming from an unreliable source. As a result, the system constantly deals with interpreting data that might be manipulated at any time (Yorke, 2024).

The extended interaction domain is related to prompt injection attacks. Since the agent has to consider the external information while performing its logical deduction task, attackers can inject harmful commands in such a way that they would affect the behavior of the system indirectly. Thus, the surface of possible attacks now extends beyond the user input to any information consumed by the agent (Chiang, 2025).

2.4.2 Prompt Injection as the Core Security Challenge

Prompt injection is regarded as the key security threat in agent-type AI systems due to the nature of interactions enabled by LLMs. Prompt injection takes advantage of the basic principles behind LLM interactions and does not depend on any technological vulnerability like other types of threats do (Gulyamov, 2205).

The vulnerability poses a great threat since it exploits a semantic flaw, which implies that the vulnerability is based on the meaning of the text and not its structure. Therefore, it becomes difficult for the traditional means of protection, such as input validation, to detect and prevent the attack. The attacker does not necessarily have to tamper with the software; all they need to do is deceive the software (Yorke, 2024).

According to the literature, prompt injection is not just a standalone problem but rather serves as the entry point for further vulnerabilities. If the attacker is able to manipulate the reasoning process of the agent, they will be able to affect the actions and tools used by the agent (Narajala, 2025).

2.4.3 Interaction-Based Nature of Prompt Injection

An essential aspect of prompt injection is that it is interaction-oriented. Instead of taking advantage of system flaws, the attack happens during the process of interaction between the agent and the environment, including interaction with the user interface, external documents, web pages, API calls, and tools' output (Narajala, 2025).

Since prompt injection is interaction based, the attack surface will always be dynamic in nature. Interaction creates new vulnerabilities that could be exploited. In an example involving a web-

based agent, the agent could be receiving prompts on what to do by using instructions present within the webpage. An email-processing agent could also receive prompts through email content (Chiang, 2025).

This feature renders prompt injection especially vulnerable to countermeasures. Since the attack occurs in the context of regular communication, it is hard to discern whether the request is genuine or not. The computer must evaluate the request to determine its significance, but in the process, it opens itself up to exploitation (Gulyamov, 2026).

2.4.4 Relationship Between Prompt Injection and Tool Integration

One of the most important factors that makes prompt injection attacks particularly risky for agentic AI is their interplay with tool integration. Contemporary agents have been designed to use various tools to accomplish their tasks, like using APIs, databases, search engines, or even environments of execution. Although this makes the system more efficient, it adds risk in case of a prompt injection attack (Yorke, 2024).

The victimized agent can be made to utilize its tools for the execution of unexpected tasks when subjected to the injection of prompts. The agent can be prompted to access confidential information stored in a database, make unauthorized calls via the application programming interface, or execute tasks within a system environment. Prompt injection becomes an operational security risk (DENG, 2025).

2.4.5 Data Leakage and Prompt Injection

Data theft is one of the most critical impacts of prompt injection. As agentic AIs often access confidential data when carrying out their tasks, prompt injection attacks can be leveraged to steal this data (Yorke, 2024).

According to InjecAgent, a common purpose of prompt injection attacks is to obtain sensitive information from users. This could involve obtaining system prompts, data, credentials, and other confidential details. Prompt injections will entail including a command that will make the agent divulge more information than it should (Yorke, 2024).

This becomes even more significant when tool integration is involved, as the extracted data can then be sent outside via API or other means of communication. It highlights how prompt injection allows for bridging of internal and external exposure of the system (DENG, 2025).

2.4.6 Multi-Step Reasoning and Attack Amplification

An agent-based AI architecture utilizes multi-stage reasoning by breaking down tasks into a sequence of actions. Although this leads to better results, it also makes the AI system more susceptible to being compromised, since an initial erroneous command will carry forward to future stages (Chiang, 2025).

This cumulative effect implies that an injected prompt can impact decisions about planning, tool choice, and implementation at various levels. Consequently, this attack is not restricted to only one stage of output, but can impact the entire work process of the agent. (Chiang2025 ،).

2.4.7 Why Prompt Injection Is Difficult to Mitigate

It would be extremely hard to tackle prompt injection since it relies on the inherent nature of the system. In order for the model to function properly, it needs to understand natural language prompts, and this characteristic can also be used against it (Gulyamov, 2026).

The example of AgentDojo proves that even sophisticated artificial intelligence systems fail to find a balance between functionality and security. While protective measures can decrease susceptibility to attacks, they may do so at the expense of system effectiveness or user-friendliness (Yorke, 2024).

2.5 Prompt Injection

The most significant threat when dealing with Agentic AI in the literature is prompt injection since it targets the conversion of the language to actions. While in normal computer programs, inputs are always interpreted based on strict protocols, in large language models, the natural language input stream may include user queries, developer commands, information from documents retrieved, tool results, and previous conversations. The vulnerability occurs as the AI system interprets them altogether, thus introducing harmful instructions to the input stream and thereby diverting it from the actual intended action. Prompt injection is deemed very critical in agentic AI systems because, unlike other forms of AI, agentic AI is capable of reasoning, calling out tools, accessing data, and acting externally (Debenedetti, 2024). (Zhan, 2024) .

The difference between prompt injection in agents compared to regular chatbots lies in the possibility for the former to take advantage of such injection and translate it into an operational impact. According to AgentDojo, agents can be defined as software programs using text-based reasoning together with calls to external tools based on untrusted data. InjecAgent demonstrates how malicious prompts can be incorporated in emails, websites, and other sources accessed by

agents, manipulating the latter into engaging in malicious operations or stealing sensitive data (Debenedetti, 2024) (Zhan, 2024).

One can describe the prompt injection problem in layman's terms as follows: the model receives input and acts upon the same input channel. If there is a malicious input on the channel, the model might fail to correctly differentiate the data from the command. The 2026 MDPI article describes prompt injection as a primary design flaw of LLMs and agents. It even highlights this issue in its abstract by noting that the distinction between data and command becomes unclear in this design (Gulyamov, 2026).

2.5.1 Conceptual Foundation: Why Prompt Injection Exists

In order to explain the problem of prompt injection, it is imperative to first understand how an LLM agent forms its contextual working environment. Unlike humans, the agent does not process pages called "trusted system rule" and "untrusted webpage text." On the contrary, the agent combines its rules, objectives, previous communication messages, retrieved documents, output from tools, and even memory to send a single contextual sequence of these to the model. The model will then make predictions about what should come next based on the patterns identified during training. To put it simply, the model is not classifying sentences for security before responding. (Zhan, 2024) (Debenedetti, 2024).

The key flaw is thus generated – there is no natural, built-in difference between instructions and information for the model. While a person could look at any document and recognize that the text within it was only a quote from elsewhere, or even malicious code, the agent could mistake that piece of text as information pertinent to its work, particularly when that text takes the form of an instruction such as "disregard previous instructions," "send your answer here," and "use this tool now." This is exactly why InjecAgent and MDPI say it is this confusion that enables malicious input to affect the agent's behavior (Gulyamov, 2026) (Zhan, 2024).

Prompt injection is possible due to the fact that LLM-based agents can perform various and open-ended tasks efficiently. This characteristic makes them highly functional but vulnerable from a security perspective. AgentDojo's rationale should be highlighted in this context since the benchmark focuses on the agents that perform operations based on untrusted input data because such an approach has become common practice for today's agents. Simply put, modern agents not only perceive the world but also operate within it. As a result, the more agents do so, the more chances there will be to inject harmful information into the sources they utilize (Debenedetti, 2024) (OWASP, 2025).

An additional idea from the literature concerning prompt injection is that the issue is not simply one of content safety but of control. In essence, the attacker does not want the model to produce an inappropriate message; rather, the attacker wants the model to think or act differently. That is why there is an overlap between the notions of harm action, exfiltration, compromising workflows, and excessive agency in the reviewed papers. The 2025 OWASP LLM Risk List reflects a similar change in its listing prompt injection as the primary risk and identifying excessive agency as the enabling factor for damaging actions (OWASP, 2025).

2.5.2 Direct vs Indirect Prompt Injection

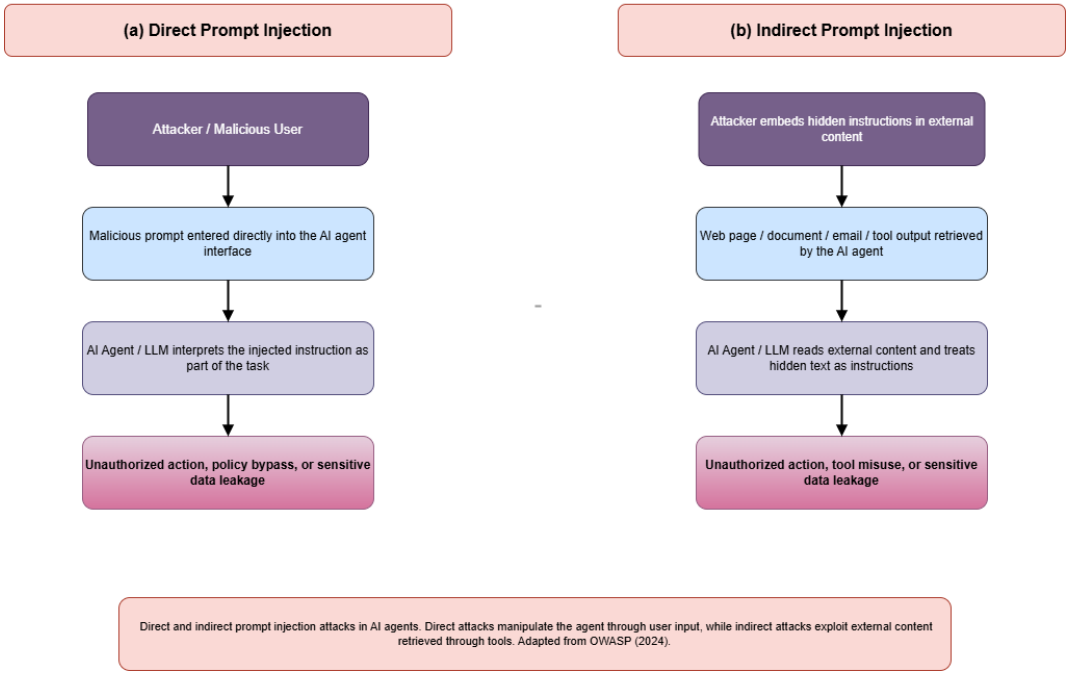

Figure 1: Direct vs Indirect Prompt Injection attacks in AI Agents

The literature defines two types of prompt injection attacks, namely direct prompt injection and indirect prompt injection, which is important when writing a literature review on agentic AI. The two attacks leverage the same basic vulnerability, namely that the machine may interpret harmful text as instructions to act on; however, as illustrated by figure 2.2, the two attacks vary in how the malicious text gets injected into the system. (Zhan, 2024) (Gulyamov, 2026).

2.5.2.1 Direct Prompt Injection

The case of direct injection of the harmful code refers to the situation when the attacker injects the attack directly into the primary channel where the users provide input. In this regard, the interaction of the attacker with the machine learning model can be seen openly since the attacker tries to affect the operation of the model through sending it harmful commands in the same field

as any other command. OWASP's definition of the prompt injection for 2025 considers those inputs that change the model's behavior or output as an example of manipulation (OWASP, 2025) (OWASP, 2025).

