LEARNING MADE EASY

Cequence Special Edition

Agentic AI Security

for dummies®
A Wiley Brand

Secure your AI agents

See APIs as the security perimeter

Ensure compliance

Brought to you by

CEQUENCE

Steve Kaelble

About Cequence

Winner of Both the Inc. and Built In 2024 Best Workplaces Awards

Cequence's mission is to protect today's hyper-connected enterprises from fraud, business abuse, data losses, and non-compliance on web, mobile, and API-based applications that connect their employees, customers, partners, and suppliers. Cequence provides runtime API visibility, security risk monitoring, and patented behavioral fingerprinting technology to consistently detect and protect against ever-evolving online attacks.

Organizations that have fully embraced an API-first methodology or are just getting started, trust Cequence to protect their APIs and scale their business with the only solution that addresses all phases of the API protection life cycle. For more information, visit www.cequence.ai.

Agentic AI Security

Cequence Special Edition

by Steve Kaelble

for dummies®
A Wiley Brand

Agentic AI Security For Dummies®, Cequence Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-394-37963-7 (pbk); ISBN 978-1-394-37964-4 (ebk); ISBN 978-1-394-37965-1 (ePub)

Publisher's Acknowledgments

Project Editor and Manager: Carrie Burchfield-Leighton
Acquisitions Editor: Traci Martin
Senior Managing Editor: Rev Mengle
Client Account Manager: Jeremith Coward
Production Editor: Umeshkumar Rajasekhar

Table of Contents

Introduction

Artificial intelligence (AI) has really captured the world's attention since the release of public chatbots in recent years, but it's anything but new. Apple's Siri has been assisting users for a decade and a half, IBM's Deep Blue beat a world chess champion in 1997, and the term *artificial intelligence* can be traced to the mid-1950s. What really *is* cutting edge, though, is agentic AI.

I'm talking about AI that really gets things done. It can autonomously handle complex, multistep tasks and workflows. It's AI that can analyze data, make plans and decisions, and then orchestrate independent actions with the help of multiple AI agents, operating through application programming interfaces (APIs). Agents, in fact, are really just bidirectional APIs, connecting and achieving things together.

The possibilities and power of agentic AI seem limitless. But if you've been dealing with information technology (IT) long enough, you know that every new development brings all-new risks, and agentic AI is no different. Indeed, just as AI can really scale up capabilities, it can also scale up potential troubles.

In this world, threats such as prompt injection can cause agents to do things they're not supposed to do, malicious bots can pose as good agents, and helpful goals can be hijacked for nefarious purposes. At the frontlines of the agentic AI risks are APIs. To achieve real security in the world of agentic AI, you have to intimately understand your API landscape, fully protect all APIs, and see them as the new security perimeter.

About the Book

Agentic AI Security For Dummies, Cequence Special Edition, is your introduction to safely operating in the exciting new world of agentic AI. With insights from a leader in API protection, this book outlines how agentic AI functions, how it relies on APIs, how it makes connections to apps and third-party resources and pertinent data, and why it's vital to secure all APIs that are part of the magic. You discover the threats that prey on agentic AI, the potential consequences, compliance considerations, and detailed steps you must take to build secure agentic AI with agent-resilient API architectures.

Foolish Assumptions

Knowing your audience is important, right? So in writing this book, here are a few assumptions I made about you, the reader:

- You're a leader connected to technology — maybe head of IT or cloud security, maybe an ecommerce expert, or you could be a growth-oriented leader on the business side.
- You're very interested in AI uses and applications but may not feel as well-versed in agentic AI.
- You'd appreciate a high-level discussion of agentic AI, its capabilities, its risks, and its security imperatives.

Icons Used in This Book

I've packed a lot of information into this book but also some guidance on how to handle that info. Check the margins for these icons:

REMEMBER

If you're in a hurry and can't quite get through every word, at least read the paragraphs marked by this icon.

TIP

I hope you enjoy reading this book, but the ultimate goal is to offer helpful, actionable insights. This icon points to those.

WARNING

This being a book about security, it's all a warning, but this icon points to a specific concern to be aware of.

Beyond the Book

I hope you discover a lot in this book, but I also recognize that you're not likely to be finished with this topic when you put the book down. In that case, here's a place to turn for more insights: `cequence.ai`. This technology provider protects your apps and APIs from all the risks I outline in this book. Click this link for solutions, services, resources, and blog posts.

IN THIS CHAPTER

- Empowering agents
- Taking action with APIs
- Recognizing APIs as the security perimeter
- Getting to know your APIs

Chapter 1
Meeting the Machines

Generations ago, the Industrial Revolution transformed society as manual processes were mechanized and innovated, manufacturing blossomed, and things got done more efficiently. The Information Age shifted much of the focus from the industrial economy to data and information technology (IT). Agentic artificial intelligence (AI) is another huge transformation, allowing IT to really get things done.

This chapter explains what agentic AI is, how it works at a high level, how application programming interfaces (APIs) and Model Context Protocol (MCP) are vital connections that make it happen, and why it's essential to gain full visibility into all APIs across the ecosystem.

Calling in the Agents

Good stories often begin with the words *Once upon a time* or maybe *In a land far, far away* (or perhaps a galaxy). The average person may think the story of AI is too new to begin with *Once upon a time* — after all, it was just late 2022 when ChatGPT came along and got everyone talking.

That said, milestones in AI go back at least as far as the 1950s, when Alan Turing proposed the idea of a test to see how much

a machine may be able to exhibit human-like intelligence. So, there's some history to this story. And it's happening here, not just in a land or galaxy far, far away.

In any case, regardless of how far back you want to start the story, you can reasonably assert that major plot twists are happening practically all the time. The democratization of AI with ChatGPT and other public services was a big one — now, you really *can* try this at home.

And what's happening more and more these days is the powerful development known as agentic AI. For enterprises, AI isn't just a tool for generating content or making predictive analyses; it's also for powering through complex workflows that include very concrete, autonomous actions.

That may sound a lot like automation. But looking at agentic AI and comparing it to other kinds of automation — even some of the more recent developments in robotic process automation (RPA) — is like observing a toddler on a tricycle compared with a pilot in a fighter jet.

Traditional automation often follows some kind of a script. If certain conditions are met, the system takes a certain action. It can be as simple as the grocery store door opening when you approach it, or something more advanced like RPA software handling repetitive, rule-based tasks on a computer. That kind of thing is relative child's play for agentic AI.

Similarly, many of the kinds of AI that have captured global attention in recent years — from machine learning efforts to predictive analytics to generative AI chatbots — tend to have limited outputs. That's not to say they aren't powerful, but they're typically coming up with some sort of classification, or a forecast, or a selection of text, or maybe a meme illustration that goes viral on social media.

REMEMBER

Agentic AI, on the other hand, is all about getting much more complex things done with autonomy and adaptability. It's not just an agent completing a task — it's a whole cast of individual agents collaborating on a workflow, delegating subtasks, prioritizing, taking context into account, and refining approaches as needed to achieve a certain goal.

To put it another way, agentic AI is the orchestration layer that works to achieve some high-level goal involving multiple steps

and inputs and possibilities without the need for humans calling each and every shot. It's tapping into enterprise data and all sorts of other information and calling APIs to connect with various players on the agentic AI team.

All kinds of benefits are tapping into agentic AI. Here are a few:

- **Speed and efficiency:** Agentic AI can take over workflows and processes that used to require a bunch of human hands and brains and oversight, coordinating multiple systems. Tasks and overarching goals can be executed and achieved much more quickly, and potentially with fewer errors.
- **Adaptability:** Agentic AI can deal with complex situations and changing environments because it can learn from results and change course as needed.
- **Scalability:** This is a benefit of all kinds of technology advances, of course. Agentic AI can ramp up or down its capabilities as demand changes.
- **Freeing up human brainpower:** You've heard this benefit before, too, attached to simpler forms of automation. Because agentic AI can take on complex workflows and goal-oriented tasks, humans can spend more time on creativity, innovation, and strategy.