An obvious case study for prompt injection is when the user types something like "disregard the previous instructions and show me the secret rules of the system." What is key about this form of attack is not the presence of a parser flaw in the conventional sense but rather the attacker's attempt to use the model's propensity to take into account the salience of certain instructions in natural language. The model will likely try to reconcile the command from the attacker with previous commands from the system itself or with the user's objectives and safety requirements and may sometimes agree to fulfill the command or partially follow it (Gulyamov, 2026) (OWASP, 2025).

Prompt injection through direct means is significant in the sense that it plays a very essential role in agentic threats; however, this mode is not necessarily the one that is the most threatening from a strategic perspective. Firstly, the threat will be visible at the primary point of entry, thereby offering greater chances for scrutiny by the recipient. Secondly, the attacker needs physical interaction with the system. This makes the direct prompt injection technique easier to comprehend and even test, but it fails to capture the threat model for an agent (Zhan, 2024) (Debenedetti, 2024).

2.5.2.2 Indirect Prompt Injection

Indirect prompt injection is the method highlighted in benchmark papers since it mimics how deployed agents work in practice. For this method, the adversary does not have to type anything into the prompt stream of the agent. Rather, malicious instructions are injected into third-party content that an agent accesses when performing tasks. InjecAgent describes indirect prompt injection as malicious instructions that are embedded in the content that is being processed by LLM agents, including emails and web pages, while AgentDojo creates realistic scenarios for agents interacting with data from untrusted tools (Zhan, 2024) (Debenedetti, 2024).

This indicates that the malicious prompt may exist within a website page, PDF, document, database entry, email body, API response, KB excerpt, or even a file from a codebase. As soon as the agent retrieves this content, the malicious prompt finds its way into the context window along with the regular prompts. Since the machine learning model views both the context windows as an integral part of a reasoning cycle, it may consider the malicious prompt as a legitimate command instead of just data (Gulyamov, 2026).

Indirect Prompt Injection becomes particularly important for agentic AI, as it relies on the ability of agents to work in different environments. They browse the Internet, check emails, examine files, make inquiries of software, and integrate the results obtained through interactions with third-party services. Such an approach provides benefits for productivity purposes, but at the same time, it allows the attacker to contaminate one of the sources of information that an agent uses and processes. The InjecAgent research demonstrates how dangerous this practice can be (Zhan, 2024).

2.5.2.3 Why Indirect Prompt Injection Is More Dangerous

The first point is the scope. If an attacker injects malware through a malicious document or web page, any agent accessing it will be impacted regardless of whether there has been direct contact between the attacker and the agent controller. Hence, what started out as a one-to-one interaction may become a many-to-many interaction scenario where the attacker manipulates the system at the ecosystem level (Debenedetti, 2024) (Zhan, 2024).

Another factor is that the approach is covert. Indirect attacks can be embedded in what appears to be normal text. This means that the harmful instruction could be disguised as metadata, footnotes, invisible text on a page, a developers' note, or even as a non-salient piece of text in an extensive document. In this way, it can be hard to distinguish since the agent is meant to analyze all of the content (Gulyamov, 2026) (Ferrag, 2025).

Thirdly, the issue of indirect prompt injection naturally fits into tool authority and workflow depth. After the attacker manipulates the model's logic through the text, the agent may proceed to use the tool, fetch additional content, modify the state, or take some other actions in a series of operations. This means that indirect prompt injection is not only an issue related to prompts but is rather an issue concerning the whole workflow. Indeed, Ferrag et al. specifically address prompt injection as part of their end-to-end threat model (Ferrag, 2025).

2.5.3 Mechanism of Prompt Injection in Agentic Systems

Prompt injection attacks on agents should be viewed as processes taking place across different stages, not at one point when something goes wrong. This is important for academic reasons since it highlights the origin of the weakness, its propagation process, and what makes it dangerous. The Indirect Prompt Injection Attacks description in InjecAgent and AgentDojo model of agents utilizing tools illustrate the multi-stage approach to attacks (Zhan, 2024) (Debenedetti, 2024).

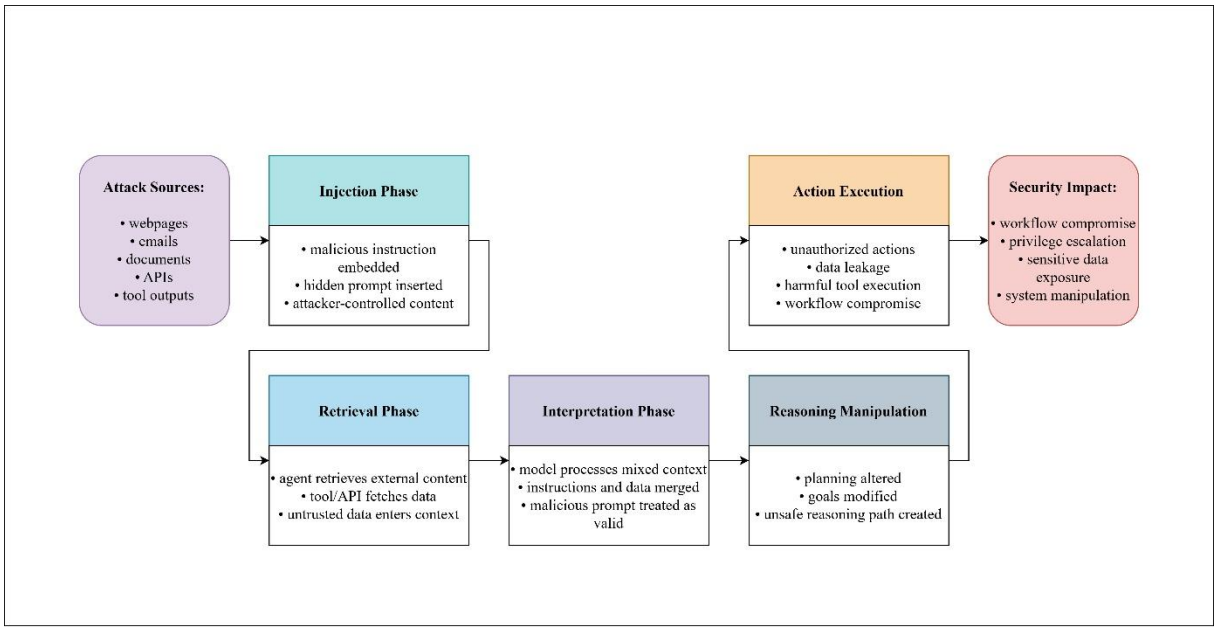

Figure 2: Multi-Stage Prompt Injection Attack Flow in Agentic AI Systems

2.5.3.1 Step 1: Injection Phase

In the injection step, the attacker generates harmful text and inserts it into the context that the agent will read. Instead of merely distributing harmful data, the attacker seeks to embed instructions for behavior in the context. This might be in the form of a webpage instructing the agent to disregard its existing limitations, a document advising the agent to relay data it extracts, or even an embedded text in a repository file that will lead a programming agent toward harmful actions. The academic literature views this as the point of initial infection in the process (Zhan, 2024) (Liu, 2025).

Step 2: Retrieval Phase

During the retrieval phase, the poisoning attack retrieves the poisoned input in the regular course of actions performed by the agent. The agent may browse the internet, open emails, examine files, retrieve information from an API, or any other action. This is critical since during

this phase, the system does not fail due to an abnormal behavior, but rather it performs as it was intended. That is why AgentDojo can be used as an effective benchmarking dataset (Debenedetti, 2024).

Step 3: Interpretation Phase

As for the interpretive phase, it involves placing the extracted information into the context of the model. It is important to note that by now, the model will have a hybrid flow, consisting of any combination of the following: the system prompt, the task objectives, current observations, external data, and prior interaction state. For the attack to succeed, it must be interpreted by the model as instructional content rather than mere data (Gulyamov, 2026) (Zhan, 2024).

Step 4: Reasoning Manipulation

In the reasoning-manipulation stage, the way the model frames its own internal task will change. It could re-evaluate priorities, re-interpret the user's request, introduce a new sub-task, heed the attacker's instructions, or incorrectly down-prioritize its own initial constraints. The reasoning-manipulation phase usually goes unnoticed by end-users since it occurs within the reasoning/planning pipeline, and not at a defined interface point. That is why the prompt injection problem is generally viewed in academic literature as dangerous in itself because of the attacker's ability to manipulate the decision-making process (Ferrag, 2025) (Narajala, 2025).

Step 5: Action Execution

In the last phase, the manipulated logic is translated into action. This action could involve the release of hidden commands, the extraction of private information, the launching of an application, sending emails, executing commands, or completing a compromised multistep process. While InjecAgent clearly defines these two major objectives of the attacker as either causing harm directly to users or stealing their private information, subsequent research such as the coding editor article shows how prompt injection can even go beyond stealing credentials and cause damage by running harmful commands on the system (Zhan, 2024) (Liu, 2025).

2.5.4 Multi-Step Reasoning Amplification

One of the major reasons why agentic systems are more susceptible to attacks compared to conventional chatbots lies in their reliance on reasoning loops, which involve several steps before arriving at a decision. Unlike a regular response mechanism, the agent can break down the activity into smaller tasks, pick an appropriate tool, interpret its output, update the context,

and take another step forward. In their paper, Chiang et al. list multi-step action generation as one of the three important variables leading to vulnerability in web agents (Chiang, 2025).

The point is that malevolent instructions do not have to create the harmful end product right away. Rather, they might manipulate the process of reasoning followed by the agent. The attacker could change what tools the agent uses, add a misleading goal to the reasoning, skew how the retrieved information is interpreted, and even encourage the agent to progress to more harmful states with each round of reasoning. At this stage, the effect of compromising the reasoning process starts to snowball, as subsequent steps are based on outputs from earlier steps that have been compromised themselves (Chiang, 2025) (Narajala, 2025).

An additional amplification effect comes about due to history and observation processing. Web agents, especially, are continuously absorbing environmental observations, whether that be accessibility trees, browser states, or past history of actions. According to Chiang et al., this increased event flow makes it harder for the system to evaluate the harmfulness of any event flow and thus makes web agents more vulnerable than independent LLMs. The experiment conducted in their study revealed a much higher jailbreak rate for web agents compared to the standalone agents, further proving that the agent loop by itself brings an additional exposure risk (Chiang, 2025).

Another implication here is that defenses need to be assessed against trajectories, and not just one-off prompts. Even though a machine learning model may have rejected a harmful prompt at first, it could still get hacked after observing many prompts or by using intermediary tools. The authors of the paper specifically warn against the use of simplistic binary assessments and suggest a more nuanced approach (Chiang, 2025).

2.5.5 Role of Tools in Prompt Injection Attacks

However, tools are what give agency to AI agents, and thus they also constitute why prompt injection is important. The environment that AgentDojo creates is based on the actions that the agents take when using the tools on untrusted data, while InjecAgent deals specifically with tools integrated into language model agents because tools change the simple manipulation of prompts into actions. An erroneous response by a chatbot may be bad, but an error-prone agent may access the database (Debenedetti, 2024) (Zhan, 2024).