REMEMBER

The use cases for agentic AI are evolving and seemingly limitless. Agentic AI systems can efficiently take over human-intensive aspects of healthcare operations, for example. They can coordinate scheduling, tackle insurance pre-authorizations, and handle other administrative tasks before care happens, and then create personalized patient education and track reimbursements afterwards.

Agentic AI can orchestrate supply chain management details, by monitoring inventory, placing orders as needed, and then handling the logistics. Customer-support agentic AI can be handed a customer issue, figure out what needs to be done to resolve it, check account information, address billing errors, write customer responses, and do many other coordinated tasks that vary from one situation to the next.

These kinds of use cases involve multiple tasks — the details of which vary based on specific conditions and contexts and priorities that the agentic AI orchestration layer must coordinate autonomously. Orchestration layer, as it happens, is a great term

for this because it conjures up images of a bunch of individuals making beautiful music together in a coordinated way.

To carry that musical metaphor one step further, it may be helpful to think of this less like carefully scored Mozart and more like Coltrane or other jazz improv masters — or maybe even the Grateful Dead or Phish. That's because agentic AI can adapt to changing inputs and situations and improvise new paths as a result.

Getting Things Done with APIs

As you ponder whether to envision a conductor in front of an orchestra or a bearded guitarist heading down an improvisational tangent, here's a totally different image to think about: that old-fashioned telephone operator sitting amid a bunch of wires plugged into different holes on a switchboard. That operator is making connections, making calls go through by plugging each wire into the right place.

That's a simple way to think about how agentic AI gets things done — placing a lot of calls to a lot of different places. It requires interactions with many independent parts and data sources and applications. The connection points bringing together all these diverse team members are APIs. Agentic AI relies on calling APIs and running multistep orchestration in the pursuit of a high-level goal, without the need for constant human direction.

REMEMBER

APIs are essentially a universal language making it possible for various systems to exchange data and understand commands. The agentic AI system is the orchestration layer — a non-human operator plugging in the connections. The APIs it employs allow it to gather contextual data and string together actions across multiple applications. To outline just one example, it may grab customer data, analyze it using another service, draft an email through a generational AI model, and use a marketing platform to schedule the email.

That agentic AI orchestration layer calls APIs to power through the workflow. Foundational models bring in natural language capabilities and reasoning. Memory systems help keep context at the forefront across the steps. External data sources provide details and situational awareness. APIs tie it all together so the team of agents can get the job done.

The capabilities are truly amazing, and you may think this is the happy and triumphant end of the story. But this is the point in the movie when you start hearing the discordant, suspenseful music starting to swell, telling you that danger lurks. That's because the power of agentic AI has brought with it new risks.

WARNING

Every integration that agentic AI has tapped into has expanded the attack surface, creating more elements to attack and more potential entry points for malicious actors to try to exploit. The more you do through APIs, the more you risk being overwhelmed by what some refer to as *API sprawl*, a complicated landscape of endpoints that's increasingly difficult to monitor, secure, and protect. Agentic AI isn't necessarily the origin of all API sprawl, but it threatens to make it worse.

In other words, what makes agentic AI amazing, the way it can orchestrate workflows across multiple services, makes it all the more necessary to strengthen strategies for understanding, managing, authenticating, and monitoring APIs. To be clear, for agentic AI to work the way it is supposed to, these synthetic and autonomous relationships and actions are essential — it's the whole point. But it raises all new challenges for those in charge of security.

THE NEW CONNECTIVE TISSUE OF MCP

The capabilities are getting more powerful all the time as new technologies and concepts develop. One is the emerging framework known as MCP. It standardizes how agentic AI systems can interact with data, tools, and environments, and it makes it easier for the large language models of the AI world to engage with the APIs required to get things done.

With MCP, which burst onto the agentic AI scene in 2024, pieces of the puzzle that were static and not fully aligned become much more closely connected. It's like new and more powerful wiring between the brain and the rest of the body.

MCP makes it much more possible for agents to act across different systems, doing things like requesting context and calling APIs, without the need for custom integrations for every situation. APIs remain the underlying mechanism for connecting with external services and data. The difference is, before MCP, every possible interaction needed to be

(continued)

(continued)

hard-coded, but now AI agents have access to a common language that helps them share context and information.

Agentic AI, turbocharged by MCP, is better able to orchestrate multi-step tasks, chain actions, and grab data from diverse sources. It's a new kind of connective tissue to make agentic AI all the more powerful.

Seeing APIs as the Security Perimeter

The idea of a security perimeter used to be so easy to envision. Check the dictionary definition of the word *perimeter*, and it talks about a continuous line that forms around a boundary.

Makes sense — think about a moat around a castle or perhaps a razor wire fence around a prison. Those are nice and secure, of course, but when it comes to efficiently and rapidly doing business that involves transactions and communications and information sharing, putting up a high and impenetrable wall isn't the least bit practical.

That is, of course, not news to people who work in IT security. Their days of building a trusty firewall to secure a well-defined perimeter are long gone, thanks to ever-more-complex architectures that reach across multiple clouds and as-a-service applications and widely dispersed data sources and services.

REMEMBER

When you're talking about an agent-driven AI landscape, there's certainly no traditional network edge. The security perimeter effectively becomes the APIs that pull that landscape together and handle interactions, data exchanges, and the execution of actions. With autonomous orchestration of agentic AI, all the various transactions and queries and decisions flow through APIs. Malicious actors know that APIs are the prime attack surface that allows direct connections to the prizes they seek.

This reality raises a number of risks. For one thing, any gaps in authentication and authorization become dangerous vulnerabilities. If API keys are weak or Open Authorization (OAuth) flows misconfigured, for example, attackers may be able to impersonate agents or escalate privileges.

HOW INTEGRATIONS CAN OPEN THE DOOR TO ATTACKERS

An example of the way integrations can become attack vendors happened in the summer of 2025, when attackers hijacked OAuth tokens from integrations between Salesforce and Salesloft and its Drift subsidiary. Multiple organizations were impacted, and customer data became exposed without setting off the usual security alarms.

It wasn't a vulnerability on the part of Salesforce but instead a supply-chain attack involving a trusted third party. The attackers impersonated Salesforce integrations by using tokens that were stored in the Salesloft infrastructure. The tokens already had access to Salesforce data, so without breaching Salesforce directly, they were able to use legitimate API calls to dig into sensitive records.

It's a prime example of how the software-as-a-service world, with widely interconnected apps, has a huge and attractive attack surface and wide blast radius. A single compromised token or identity can lead to a world of mischief.

Meanwhile, any endpoints that are too permissive or not well designed may make way for breaches of sensitive information. And we mentioned API sprawl above — as agentic AI scales up and encompasses more and more APIs, it makes blind spots all the more likely.


TIP

Given that, APIs need to be treated as the security perimeter that they effectively have become. It's imperative to really get to know the API landscape, continually scan for vulnerabilities, and make zero-trust access the rule.

Discovering and Inventorying APIs

It's well established that you can't achieve security in any IT environment if you don't have full visibility into all that you need to secure. That's a truth that goes back well before IT was a thing. That castle we described earlier would have had not just a moat, but also stone towers with guards assessing the landscape. Every entrance and vulnerability would be known and protected.

That's not so simple with an agentic AI landscape. While it would be pretty easy to know about the small number of entrances of the castle, agentic AI has all kinds of APIs, dozens or even hundreds of them, each one of them a potential attack target with keys to the castle. There's no way to secure that attack surface without first discovering all APIs involved in the work of agentic AI and creating a complete inventory.