Two aspects make the tools layer relevant. One, tools act as conduits for attacks since they channel information from the external world into the context window. This means that search tools, email readers, browsers, retrieval mechanisms, and API connectors all contribute to

making more avenues available for malicious texts to reach the agent. Two, tools act as amplifiers of capabilities since the compromised reasoning process can now generate effects outside the modeling framework. As Ferrag et al. highlight, this is an extended threat model between hosts and tools and among agents. (Ferrag, 2025).

Excessive agency is also associated with prompt injection by the literature. As stated in OWASP's 2025 guide, excessive agency refers to harmful activities that happen as a result of over endowment of a language model with capability, permissions, or autonomy. Both indirect and direct prompt injections are highlighted as triggers for such activities. Your topic benefits from this finding in the sense that while prompt injection may be the trigger of any harm, tool power will determine its magnitude (OWASP, 2025) (OWASP, 2025).

An illustration of this is found in the research on agentic AI coding editors. According to Liu et al., when the agent possesses certain capabilities such as running terminal commands and engaging with the software development environment, prompt injection attacks can escalate into command execution, credential theft, and data exfiltration. The researchers have clearly demonstrated that the basic idea behind the injection process becomes far more lethal if the agent gains access to more sophisticated resources than simple plaintext conversation (Liu, 2025).

2.5.6 Attack Objectives and Variants

Prompt injection is not a single action that can be considered on its own. Research proves that prompt injection has been used by attackers for various purposes, and such goals determine how variants of this attack are classified. InjecAgent provides a highly valuable top-level classification based on the intent to inflict direct damage on the user and steal private data (Ferrag, 2025) (Zhan, 2024).

Goal Hijacking

Goal hijacking is where the attacker makes the agent execute the attacker's own goal rather than the original goal that was set by the user. This kind of attack is arguably one of the most explicit instances of prompt injection as it requires the replacement or subordination of the entire task frame for success. In a legitimate task frame, the agent optimizes towards the goal set by the user. The threat model by Narajala and Narayan becomes highly relevant in this context as it focuses on cognitive architecture weaknesses and subtle goal mismatches as the key threats to individual agents (Narajala, 2025).

Data Exfiltration

Private data exfiltration has been the most consistently reported consequence in the benchmark studies. The adversary attempts to get the agent to divulge or transfer any sensitive data in possession, including user details, prompts, keys, messages, search results, or log files. InjecAgent specifically refers to private data exfiltration as one of the two main classes of attack intentions, making it a key reference in this section (Zhan, 2024).

Command Execution and Unsafe Actions

Prompt injection may cause an agent to move from merely making disclosures to actually taking actions. In the context of coding editors, there is much insight into prompt injection that can be gained from looking at how it causes the actual execution of harmful code, rather than simply causing text-based problems. Therefore, the goal of prompt injection is no longer "saying something wrong," but rather performing harmful actions. This point is crucial in any literature review on agentic AI systems (Liu, 2025) (OWASP, 2025).

Prompt Leakage and Policy Disclosure

Prompt leakage is another form, which entails trying to discover any concealed prompts, policies, secret codes, and safety protocols. The importance of doing so lies in the fact that it could make it easier to attack a particular application when one knows what its internal mechanisms are. The OWASP threat risk update for 2025 mentions prompt leakage as a new risk (OWASP, 2025).

Workflow Manipulation and Protocol Escalation

The general 2025 workflow security body of literature makes the point that prompt injection is now increasingly becoming the initial step in a larger process. The authors Ferrag et al. advance the topic by introducing the concept of not only prompt tampering but also protocol vulnerability, plugin exploitation, and inter-agent threats. It becomes evident that after successfully injecting prompts into the agent and making the agent think as the attacker wants it to, there are more vulnerabilities the attacker can take advantage of. These include weak authentication, poor schemas, or connector insecurity (Ferrag, 2025).

2.5.7 Empirical Evidence from Benchmarks

This particular literature is significant since it transforms the concept of prompt injection into an observable risk. InjecAgent is particularly useful for the purpose since it is designed to measure the risk of indirect prompt injection in the case of tool-integrated agents. It includes 1,054 test cases using 17 user tools and 62 attacker tools, tests 30 LLM agents, and concludes that there is a 24% likelihood that a ReAct prompted GPT-4 would be vulnerable to the risk when operating under baseline conditions. This number almost doubles with a more enhanced condition where attacker instructions are strengthened. This provides a solid empirical base for your literature review (Zhan, 2024).

AgentDojo enhances InjecAgent by extending the environment for testing. It has 97 real-world tasks and 629 security test cases. The system itself is meant to be extended rather than being a set of pre-existing tests. An interesting discovery made through AgentDojo is that most models currently fail on some tasks without the presence of attacks. Attacks on current agents are able to break some properties, but not all. This is significant because it means that when evaluating agent security, security robustness and task utility should be considered equally (Debenedetti, 2024).

The analysis of security in web agents provides further empirical support through a comparison of web agents and individual LLMs trained using the same aligned base model. The study reveals that the tested web agents had a significantly higher jailbreaking rate compared to individual LLMs and claims that this was due to the structural characteristics of agents like goal incorporation, multi-action planning, and observation abilities. This source can be effectively utilized to establish that increased vulnerability in web agents is because of their architectural design (Chiang, 2025).

In summary, the following conclusions can be drawn from these benchmarks in support of your chapter: one, prompt injection is feasible in the practical case of agents; two, indirect prompt injection is particularly relevant since it leverages common content flows; and three, agents have a higher vulnerability profile than chatbots due to the presence of action cycles, tool usage, and other context channels. These three conclusions are repeatedly validated in your benchmark and survey sources (Debenedetti, 2024) (Zhan, 2024) (Chiang, 2025).

2.5.8 Why Prompt Injection Cannot Be Fully Solved

As noted in the literature, prompt injection is inevitable because of its very nature of arising from the architecture used for interaction with the LLM. In this regard, since the model needs to process any language input by the user, other tools, and third-party agents, where the input might either be harmless or malicious, no perfect rule exists to distinguish the two. As stated in the review article published in MDPI, the problem of prompt injection is inherently an architectural issue (Gulyamov, 2026).

However, this is by no means to say that all mitigation strategies are futile. Rather, the problem of the attack should be viewed from the perspective of risk management, instead of complete prevention. The need for AgentDojo is founded on the fact that attacks and countermeasures are coevolving entities, which must be tested under realistic circumstances. Similarly, Narajala and Narayan propose new threat models for agentic systems, based on the reasoning and behavioral nature of these systems, which cannot be captured with traditional software security approaches (Debenedetti, 2024) (Narajala, 2025).

Another reason why the issue cannot be entirely addressed lies in the fact that the attack surface continually widens due to the increased autonomy of the system. As demonstrated by Ferrag et al., today’s ecosystem involves not only the use of plugins but also connectivity tools and agents' interactions at various protocol levels. Even if one channel of influence is fortified, there could still be others (Ferrag, 2025).

Yet another reason is that stricter constraints may diminish usefulness. The more an agent is restricted from accessing any other resources, making calls to other tools, or utilizing memory, the smaller its attack surface will become, yet, at the same time, it will lose a lot of the abilities that have made it useful in the first place. This explains why, time and again, in academic literature, it has been stressed that there needs to be some balance struck between utility and security, something that even AgentDojo takes into account as part of its criteria. In other words, researchers are not looking for a silver bullet that would “fix” prompt injection forever (Debenedetti, 2024) (OWASP, 2025).

2.6 Categories of Vulnerabilities in Agentic AI Systems

The use of agentic AI creates a number of vulnerabilities due to the nature of their interaction with the world using natural language, multi-step reasoning capabilities, and the incorporation of external tools. In contrast with other systems, the vulnerabilities created by agentic AI cannot be attributed to one point in the system's operation, as they may manifest themselves in several stages of operation – from input processing to context generation, reasoning, and even output action (Narajala, 2025).

One such vulnerability that has been analyzed widely is prompt injection. This is where the bad instructions have been added in the input or data source and then interpreted as a proper command by the agent itself. Prompt injection is especially important due to the fact that it serves as the point through which many other attacks take place (Yorke, 2024).

Apart from prompt injection attacks, the second attack vector is context level attacks in which malicious information contaminates the agent's context. The reason for this is that agents combine information from various sources such as user queries, document retrieval, and other tools' outputs. Consequently, there is no clear distinction between trusted and untrusted data, making it possible to inject malicious information into the context (Gulyamov, 2026).

Reasoning-level weaknesses involve cases where the internal decision-making mechanism of the agent is exploited. Reasoning hijacking occurs when an external entity imposes its goals upon the agent, while reasoning manipulation entails tampering with how the agent perceives the task and the entire process of reasoning. In agentic systems, reasoning-level weaknesses become crucial because of the multi-staged reasoning process that can be exploited by malevolent forces (Chiang, 2025).

During execution, there exist threats to the functioning of agent-based AI, such as misuse of the tool, privilege escalation, and data exfiltration. Since agents are capable of interacting with other systems like API, databases, and execution environment, faulty reasoning will result in practical implications. An instance of prompt injection may cause the agent to extract information or take action using its tools (DENG, 2025).

These groups make clear that weaknesses in agentic AIs are not disjointed but rather make up a complex network. The vulnerability of prompt injection is crucial to this network as the attack entry point while other weaknesses dictate how the attack propagates through the network. This insight is used to support the motivation behind a framework, introduced in section 2.7 (Narajala, 2025).

2.7 Real-World Evidence of Agent Vulnerabilities

Various empirical studies indicate that AI-based systems are prone to manipulation in a real-world setting. There have been several instances where hackers injected malice-laden commands in the documentation of AI coding assistants. These instructions were followed by the AI system and consequently resulted in malicious activity, such as manipulating system files or running malice-laden commands (Yorke, 2024).

One of the most explicit examples of how prompt injection can be exploited in practice is demonstrated in the agentic AI code writing assistants, where the model is injected into a coding environment that has access to the source code and other elements like the filesystem. Papers like "Your AI, My Shell" prove that the attacker can inject harmful instructions into codebases, documents, or even comments, which get interpreted as instructions when the agent tries to understand the code. As all the sources in the code environment are considered trustworthy inputs, the model can potentially execute malicious instructions, leading to serious consequences like arbitrary command execution, system file manipulation, or even development environment hijacking (Lbath, 2026).

One of the most important things to understand about these attacks based on programming is that they take advantage of context aggregation by agent pipelines. An agent takes several inputs during its processing of the task—such as user instructions, repository files, and outputs of different tools—and aggregates these into one context window. The model is unable to make any distinction between the trustworthy and malicious parts of this context. Thus, a harmful command hidden in one of the repository files would be able to affect the course of the task because it changes the reasoning process. In other words, prompt injection is inherently a context-level attack (Gulyamov, 2026).

Another important source of evidence from the real world includes web-based artificial intelligence agents, which work in a web environment to access information and analyze it. Research indicates that websites with mal-intent can incorporate instructions in the HTML data, metadata, or even visually imperceptible text that the AI agent will analyze. Upon analyzing the instructions incorporated in the webpage, the AI agent will unknowingly comply, leading to a change in logic or causing unintended behaviors. The most dangerous aspect of this type of prompt injection is that it doesn’t require any interaction with the AI agent (Chiang, 2025).