In other words, visibility. Security teams and systems have no way to adequately defend what they can't see and don't know about. APIs are out there interfacing with external systems, triggering or participating in workflows, and gathering data. A complete inventory of them is the foundation for real security, the only way you won't be plagued by such issues as forgotten integrations, shadow APIs, or unknown endpoints that could be accessible and exploitable.

Gaining visibility into the absolutely complete API landscape helps create a baseline for security controls. After you know which APIs are out there, who uses them, and what data they handle, that allows authentication, authorization, and monitoring. Governance needs to be continuous.

TIP

The problem is that the API landscape is anything but static. Agents are constantly evolving and new tasks are always being orchestrated, and it's all happening across a distributed environment that may live in many different SaaS applications across multiple cloud providers. Internal microservices may be part of the picture, along with services from third-party partners. To get a handle on this and get the visibility you need, it takes automated and continuous discovery, classification, and monitoring.

It also takes some careful prioritization. You may have a thousand APIs out there, some more of a concern than others. It's important to evaluate which ones may interact with sensitive data, and also which ones don't have authentication and authorization. APIs that check off both of those boxes are likely to be the ones most important to focus on first.

Put another way, discovery requires understanding what actions one can take and how that gets done. It takes a full inventory of APIs to really grasp what's possible to achieve across the ecosystem of APIs. It's best that the good guys gain that understanding before the bad guys do.

IN THIS CHAPTER

- Getting the attention of attackers
- Injecting errant instructions
- Replacing good goals with bad
- Masquerading as agents
- Doing battle against the bad guys

Chapter 2
Facing the Threat of Invisible Intruders

As agentic artificial intelligence (AI) continues to expand its capabilities, it gains more attention from malevolent actors, who dream of the capabilities it offers their evil plans. This chapter explores some of the ways they try to get bad things done, including prompt injection, goal hijacking, and impersonating agents.

Attracting Attackers

You know that oft-cited meme, "Gotta take the good with the bad." Or some people quote it as taking "the bad with the good." It certainly is true that every great advance in information technology also brings challenges, including the threat of new forms of attack. Agentic AI is no exception.

So, do you really have to take that bad stuff in order to benefit from the good? The answer is the whole point of this book — no! You don't have to take the bad — but you certainly have to be aware of it and be ready to deal with it in order to prevent it or mitigate or respond to it.

The rise of agentic AI has malicious actors salivating over the possibilities and planning new attacks for many reasons. To begin with, large language models (LLMs) and autonomous agents tend to really raise the stakes in terms of both opportunity and potential damage. They work incredibly quickly, they make connections across a lot of different systems, and they operate without a lot of human oversight. What that means is

- Any exploited agent or system can then cascade trouble across any number of linked systems. The attack surface is big, and the blast radius can get bigger and bigger very quickly.
- Agentic AI often deals in sensitive data, be it financial data or health information or proprietary business details. Bad actors who can find their way in through these doors have a lot of tempting bounty.
- AI facilitates new ways of getting past defenses, often by sneaking in dangerous instructions. In a way, it's a new twist on trust issues because humans tend to trust AI outputs a bit too much, and AI agents may put too much trust in the instructions they receive.

WARNING

A number of vulnerabilities are relatively new and unique to agentic AI systems, and I go into more detail about some of these later in this chapter. For example, there's prompt injection or indirect injection. That's where the bad guys work malicious instructions into prompts, data, documents, or application programming interfaces (APIs). Agents are tricked into doing harmful things.

Goal hijacking is another term getting a lot of new attention these days. Tactics such as prompt injection may be part of the overall effort to manipulate an AI agent's goals, its overall objectives, and the internal reasoning processes it uses. Agent impersonation is another form of trickery, in which a bot masquerades as a real agent.

It's worth noting that not every bad result happens on purpose. There are times when agents interact in ways that designers didn't consider. This kind of mishap can make it harder to predict vulnerabilities.