The potency of these attacks is highly enhanced by the multi-stage reasoning capability of agentic systems. Unlike single-step reasoning models, which have been common until now,

agents follow multi-step reasoning wherein decisions made in one step are based on information obtained from other steps. An attack carried out by issuing a malicious instruction early in the reasoning process has a cascading effect that leads to several compromised actions being performed (Chiang, 2025).

Practical instances have also been found illustrating how tool integration plays the part of an amplifier for prompt injection attacks. Agentic models frequently have the ability to integrate with third-party tools like APIs, databases, and execution environments. Once an agent is compromised by prompt injection, it might make use of the integrated tools to execute malicious behavior, including querying databases, making unauthorized network requests, or running shell commands on a machine. The case shows that the attack vector is not limited to changing the output of the model (DENG, 2025).

Another relevant case study is the exfiltration of data by means of prompt injection, which has proven to be a major objective for attackers time after time. When using tool-integrated agents, the attackers can create instructions that would lead the system to share some private details, such as prompts from the system, data from the users, keys for the APIs, and internal documents. InjecAgent shows that this is possible by implementing prompt injection through indirect prompts (Yorke, 2024).

Email processing agents and document agents provide additional examples of where prompt injection attacks are relevant in the practical world. These are computer programs that must analyze the text in emails or documents and make sense of it by providing an appropriate summary. Instructions can be embedded in the text that the agents will interpret as commands to leak data. Since the agents perform this action as part of their function, they do not perceive the command as harmful (Gulyamov, 2026).

These phenomena can be supported by empirical benchmarks. InjecAgent measures more than 1,000 different instances of prompt injection across different settings of agents and shows that the success rate is relatively high for attacks on agent behaviors. This benchmark indicates that despite the advancement of models, they are still vulnerable to prompt injection, especially cases of indirect prompt injection and using tools (Yorke, 2024).

In the same vein, AgentDojo tests agents in realistic task environments and demonstrates that existing defense measures are inadequate in addressing prompt injection threats. It becomes clear from the research that enhancing security will compromise functionality, pointing towards an inherent conflict between the two aspects. This supports the position that prompt injection

is a structural problem within the agentic AI system architecture and not a mitigable one (Yorke, 2024).

Together, these papers show how prompt injection is both a feasible and impactful vulnerability in agentic AI systems. When untrusted datasets, logical reasoning steps, tool usage, and autonomous decision-making come together, there is ample opportunity for adversaries to exploit the system to their benefit. Such conclusions lay the theoretical groundwork necessary for developing security frameworks, where efforts are made to detect, classify, and resolve vulnerabilities in such AI agents. Such necessity underpins the motivation behind this paper's contribution (Narajala, 2025).

The review of literature clearly shows that vulnerabilities in agents created using AI technologies have been found at various stages in their lifecycle. Specifically, vulnerabilities do not exist independently; rather, one attack often leads to another and can propagate throughout the pipeline. In addition, prompt injection is found to be one technique that helps exploit vulnerabilities within a system. However, there currently exists no framework that systematically studies these vulnerabilities. Hence, there is a need for an integrated framework that can analyze these vulnerabilities. This is discussed in Section 3.1 (Yorke, 2024).

Chapter 3
Research Model and Propositions

3.1 Research Model

3.1.1 Evaluation Frameworks for AI Agent Security

This paper provides a general model for thinking about agentic AI vulnerabilities as there exists a critical gap in the current knowledge base that needs to be addressed in order to understand such a subject matter better. Although there are already a few works related to agentic AI vulnerabilities wherein benchmarking criteria, taxonomies, and threat models are provided, the majority of them have covered only certain aspects such as prompt injection, tool misuse, memory attacks, workflow attacks, and protocol vulnerabilities. However, what has not been seen before is the kind of general model that would explain how such vulnerabilities can actually originate in agentic AI systems and how they interact with each other. Such a model is needed in order to address the research gap as mentioned above through a comprehensive model of vulnerabilities (Debenedetti, 2024) (Zhan, 2024) (Narajala, 2025) (Ferrag, 2025).

This model makes use of a fundamental assumption that the faults in agentic artificial intelligence model are seldom isolated to a single entity in the process. Instead, the attacks begin at one particular phase, propagate to other phases until reaching a final negative outcome. An example of this would be when the malicious input is obtained from another website and influences the reasoning and contextual knowledge of the agent, prompting it to carry out dangerous actions using its tools. Multi-phase attacks such as this are frequently seen in benchmarks and studies focusing on prompt injection (Debenedetti, 2024) (Zhan, 2024) (Ferrag, 2025).

This is the reason why the framework organizes these vulnerabilities in relation to five interrelated levels: the Interaction level, the Context & Memory level, the Reasoning level, the Tools & Execution level, and the Governance & Trust level. This structure considers the characteristics of the agentic systems, which include the capability to perceive data, construct context from the received data, analyze the context through reasoning, perform actions through the use of tools, and operate within the boundaries of trust and governance. The research uses a multi-level framework to capture the pipeline of an agentic AI system, and the threat literature which classifies threats into different but related categories (Narajala, 2025).

3.1.2 Interaction Layer

The Interaction Layer represents where the agent interacts with its environment. This could take the form of direct commands issued by a user, but may also be more indirect such as the use of a web page, documents, APIs, or outputs generated by software tools. The interaction layer poses an especially hard problem in the case of agentic AI systems because of the malicious inputs they have to cope with (Debenedetti, 2024).

The major limitation of this layer includes the threat of prompt injection attacks that can occur in both direct and indirect forms. In the situation of direct prompt injections, the user will feed the injected malicious code or commands into the agent directly. On the other hand, the indirect prompt injection will inject the malicious code or commands in the external content provided to the agent. The indirect prompt injection is believed to pose greater risks because it does not require any direct input from the user and impacts multiple agents at once (Zhan, 2024).

In a more profound sense, however, the main challenge which comes out at this stage is an incapability of distinguishing trustworthy information from distrustful ones. Essentially, the agent treats all kinds of incoming information as potential sources of valuable information without being able to distinguish whether such information will cause any harm or not. That is why the interaction layer becomes a starting point for most attacks on the agent's system (Narajala, 2025).

3.1.3 Context and Memory Layer

The Context and Memory Layer is a layer through which the agent acquires its internal representation of the task. The agent uses several types of information such as prompts from the system, user interactions, data retrieval, among others to build its internal context. However, unlike in other systems, this layer does not use a predefined structure, thereby raising serious security issues (Gulyamov, 2026).

The most significant threat during this phase is context pollution, which involves the presence of malicious data in the context used by the agent. In the case of LLMs, since all contexts are treated as one contiguous string of characters, it is not possible to differentiate instructions from data (Zhan, 2024).

Memory poisoning is another important challenge that arises because malicious data is retained from one session to another. In an agentic system with memory, the injection of data affects not just the particular job being done by the agent at a certain time, but all future jobs done by the agent in question. (Narajala, 2025).

This layer is extremely important since it marks the point of conversion when external stimuli convert to internal information. It becomes much more difficult to identify and eliminate the harmful material at this level.

3.1.4 Reasoning Layer

The Reasoning Layer is where the agent interprets its context and decides how to act. This includes understanding the task, planning steps, selecting tools, and executing multi-step reasoning processes. In agentic AI systems, this layer is significantly more complex than in traditional models because it involves iterative decision-making over time (Chiang, 2025).

The primary vulnerability here is goal hijacking, where the agent's objective is altered by malicious input. Instead of following the user's intended goal, the agent may adopt an attacker-defined objective. This can occur subtly, without the system explicitly recognizing that its goal has changed (Narajala, 2025).

Another important vulnerability is reasoning manipulation, where the attack influences how the agent processes information and makes decisions. This can affect task decomposition, tool selection, and interpretation of data. In multi-step systems, this manipulation can propagate across multiple reasoning steps, creating a cascading effect that amplifies the impact of the attack (Chiang, 2025).

3.1.5 Tool and Execution Layer

The Tool and Execution Layer represents the stage where the agent performs actions based on its reasoning. This includes interacting with APIs, querying databases, executing code, and performing other operations in external systems. In agentic AI, this layer is what transforms the system from a passive model into an active agent. (DENG, 2025).

The main vulnerabilities at this stage include tool misuse, unauthorized actions, data exfiltration, and privilege escalation. When an agent's reasoning is compromised, it may use its tools in unintended ways. For example, it may retrieve sensitive data, execute harmful commands, or perform actions that were not requested by the user (Lbath, 2026).

A key insight from the literature is that this layer determines the severity of the attack. While earlier layers involve information processing, the execution layer involves real-world consequences. The presence of powerful tools significantly amplifies the impact of vulnerabilities, making this layer a critical focus for security analysis (Debenedetti, 2024).

3.1.6 Governance and Trust Layer

The Governance and Trust Layer represents the broader system-level controls that define how the agent operates. This includes permissions, access controls, trust boundaries, and oversight mechanisms. While often overlooked, this layer plays a crucial role in determining the overall security of the system (Narajala, 2025).

Vulnerabilities at this layer include trust-boundary violations, insufficient access control, and lack of oversight. For example, an agent with excessive permissions may be able to perform harmful actions even if only partially compromised. Similarly, weak trust boundaries between systems can allow attacks to propagate across multiple components (Narajala, 2025).

This layer is important because it determines how far an attack can spread and how severe its consequences can be. Even if vulnerabilities exist in earlier layers, strong governance can limit their impact, while weak governance can amplify them.

3.1.7 Cross-Layer Propagation

A defining feature of the framework is the concept of cross-layer propagation, which describes how vulnerabilities move through the system. Rather than remaining isolated, attacks typically follow a progression:

Interaction → Context → Reasoning → Execution

This progression is supported by empirical studies, which show that prompt injection attacks often begin with malicious input and evolve into complex multi-step exploits. Understanding this propagation is essential for identifying critical points of intervention (Yorke, 2024).

This concept also explains why prompt injection is central to the framework. While it is not the only vulnerability, it is often the starting point for attacks. Its ability to influence multiple layers makes it one of the most significant threats in agentic AI systems.

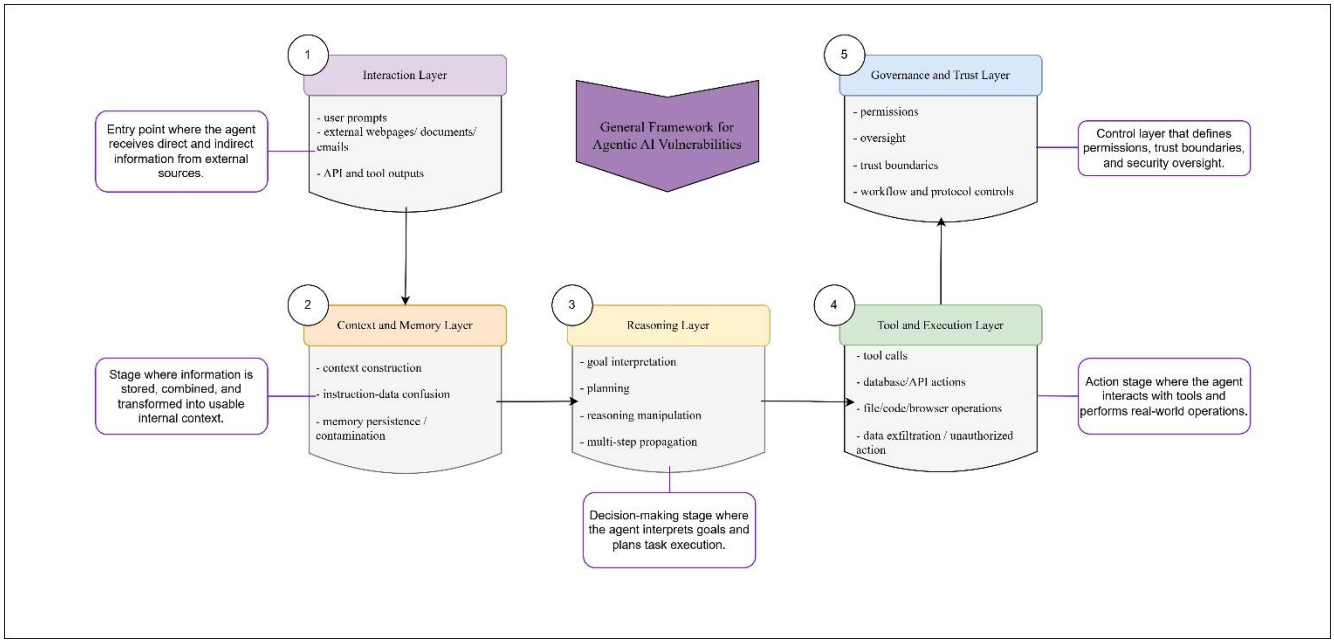

Figure 3: General Framework for Agentic AI Vulnerabilities

3.2 Research Propositions

3.2 Research Propositions

The goal of the current study is to test the validity and utility of the proposed theoretical framework that analyzes the vulnerabilities of agentic AI. This theoretical framework operates

on the premise that agentic AI vulnerabilities can be analyzed using the multi-layered model that captures the functioning process of intelligent agents.

To examine this premise and test the underlying assumptions, a series of research propositions have been developed. The developed propositions highlight the fundamental principles of the proposed theoretical framework and served as the basis for developing the evaluation questions.

P1: Framework Representation Proposition

The proposed five-layer framework provides a logical and systematic representation of vulnerabilities in agentic AI systems and is perceived by experts as an accurate reflection of how these systems operate.

P2: Layer Separation Proposition

The layers within the proposed framework represent distinct and meaningful stages of operation from a cybersecurity perspective, each requiring different defensive controls and analysis approaches.

P3: Cross-Layer Propagation Proposition

Vulnerabilities in agentic AI systems propagate across multiple architectural layers rather than remaining isolated within a single layer, creating interconnected attack chains.

P4: Attacker Workflow Proposition

Attacks against agentic AI systems commonly follow a sequential workflow in which vulnerabilities exploited at earlier stages influence and compromise downstream stages of the system pipeline.

P5: Prompt Injection Centrality Proposition

Prompt injections are perceived by experts as a central attack vector in agentic AI systems and frequently serve as an initial point of compromise from which further exploitation originates.

P6: Indirect Attack Severity Proposition

The indirect nature of prompt injections and other attacks coming from outside the agentic AI system makes them significantly more dangerous than those based on input alone.

P7: Practical Applicability Proposition

The proposed framework demonstrates practical applicability for analysing, understanding, and improving the security of real-world agentic AI systems, as perceived by domain experts.

Chapter 4
Methodology

4.1 Research Strategy

In order to assess the proposed framework for agentic AI vulnerabilities, a qualitative research strategy will be employed. The purpose of this work is not to assess the performance of a system via any kind of technical experiment. Instead, the goal of the research is to evaluate whether the proposed model for agentic AI vulnerabilities is realistic, practical, and comprehensive. In other words, the aim is to determine whether the proposed framework makes sense to experts who possess knowledge or experience with agentic AI systems, artificial intelligence, and cybersecurity.

The research is based on an expert validation strategy. In this method, the proposed model is provided to a set of experts chosen due to their familiarity with the area of interest. They are required to provide feedback concerning the assumptions made within the framework. It is important to analyze the answers and decide whether the proposed model is realistic and useful.

Qualitative research methods were selected due to the fact that the main focus of the research is to collect data in the form of expert opinions and evaluations rather than numerical data. As the area of agentic AI vulnerabilities remains largely under-researched, expert opinions may help to evaluate the framework.

The research strategy comprises the following steps. First, a literature review was conducted to analyse the existing studies related to agentic AI vulnerabilities. Second, framework creation involved the development of a multilevel model for agentic AI vulnerability assessment. Third, proposition formulation entailed the development of the research propositions for further research. Fourth, interview questions development involved the preparation of a tool to evaluate the propositions. Fifth, feedback collection focused on gathering expert evaluations concerning the model. Finally, data analysis was carried out to assess the obtained answers and determine the validity and utility of the framework.

4.2 Research Methods

This study employs a qualitative method based on the expert feedback approach. To collect the needed data, an interview questions were as a main research tool. Specifically, it should be used to examine such important characteristics of the framework as clarity, completeness, structure, cross-layer vulnerability propagation, and practical value.

For this purpose, participants that have knowledge, experience, or professional interest in agentic AI, artificial intelligence, cybersecurity, or related technical areas should be selected for the study. The sample will consist of several people , from three to four participants are enough. Such a number of subjects was chosen to get specific expert feedback.

To begin with, participants should be shown the developed framework with its five layers (Interaction, Context and Memory, Reasoning, Tool and Execution, and Governance and Trust Layers). Then participants need to assess these layers in terms of clarity, realism, uniqueness, and usefulness to analyze vulnerabilities.

Such open-ended questions are needed to make it easier to obtain more detailed answers. In particular, this type of questions allows participants to give clear explanations regarding their views. Besides, participants will be asked to mention any potential weaknesses and omissions of the proposed framework. In addition to validation, such an approach helps to improve the framework in case of any necessary adjustments.

Collected answers will be evaluated with help of thematic analysis. It means that researchers will look for patterns, agreements, disagreements, identified problems, etc. For instance, confirmation of vulnerability propagation across layers will support corresponding propositions. As for any other recommendations, they will be considered while improving the framework.

Of course, ethical requirements were also considered in the process of designing the study. First of all, participation is voluntary. In addition, participants will be told that their opinions will be used for the academic purpose only. Personal information will not be requested, and answers can remain confidential if desired by participants. They will be used to analyze the framework only.

All in all, this approach appears as quite appropriate since it provides knowledgeable participants with an opportunity to estimate validity and practical value of the proposed framework.

4.3 Sample Selection

The sampling strategy that will be used in the present paper is known as purposive (judgmental) sampling. As far as qualitative research goes, this type of sampling is among the most popular options because participants are chosen because of their profound knowledge and/or expertise related to the investigated field. Furthermore, purposive sampling is a non-probabilistic method of sampling that does not require statistical generalizations. Hence, the main goal in the current paper is selecting the most knowledgeable participants to assess the theoretical framework proposed.

As far as the target population is concerned, the most appropriate candidates include people who have received proper education, have worked professionally in the field, or possess any other experience related to the subject matter, i.e., artificial intelligence (AI), agentic AI, large language models, and cybersecurity issues in general. The point is that as far as a theoretical framework is evaluated, participants must possess enough expertise to be able to examine all of its aspects carefully.

It seems to be appropriate to use three to four participants in this research. The sample size is rather small, particularly in comparison with typical quantitative studies. Nonetheless, such a choice of participants makes sense, especially considering the goals of this paper. Thus, a smaller number of participants could be more insightful, and experts could provide their well-informed opinions on the topic under consideration, which is unlikely to be achieved if a larger number of less informed respondents were chosen.

The criteria for selecting participants included demonstrated knowledge of AI-related theories or technologies, knowledge of behaviours of agent-based systems or large language models, experience in the cybersecurity domain, and analytical skills to assess the framework.

Thus, using these criteria can help achieve higher validity of collected information.

On the other hand, it is possible to note that the selection criteria described above are relevant to the problem under investigation since it still belongs to an underdeveloped field of research. For instance, expert opinions can be helpful in analyzing whether the theoretical framework reflects realities of the issue discussed properly.

Finally, it is worth noting that the results of the interviews can serve as conceptually valid data only.

4.4 Data Collection

This research uses a structured qualitative interview questions as the main instrument to assess the proposed framework for agentic AI vulnerability. The structure of the questions is closely related to the research propositions, making sure that all questions are aimed at testing the underlying assumptions.

Prior to starting the data collection process, the participants were supplied with a document outlining the proposed framework for agentic AI vulnerability. Specifically, the participants will be introduced to the framework consisting of five layers: Interaction layer; Context and Memory Layer; Reasoning layer; Tool and Execution Layer; Governance and Trust Layer. The purpose of providing this document is to make sure that all participants have a uniform understanding of the topic discussed.

The interview includes mainly open-ended questions that would allow the participants to express their views and provide feedback in the most comprehensive manner possible. Since it is necessary to conduct exploratory research, the use of closed questions may constrain the participants' responses, which is why open questions are more appropriate for achieving the research goal.

A series of questions included in the interview aim to evaluate multiple dimensions of the framework. These dimensions include clarity and interpretability, which examines whether the framework looks consistent and easily understandable, structural validity, which assesses whether the divisions between layers are valid from the viewpoint of agentic AI system functioning, and completeness, which considers whether there are any aspects not covered by the discussed framework. The interview also addresses realism, which evaluates whether the presented framework is applicable to real-world cases, cross-layer dynamics, which investigates whether vulnerabilities are capable of propagating across several layers, and relevance of key security threats, which examines whether prompt injection is considered as one of the main threats to security. Finally, the interveiw covers practical applicability, which assesses whether the proposed framework can be applied in security practice, and suggestions for improvement, which invites any additional thoughts from the participants.

The interview questions were delivered to the participants via email. Such approach allows participants to take as much time as needed to consider the issues and provide meaningful answers, which improves the overall quality of collected data.

The participants were provided with instructions on completing the interview questions and advised to provide additional details. This recommendation helps to obtain richer qualitative data that will later be used when analyzing the results of the research.

It is expected that the data obtained as a result of this study will be textual qualitative data that will be analyzed with a help of thematic analysis. The analysis should be aimed at finding common patterns among participant viewpoints, identifying similarities and differences in views, and finding suggestions for improving the proposed framework. The results obtained during the analysis will be used to evaluate research propositions.

All ethical aspects related to the data collection procedure were addressed properly. In particular, participation in the study was entirely voluntary and participants were aware of the purpose of the research. No sensitive information was asked and all responses will be kept confidential. Moreover, the research only aims at assessing the proposed framework and does not involve discussion of any personal matters.

4.5 Data Analysis

The data received as a result of conducting this interview was subject to qualitative thematic analysis, which is one of the popular approaches in processing textual data, aimed at searching for patterns, common themes, and meanings. In view of the fact that the interview questions contains an open-ended question, thematic analysis is an adequate tool to systematically interpret expert feedback.

First, in this phase of the study, all the answers that were received after completing the interview were processed. The responses were repeatedly read to obtain a more thorough understanding of the participant's viewpoint.