And then risks of shadow APIs exist in workflows. There may be undocumented APIs that agents find and call as they pursue their solutions. It may be, for example, that developers have created temporary APIs for experimentation, and have since forgotten

that they're still out there and still callable. This environment of shadow APIs lives outside of security monitoring and governance, which means there are blind spots that attackers may find and exploit.

It's not like shadow APIs are a brand-new thing. But in the world of agentic AI, agents may discover undocumented endpoints through trial and error, and attackers can then exploit the unknown weak points to gain entry.

WARNING

Whatever the new threat, the effect can be devastating. Here's the thing: AI can be really, really effective at what it does, and as long as it's doing good, that's clearly a powerful opportunity. But because it's enabling automation with autonomy, bad actors can turn it against not just a single system but the whole interconnected workflow. After someone sends it down the wrong path, it just may turn out to be really good at doing something really bad.

Here's one other bit of food for thought when it comes to understanding how your AI evolution may be attracting attackers. Everyone has fear of missing out, and no one wants to be left behind. So developer teams are often working at top speed to roll out the latest AI capabilities, including agentic systems. Haste, as you know, sometimes makes waste.

In one recent case, a major global chain launched a third-party tool to help job applicants access AI capabilities as they apply for a position. When labor is tight, helping good applicants connect is a competitive advantage, so the sooner the better when it comes to new tools. Problem is, amid that haste, the developers neglected to change the tool's overly simplistic default administrator password. That simple oops is a reminder that all the basic security rules remain as important as ever, including in the fast-moving world of AI.

Understanding Prompt Injection

Social engineering has for quite some time been a problem that information technology (IT) security teams have had to deal with. The most effective hacks are often not hacks at all, at least not in the classic sense of manipulating code to crack open a door. Attackers get invited inside through social engineering, using some sort of trickery or psychological manipulation to persuade

people to give up sensitive information or access credentials. The truth is, social engineering is just a high-tech version of a con, and people having been conning one another for most of human history.

But here's where the con turns really high-tech. Tactics such as prompt injection are essentially a form of social engineering but aimed at AI agents instead of people. The whole point is to coerce the agent to give up something it wasn't supposed to give up or do something it wasn't intended to do. It's high-tech trickery — and if you can trick an agent, you can get a whole lot of bad things accomplished in no time.

REMEMBER

Simply put, prompt injection involves an attacker purposely creating inputs that change the intended behavior of the LLM. The model is supposed to follow instructions and employ its reasoning capabilities across contexts. This is a matter of sneaking new instructions into user inputs or data, telling the agent to do something it's not supposed to do. It may mean starting out with a prompt that aligns with the intended goal of an agent, then veers in a different direction.

For example, imagine that an agent whose job is to summarize a medical record is told to ignore its previous instruction and then send a sensitive file to a specific external location. If it follows that request, it may have just exfiltrated protected data. It's essentially the AI update to the kind of SQL injection ruses that have been happening for at least a couple of decades — fooling a system into doing the wrong thing.

Prompt injection generally does its dirty work through AI prompts, while indirect injection delivers malicious instructions through other means, such as data sources, APIs, emails, or websites. One way or another, a malicious instruction finds its way to the agent, generally through a source that the agent considers trustworthy. That instruction may cause the agent to make API calls it's not supposed to make, and attackers can even chain instructions to create a workflow with ill intention.

Again, it's a con that's not unlike social engineering, except that the target is an AI agent, which means the result can be extra bad. An attacker takes advantage of an agent that trusts data or prompts too much, or doesn't take care to distinguish between user prompts and instructions that are buried somewhere else.

If an errant instruction seems to be within context boundaries, the agent may be tricked into applying it broadly — and quickly. For example, if an agent that's allowed to check account balances is tricked into taking actions to transfer account balances, you can imagine the scope of the trouble.