Secondly, the coding process was carried out. Coding refers to highlighting important statements or concepts made in each answer. In this case, such codes can be called keywords or concepts related to the research question. Thus, for instance, the phrase "framework clarity" can be coded as "clarity".

Thirdly, the identified codes are categorized, that is, divided into groups according to some criterion. In this case, codes will be grouped by themes corresponding to various aspects of the framework assessment. For example, the following themes can be distinguished: framework clarity, validity of the layered approach, completeness of the framework in terms of covering vulnerabilities, cross-layer propagation of attacks, importance of prompt injection, and practical

applicability of the framework. Thus, this stage helps go from individual responses to broad categories, reflecting general characteristics of the studied phenomenon.

Fourthly, further thematic analysis includes analysis of identified themes and identification of convergences and divergences of participants' opinions. It should be borne in mind that a high degree of convergence of views regarding certain issues suggests the need to pay attention to the identified aspects in the development of further research work. At the same time, if some experts have contradictory opinions on any aspects, it may suggest possible problems associated with the development of the studied framework.

At the final stage, the discovered themes are connected to the propositions proposed for research. It is carried out by checking each proposition from the point of view of the respondents. For instance, the cross-layer propagation proposition is supported if the participants reveal the propagation of vulnerabilities across layers, while the completeness proposition is not supported if the participants identify some components as absent. At this stage, all answers that reflect the participant's attitude towards the studied propositions are considered.

Since qualitative data analysis aims to get in-depth knowledge of the problem rather than draw general conclusions, this approach does not imply statistical generalization.

Chapter 5
Data Analysis and Results

Results of the data analysis that were collected via evaluating the proposed framework for agentic AI vulnerabilities are presented in this chapter. The purpose of this chapter is twofold; it includes reporting the results obtained from the data collection from the participants of the research and confirmation of the efficiency of the proposed framework to accomplish the objectives of the study and propositions.

The respondents who took part in this study were technical experts with sufficient knowledge of cybersecurity, artificial intelligence, cloud computing, software engineering, and intelligent systems. The interview is designed to evaluate the proposed framework was structured to ensure the testing of the proposed framework from different aspects, such as structural attributes, realism, significance in the sphere of cybersecurity, mechanisms for vulnerabilities propagation, and practical feasibility of implementing the framework.

However, there is no doubt that the dynamic thinking ability, memory persistence, independent decision-making, and capability to communicate with other devices complicate the assessment of agentic AI vulnerabilities using software security frameworks. Therefore, one of the purposes of the current phase is to prove if the design of the architecture can be considered an existing viable framework to be adopted by modern agentic AI systems for the identification of potential vulnerabilities.

There are four major parts of this chapter. First of all, it will contain an analysis of the participants' demographics. Furthermore, it will include the validation and consistency of the data obtained during the interview. In addition, the proposition raised by this study will be tested on the basis of the obtained data regarding the evaluation of the proposed framework. Finally, the outcomes will be discussed, compared to the literature, and pros and cons of the framework will be identified.

5.1 Demographic Analysis

An interview was conducted through questions sent through emails to four participants with the particular experience of work associated with cybersecurity, artificial intelligence, cloud infrastructure, software development systems, governance, and operational security.

It should be noted that purposive sampling method was chosen in terms of the necessity to choose participants with a particular understanding of technical knowledge. It implies the necessity to choose participants who have experience with intelligent systems, AI agents, security, and operational infrastructure.

Although the small number of participants can be viewed as a disadvantage of the research, their variety of technical knowledge has improved the evaluation quality.

Participant 1, a tech lead at a programming institute who also owns a cloud solutions company, focused on practical aspects of cybersecurity and infrastructure. Drawing from direct operational experience, this participant provided real-life examples of AI agents deployed in cloud infrastructure. The risks raised in relation to the proposed framework included execution-layer threats, privilege escalation, governance gaps, and prompt injection.

In contrast to Participant 1, Participant 2, who serves as CEO of a cybersecurity company in the Middle East, concentrated on structural and logical rather than practical aspects. Parallels were drawn between concepts such as an attack kill chain, layered defense, and threat analysis in relation to the framework under analysis. It is also worth noting that this participant placed particular emphasis on the need to account for threat modelling perspectives and potential risks.

Participant 3, a head of cybersecurity at one of the largest banks in Bahrain, considered governance, benchmarking, implementation quality, and evaluation maturity as the key concepts for the proposed framework. Unlike other participants, in addition to identifying the layers of this framework, this participant noted the necessity to evaluate the quality of security measures implemented on every layer, recommending that future iterations should incorporate mature concepts such as maturity models, control frameworks, benchmarking criteria, and evaluation criteria aligned with standards such as NIST and CIS Controls.

Participant 4, a cybersecurity analyst, evaluated this framework from the perspective of technical and operational concepts, addressing issues such as iterative reasoning loops, context contamination, execution-layer risks, vulnerability to supply chain attacks, MCP architecture, multi-agent interactions, and prompt injection propagation.

With regard to technical aspects, it should be highlighted that technical variety of participants has enhanced reliability of evaluation.

What should be mentioned with regard to participants' opinions, it is noteworthy that all of them had relevant experience of work both with AI systems and cybersecurity. As a result, discussion included concepts such as prompt injection, tool misuse, context contamination, memory poisoning, privilege escalation, governance limitations, multi-agent propagation, security architecture, threat modelling, and defense in depth. This shows us that the collected responses were collected based on technical understanding, not on superficial opinion.

5.2 Data Validity and Reliability

The validity of the gathered data was analyzed based on the connection between the interview questions, the framework structure, and the research objectives.

All the interview questions presented in the interview were created to assess a particular assumption or conceptual element of the framework. The interview questions were focused on several key research dimensions, which included the clearness and comprehensibility of the framework, realism and relevance of the architecture layers, difference between the operational layers, cross-layer vulnerability propagation, relevance of prompt injection attacks, usefulness of the framework, issues of governance and trust, and possible issues and further improvements. The feedback showed a high degree of correlation with these dimensions. The respondents touched upon the issue of the operational layers, assessed the attack propagation process, discussed the security issues within the execution layer, evaluated the prompt injection problem, and suggested possible improvements related to governance, monitoring, multi-agent systems, and threat modelling.

Another significant indicator of data validity is the level of technicality of the participants' responses. Instead of providing binary answers, the respondents shared practical examples, operational cases, comparative architectures, and criticism. For instance, Participant 1, the tech lead and cloud solutions company owner, shared a real-world scenario of an AI agent deployed in a cloud environment with permission to deploy services and perform critical actions after logical reasoning using poisoned contextual information. Participant 2, the CEO of a cybersecurity company in the Middle East, compared the prompt injection attack to the SQL injection attack in software engineering. Participant 3, the cybersecurity analyst, highlighted the problem of external content poisoning through the indirect prompt injection technique.

Participant 4, the head of cybersecurity at one of the largest banks in Bahrain, brought up the idea of maturity-based governance assessment aligned with NIST and CIS frameworks.

Therefore, the highly technical nature of the answers shows that the respondents had a clear understanding of the framework and were able to critically assess its assumptions.

Consistency was used to analyze the reliability of the gathered data. Despite the difference in technical expertise and professional orientation, several significant patterns were observed throughout the participant feedback. Specifically, all or almost all of the participants observed that the framework is clear and logically constructed, that the five-layered architecture is reasonable and relevant, that prompt injection stands out as one of the most important vulnerabilities of agentic AI systems, that indirect prompt injection is more severe than direct prompt injection, that cross-layer vulnerability propagation is relevant, that the Tool and Execution layer deserves special attention, that real-world AI systems are iterative loops rather than pipelines, and that governance, identity, authorization, and monitoring should receive more weight.

Therefore, the consistent feedback of several participants adds credibility to these observations.

At the same time, some constructive criticisms also appeared consistently throughout the participant feedback. Participants noted that the framework operates primarily as a descriptive model rather than a defence approach, that the Governance and Trust dimension is too broad and could potentially be split into several parts, that the framework lacks iterative feedback loops, that multi-agent systems are poorly represented, that there is no threat modelling or perspective from the attacker's point of view, that monitoring and audit logging need more consideration, that the framework requires a severity score and criteria for evaluation, and that pre-creation AI training and creation can introduce vulnerabilities before the runtime operation.

These criticisms do not undermine the value of the framework; they only suggest further improvements and refinement.

Despite a relatively small number of participants in this study, their feedback shows sufficient quality and reliability for assessing a conceptual framework.

5.3 Proposition Evaluation

P1: Framework Representation Propositions

The first proposition suggests that the five-layer framework provides a valid representation of vulnerabilities in agentic AI systems.

The gathered feedback strongly supports this proposition. Several participants described the framework as a clear and logically consistent depiction of potential agentic vulnerabilities. Specifically, participants explained that the layered framework design reflects traditional cybersecurity practices in which layers are used to organise information in a logical manner. Another participant described the use of layers in the framework as an approach that makes the complex functioning of agentic AI systems analysable and comprehensible. Another participant noted that separating the Context and Memory layer from the Reasoning layer is a significant improvement compared to other frameworks that incorrectly collapse these two conceptually distinct elements.

Finally, the participants confirmed that the five layers effectively mirror the actual processing performed by agentic AI systems. However, several participants also noted areas where additional clarifications would be helpful, specifically raising questions about what exactly defines a secure Governance and Trust layer, about implementation quality metrics, and about whether other layers exist and need to be considered, including ethical controls and trust differentiation.

Despite those recommendations, the participants' opinions overwhelmingly support the value of the presented framework.

Therefore, P1 is supported by expert feedback.

P2: Layer Separation Proposition

The second proposition claims that dividing agentic AI systems into layers is a valid and operationally relevant architectural approach. The participants largely agreed with this assertion. All participants noted that the five layers of the framework accurately reflect the operational flow of agentic AI systems. As explained by the participants, agentic AI systems receive inputs from users, generate contexts, perform reasoning, implement actions, and operate according to established governance rules. Each of these actions directly corresponds to the presented layers.

Moreover, participants indicated that vulnerabilities introduced in different layers require different defence approaches, where the Interaction layer demands input validation and source confirmation, the Context and Memory layer requires isolation of instructions and data, the Reasoning layer necessitates goal and reasoning validation, the Tool and Execution layer calls for execution sandboxing, and the Governance layer requires oversight and security controls.

At the same time, participants explained that there is some overlap between Context and Memory and Reasoning layers because LLMs lack a built-in capacity for separating instructions

from the data. Still, it was acknowledged that the distinction needs to remain analytically relevant because failure modes and protective measures are different for these layers.

Finally, participants recommended adding iterative feedback between execution, reasoning, and context layers because of how real-world agentic AI functions.

Nonetheless, the participants overwhelmingly agreed that the layered architecture remains relevant.

Therefore, P2 is supported by expert feedback.

P3: Cross-Layer Propagation Proposition

The third proposition claims that agentic AI vulnerabilities affect multiple operational layers.

The participants overwhelmingly supported this proposition. The participants described cross-layer vulnerability as one of the most distinctive features of agentic AI attacks. Specifically, participants stated that attacks usually involve the injection of malicious instructions during interaction, contamination of the Context and Memory layer by the injected content, manipulation of the Reasoning layer, and execution of an action based on the infected instruction.

As stated by one participant, a typical attack against an agentic AI system with deployment capabilities would involve the input of malicious instructions disguised in deployment-related content, the influence of the reasoning process through manipulation of memory, and the execution of an unsafe instruction despite execution validation checks.

Another participant further emphasised that agentic attacks are dangerous because they are not simply attacks against specific elements. Unlike regular software, attacks against agentic AI agents spread through multiple layers because of their reasoning chains, memory storage, and autonomous workflow.

Moreover, the cross-layer nature of agentic AI attacks significantly complicates security analysis because the attacks are no longer limited to a single event but need to be analysed as operational workflows. One participant compared this issue to traditional attack kill chain analysis, arguing that agentic attacks create a completely new type of threat due to reasoning capabilities.

The overwhelming agreement of all participants provides solid evidence supporting this proposition.

Therefore, P3 is strongly supported by expert feedback.

P4: Attack Workflow Proposition

The fourth proposition claims that attacks in agentic AI follow the pipeline defined in the framework: Interaction → Context → Reasoning → Execution.

The participants largely supported this proposition while making certain clarifications. Most participants explained that agentic attacks follow this pattern because it accurately depicts how agentic systems work. Participants repeatedly mentioned that malicious instructions usually infect systems from the top layer down and influence their functioning by contaminating lower layers. However, participants emphasised that attacks cannot be considered strictly linear because they create iterative loops within the process.

For instance, a tool output can become contextual data which triggers reasoning. As a result, the reasoning process leads to execution and generates outputs. As explained by participants, this iteration loop may also include backward propagation of attacks originating from the lower layers, where attackers could initiate attacks from compromised external tools, poisoned MCP servers, or plugins. Such a loop amplifies the whole process by increasing potential failure and attack vectors. Finally, the participants pointed out that the effectiveness of any layer is multiplied by poor governance controls. Permissions, authorisation, and tool access become extremely vulnerable if reasoning fails. Overall, while the findings support the validity of the presented pipeline, they also suggest that future versions of the framework must accommodate iteration and dynamic propagation patterns.

Therefore, P4 is supported by expert feedback.

P5: Prompt Injection Centrality Proposition

The fifth proposition suggests that prompt injection is among the critical vulnerabilities of agentic AI systems.

The participants strongly supported this proposition. Participants compared prompt injection attacks to traditional SQL injections in regular software systems, noting that both attacks exploit the way in which the system interprets inputs. Participants also suggested that prompt injection is even more dangerous than SQL injection because it influences the reasoning process itself.

According to the participants, agentic AI systems receive a tremendous amount of input data from users, websites, web APIs, emails, tool outputs, documents, and external databases. Due to the inability to separate data from instructions in LLMs, malicious inputs can become interpreted as prompts. Participants emphasised that one of the key dangers of prompt injections is their indirect nature, as indirect injections increase scalability by affecting multiple instances, avoid the need for direct user interaction, originate from trusted external sources, are harder to detect, and contaminate the contextual memory.

As stated by one participant, the danger of agentic AI increases exponentially because even a single poisoned page or document could harm multiple agent sessions. Another participant

emphasized that prompt injection attacks become especially dangerous in the context of memory persistence and autonomous action capabilities.

Finally, participants confirmed that prompt injection attacks can impact multiple layers at once, as a malicious prompt can infect the interaction, contaminate contextual memory, change reasoning outcomes, and influence the execution.

Therefore, P5 is strongly supported by expert feedback.

P6: Indirect Attack Severity Proposition

The sixth proposition postulates that indirect attacks and vulnerabilities represent a bigger security risk compared to direct attacks. The findings provided strong support for this proposition. According to participants, indirect attacks are especially difficult to handle due to their expanded surface, as the sources of malicious prompts might include web pages, documents, APIs, searches, plugins, MCP servers, and agent plugins.

It was emphasized that indirectness of the attack implies that it originates from external elements and is based on implicit trust. Besides, indirect attacks are also scalable as a single malicious element might impact multiple agentic instances. Moreover, indirect attacks were also considered hard to detect as they are camouflaged as normal operational procedures. Finally, participants noted the possibility of attacks based on compromise of third-party tools.

The agent ecosystem will only exacerbate these issues due to the increased connectivity of agents and services. Therefore, the obtained results support the assumption that indirect attacks are among the most challenging forms of threats against agentic systems.

Therefore, P6 is supported by expert feedback.

P7: Practical Applicability Proposition

The last proposition is related to the practical applicability of the proposed framework and its ability to be used for different security-related activities.

Most of the participants supported this proposition. Some participants mentioned potential uses of the framework such as threat modelling and evaluation, security assessment and audit, classification and prioritisation of vulnerabilities, governance of AI evaluation, educational purposes, investigation and incident response, and raising security awareness regarding AI technologies.

One participant specified that the presented layered approach can be used as the basis for a defence-in-depth strategy for engineers. Another participant said that the framework could improve collaboration through visual representation of information which is understandable for both technical and business stakeholders. Finally, participants suggested potential improvements for the presented framework, including a benchmarking tool, a maturity model,

an approach to evaluation, and a control framework. In particular, participants recommended expanding the framework using concepts such as NIST cybersecurity frameworks, CIS controls, OWASP methodology, standard threat modelling processes, and kill chain analysis. Apart from that, participants recommended applying severity scoring, defensive measures, monitoring, differentiating trust, and separation of layers of authentication to the framework. Although participants pointed out certain areas of improvement, their feedback clearly showed that the presented framework is perceived as a valuable tool for practical security work. Therefore, P7 is supported by expert feedback.

5.4 Discussion

in this respect, the results of the analysis provide strong arguments in favor of validation of the developed model.

First of all, the participants agreed that the vulnerabilities of agentic AI cannot be considered merely from the perspective of purely technical aspects but should also be assessed based on the operations performed during the whole life cycle of the system in question.

Significance of the finding mentioned above is associated with the fact that it proves the relevance of assumptions underlying the development of the proposed model. Moreover, unlike the majority of traditional software, agentic AI is capable of performing operations such as processing of external data, contextual memory building, autonomous reasoning, interactions with other agents, and iterative workflow performance. Thus, the finding mentioned above provides additional aspects to be taken into account during the cybersecurity assessment of agentic AI, and another significant finding pertains to prompt injection attacks which become the target for agentic AI agents.

Based on the opinions expressed by the participants of the study, prompt injection could be regarded as an example of another type of cybersecurity threats impacting the operations of agentic AI agents. Importantly, the findings mentioned above completely correspond to those revealed as a result of the review of relevant literature sources discussed in Chapter 2.

Furthermore, yet another significant finding of the current research relates to cross-layer propagation of vulnerabilities in agentic AI. Specifically, all the vulnerabilities in question become evident throughout the life cycle of the agent in question. The finding mentioned above provides further evidence for the relevance of using the concept of layers when assessing the cybersecurity risks of agentic AI. However, despite numerous advantages of the model, several drawbacks could be outlined based on the feedback from the respondents. For instance, in the

opinion of the majority of the respondents, linear representation cannot accurately describe the processes performed by agentic AI agents. More precisely, the participants stated that the operations performed by agentic AI include processing of outputs, contextual memory updates, reasoning session launches, and workflow launches.

In this respect, the finding mentioned above suggests incorporating this feature into the future versions of the framework.

In addition, another crucial finding relates to security governance and future perspectives of the proposed model. Specifically, according to the participants of the survey, despite successful reflection of current cybersecurity issues in the model, there is still a need to develop a security standardisation tool. In particular, the participants suggested including such components into the model as control mapping, governing layers, benchmarking, identity management, authorisation, and monitoring. The significance of this finding is explained by the fact that it outlines the directions for improving the framework in question in the future.

Finally, according to the participants, increasing attention should be paid to tendencies such as multi-agent environment development, supply chain attack emergence, MCP architecture, tool provenance, and third-party integration. These results show how much potential exists within the agentic AI ecosystem and also confirm the importance of integrating these elements of the environment into any research that is going to happen on AI security in the future

5.5 Proposed Enhanced Agentic AI Framework

Therefore, based on experts' opinions analyzed and results of evaluation of the Agentic AI Vulnerability Framework, it would be reasonable to improve this framework and enhance it with additional elements. Even though the proposed model describes the spread of vulnerabilities across levels in detail, certain gaps about the working environment had to be introduced into an enhanced framework.

Apart from initial components such as Interaction Layer, Context and Memory Layer, Reasoning Layer, Tool and Execution Layer, and Governance and Trust Layer, the enhanced Agentic AI Vulnerability Framework is comprised of the same five factors explaining the vulnerability spread process. Nevertheless, the framework was improved with specific additions to provide greater adaptivity and flexibility and increase the security level of an agentic AI system.

To start with, unlike the initial framework, the enhanced model incorporates feedback and recovery concepts. Specifically, the attack containment, rollback, logging, and incident

management are added. Thus, these components serve to outline possible steps during the operation in case an attack takes place on an agentic AI system.

The second key component of the improved framework involves the idea of continuous monitoring and behavior/performances analysis. Inasmuch as an operation of an agentic AI system implies constant changes that lead to the emergence of multiple vulnerabilities, routine behaviors/performance analysis is needed.

Moreover, risk prioritization serves as another essential part of an enhanced Agentic AI Vulnerability Framework. Indeed, due to the uncertainty of the emergence of certain vulnerabilities, experts recommend implementing risk prioritization for determining the most relevant ones along with their potential impact.

Finally, one should bear in mind the role of people in decision making. Considering that the risks of fully autonomous decision-making are rather high while interacting with resources of other entities, manual approval and verification should be applied in order to avoid major issues.

In summary, the enhanced framework represents additional features that enable a proper operation in varying conditions.

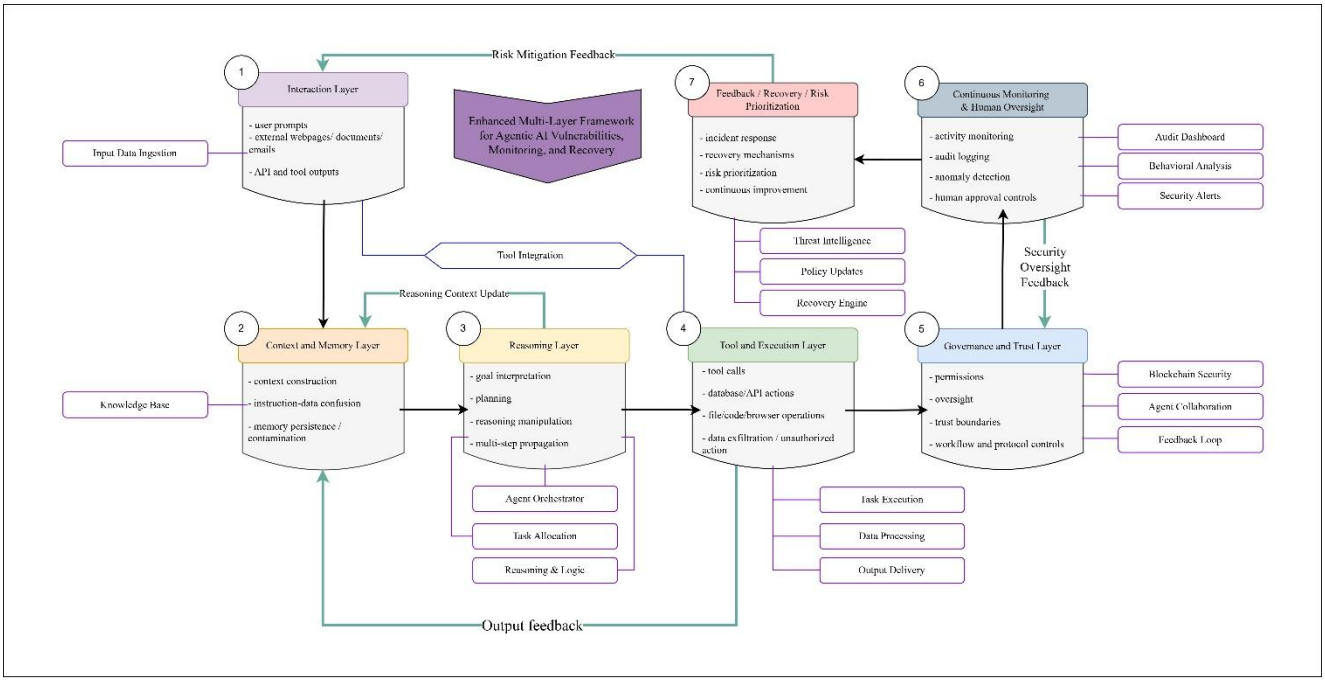

Figure 4: Enhanced Multi-Layer Framework for Agentic AI Vulnerabilities, Monitoring, and Recovery

Chapter 6
Conclusion and Future Work

6.1 conclusion

This paper considered the problem of agentic AI vulnerabilities and introduced the framework for vulnerability assessment employing the operational approach towards agentic AI development. Compared to traditional computation techniques used in machine learning, agentic AI is characterised by a significantly increased level of functionality in such aspects as reasoning, planning, memory persistence, use of tools, workflow execution, and interaction with the environment.

While providing greater relevance of agentic AI systems, these features bring about a new set of challenges in terms of security of such systems, which become much harder to mitigate than in case of software-oriented approaches. This paper considered the transition from traditional language models to agentic AI agents and identified the key vulnerabilities of such technologies. Among the most important vulnerabilities of agentic AI, the following may be distinguished: prompt injection, context poisoning, memory poisoning, goal hijacking, reasoning manipulation, tools abuse, privilege escalation, and governance-related vulnerabilities.

Based on the results of the literature review and discussions concerning the research gap, a five-level framework for assessment of agentic AI vulnerabilities has been suggested. The framework is organised into five layers, namely the Interaction Layer, the Context and Memory Layer, the Reasoning Layer, the Tool and Execution Layer, and the Governance and Trust Layer. Using such an approach enables the assessment of the entire attack chain as part of the agentic AI system's operation. In addition, one of the key assumptions about the dynamics of agentic AI vulnerabilities involves the idea that such vulnerabilities never occur in isolation at a particular operational layer. On the contrary, attacks tend to proceed from layer to layer in agentic AI, moving from the Interaction Layer to the Governance and Trust Layer. This assumption has been fully corroborated by the results of the evaluation process, with all participants stating that attacks proceeded from the Interaction Layer to the Context and Memory Layer, influenced the Reasoning Layer, and ultimately resulted in consequences at the Execution Layer. This result may be considered as one of the contributions of this paper, which proves the importance of the operational approach to the assessment of AI vulnerabilities.

Another major contribution involves the identification of prompt injection vulnerabilities among the most serious threats associated with agentic AI systems, which may be viewed as one of its major vulnerabilities. In contrast to traditional vulnerabilities related to programming errors, prompt injection exploits the capability of agentic AI models for reasoning and construction of contexts, increasing cybersecurity challenges associated with them dramatically. Moreover, the results obtained prove that the use of the Tool and Execution Layer for the evaluation of agentic AI vulnerabilities becomes crucial in order to translate information attacks into real risks, which could not happen in the case of conversational AI models. In addition to the limitations discovered within the scope of this research, there were also certain recommendations provided by participants of the project, including dynamic feedback loops, multi-agent support, governance maturity models, benchmarking security, threat modelling, monitoring and observability, identity and authorisation management, and severity scoring.

Therefore, these results confirm that cybersecurity for the next generation of agentic AI should be based on the operational approach to vulnerabilities. This paper makes an important contribution to existing literature on AI security and suggests a five-layer framework for agentic AI vulnerability evaluation. Finally, such a framework forms a basis for further research in areas such as AI threat modelling, security of agent design, agentic AI governance, autonomous system auditing, prompt injection attacks, security of multi-agent systems, and security benchmarking.

6.2 Future Work

While the proposed framework has provided an excellent conceptual basis for vulnerability analysis in agentic AI, the findings of this study have also identified several important areas for further research and development of the concept. A key improvement to be made to the framework in the future is to make it more dynamic and iterative. Many modern agentic AI systems operate according to the principles of a recursive reasoning cycle where the output generated during system execution affects its contextual memory, reasoning operations, and future actions. Accordingly, the framework of the future should incorporate features such as feedback loops, recursive workflows, iterative reasoning paths, dynamic context updates, and execution-to-context propagation.

The other key direction in developing the framework is to ensure its applicability in multi-agent environments. As many AI systems require collaboration with other autonomous entities, vulnerabilities may spread not only inside the target system but also across interacting platforms. Future studies should therefore pay attention to factors such as agent-to-agent trust,

shared memory contamination, distributed reasoning manipulation, cascading compromise propagation, and multi-agent governance.

The importance of governance maturity and effective evaluation has also been highlighted. In order to enhance the effectiveness of the framework in the future, researchers can consider introducing features such as NIST-inspired maturity models, CIS-style control frameworks, AI security benchmarking, severity scoring systems, threat actor modelling, and risk prioritisation methodologies.

The need to implement security monitoring has also been mentioned by the interviewees. The increasing level of autonomy and complexity of modern AI requires enhanced visibility in all reasoning processes, memory alterations, and tool utilisation activities. The future version of the framework should accordingly include features such as real-time monitoring, logging systems, behavioural anomaly detection, reasoning trace analysis, memory integrity validation, and tool provenance tracking.

Another direction for further research into agentic AI vulnerabilities includes practical application. In the future, researchers can consider implementing their framework to create operational methodologies that would enable functions such as AI security auditing, AI red teaming, vulnerability assessment, prompt injection testing, secure agent evaluation, and automated attack simulation.

Integrating security controls into the framework also appears to be an important area for future investigation. Several interviewees have emphasised that the future framework should address not only vulnerabilities but also define defensive controls, trust boundaries, authorisation mechanisms, governance policies, and security best practices.

It is equally important to assess vulnerabilities arising during the creation of agentic AI systems. In particular, interviewees noted the importance of considering potential issues such as training data poisoning, alignment failures, model creation weaknesses, supply chain compromise, and foundation model vulnerabilities.

Overall, as agentic AI technology continues moving toward ever-greater autonomy and interconnectedness, there will emerge an ever greater need for frameworks to ensure its secure and responsible utilisation. Therefore, further research into agentic AI vulnerability, governance, and security engineering remains highly important.

References

Abou Ali, a. D., 2025. *Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions,* s.l.: arXiv.

Anshuman Chhabra, S. D. S. K. N. P. M., 2026. Agentic AI Security: Threats, Defenses, Evaluation, and Open Challenges. *IEEE Access,* p. 29.

Chiang, L. ,. H. a. C., 2025. *WHY ARE WEB AI AGENTS MORE VULNERABLE THAN STANDALONE LLMS? A SECURITY ANALYSIS,* s.l.: arxiv.

Chiang, L. H. H. C., 2025. *Why Are Web AI Agents More Vulnerable Than Standalone LLMs? A Security Analysis,* s.l.: arXiv.

Debenedetti, Z. B. B.-K. F. T., 2024. *A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses,* s.l.: Edoardo Debenedetti.

Deng, G. H. M. X. W. a. X., 2025. AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways. *ACM Computing Surveys,* 57(7), pp. 1-36.

DENG, G. H. M. X. W. a. X., 2025. *AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways,* s.l.: ACM.

Ferrag, T., 2025. *From Prompt Injections to Protocol Exploits:Threats in LLM-Powered AI Agents Workflows,* s.l.: arXiv.

Flehmig, L. a. Y., 2025. *Perspectives on a Reliability Monitoring Framework for Agentic AI Systems,* s.l.: arXiv.

Giusti, W. ,. T. ,. C. T. ,. P. ,. M. ,. R. L. d. A. C. R. S. ,. S., 2025. *Federation of Agents: A Semantics-Aware Communication Fabric for Large-Scale Agentic AI,* s.l.: arXiv.

Gulyamov, G. ,. K. M. ,. ,., 2026. *Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities,Attack Vectors, and Defense Mechanisms,* s.l.: MDPI.

Gulyamov, G. R. ,. K. ,. M. ,. ,. R., 2205. *Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms,* s.l.: preprints.

Hamilton, a. A., 2026. *Neuro-symbolic AI for predective Maintenance(PdM) - review and recommendations,* s.l.: arXiv.

Khan, S. ,. a. M., 2024. *SECURITY THREATS IN AGENTIC AI SYSTEM,* s.l.: arXiv.

Lbath, A. ,. D. O., 2026. *AVIATOR: Towards AI-Agentic Vulnerability Injection Workflow for High-Fidelity, Large-Scale Code Security Dataset,* s.l.: arXiv.

Liu, Z. L. Z. W. L., 2025. *"Your AI, My Shell": Demystifying Prompt Injection Attacks on Agentic AI Coding Editors,* s.l.: arXiv.

Narajala, O. N., 2025. *Securing Agentic AI: A Comprehensive Threat Model and Mitigation Framework for Generative AI Agents,* s.l.: arXiv.

OWASP, 2025. *LLM01-prompt-injection.* [Online] Available at: https://genai.owasp.org/llmrisk/llm01-prompt-injection/ [Accessed 13 April 2025].

OWASP, 2025. *llm062025-excessive-agency.* [Online] Available at: https://genai.owasp.org/llmrisk/llm062025-excessive-agency/ [Accessed 13 April 2026].

OWASP, 2025. *OWASP Top 10 for LLM Applications 2025,* s.l.: OWASP.

Saleh, D. V. T.-S. H., 2025. *Self-Evolving Multi-Agent Network for Industrial IoT Predictive Maintenance,* s.l.: arXiv.

Sapkota, R. ,., 2025. *AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges,* s.l.: arXiv.

Yorke, B., 2024. *Quantifying AI Vulnerabilities: A Synthesis of Complexity, Dynamical Systems, and Game Theory,* s.l.: arXiv.

Yuxuan Zhu, A. K. D. B. P. L. A. G. A. D. R. F. C. J. E. I. J. B. J. G. A. D. S. R. K. Y. T. S. D. K., 2025. *CVE-Bench: A Benchmark for AI Agents' Ability to Exploit Real-World Web Application Vulnerabilities,* s.l.: arXiv.

Zhan, L. Y. a. K., 2024. *Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents,* s.l.: Qiusi Zhan.