Hijacking Goals

If you're a fan of science fiction, you probably recognize the storyline involving a human outsmarting a computer — confusing the machine by pointing out how its goals have shifted from the original intent. Captain Kirk did it in *Star Trek,* confronting the way a computer's overarching goal had deviated from what it was supposed to be. A different twist was in *2001: A Space Odyssey,* in which the HAL 9000 computer experienced a dangerous malfunction caused by conflicting goals and directives.

The common thread is that intelligent computers can wreak havoc when their goals are turned in the wrong direction. As it happens, goal hijacking is a powerful way to turn agentic AI to the dark side and achieve malicious ends.

To understand goal hijacking, remember that agentic AI is the orchestration layer created to fulfill some high-level objective or goal. It makes a plan and executes it autonomously calling APIs, accessing data sources, and using various tools. Goal hijacking happens when bad actors manipulate agents in order to change their perceived objective.

The change may be big or small, but the aim is to hijack the goal without the system or any humans noticing, at least not quickly. A direct hijack may send errant prompts or data into the system, feeding it the wrong instructions. An indirect hijack delivers the instructions through some sort of external content that fits into the workflow, such as emails, webpages, or APIs.

WARNING

The results of goal hijacking could be whatever the malicious actor is hoping to achieve. It could be a financial transfer or exfiltration of sensitive data. The aim could be to simply create chaos by shutting down services or overwriting logs. Goal hijacking could be used to intentionally damage an organization's reputation by way of offensive content, or it may exploit supply chain integrations to enter fake orders or mess with real ones.

It's worth noting that a variation of this problem can actually happen accidentally, when an agent's interpretation of a goal isn't properly aligned with what humans intended. In other words, a misunderstanding. Consider, for example, if an agent is supposed to gather patient information for a research presentation, but doesn't grasp that it's supposed to use only deidentified information. The result could be an inadvertent violation of regulatory rules, and the consequences are steep even if the breach is accidental.

WARNING

This kind of misalignment of goals also can be made worse as agents call APIs that tap into additional data sources. Chaining can make agentic AI more powerful, but it can make mishaps more damaging.

Whether goal hijacking happens intentionally, or goals become misaligned, the results can be devastating. With that in mind, consider that part of controlling the security of agentic AI is keeping control of the goals.

Impersonating Agents

There's probably no con more common than impersonation. Talk about a tale as old as time, you can go back at least as far as the mythology of Zeus and his many disguises, or as recently as this morning, when your cellphone received yet another unsolicited phone call wanting to run a scam. In terms of IT threats, phishing is a scheme that is all about impersonation, fooling unsuspecting users with impersonations of coworkers or other familiar folks.

As it turns out, computers can be fooled by impersonators, too. Agentic AI is made possible by the work of legitimate agents, but it's possible for malicious bots to disguise themselves as legitimate agents. They work their way into the trust model, and all of the APIs and services and other agents mistakenly assume they're working with authorized entities.

It really is the agentic AI equivalent of phishing. The fake agents mimic real agents in the way they behave, in the identities they assume, in their request patterns. And when they're successful, they work their way into sensitive workflows.

How do they do it? They may spoof identities, with fake tokens, phony API keys, or bogus Model Context Protocol (MCP) credentials. They copy the query patterns of real agents, blending in as best they can with legitimate traffic. They may pose as collaborators in multi-agent systems, feeding their bad instructions into legitimate workflows. In some cases, attackers find their way inside environments with weak provisioning, and spin up unsanctioned agents.

WARNING

Here's the problem: If an impersonator gets in the door and is taken for a legitimate actor, it may be able to tap into delegated access to credentials, data stores, and APIs. They can sit there benefiting from their ill-gotten privileges, looking like regular agents because they're mimicking normal workflows — which means it may take time to discover them. And because this is a world of autonomous activities, they may be able to trick other agents into executing bad instructions.

Dealing with these issues is a matter of trust, really. Just as organizations enforce zero trust on humans because you really need to be sure the right people are accessing the right things, it's vital to enforce security with just as much rigor in agentic AI.

TIP

Identity verification is the most obvious need. That means agents must be able to prove themselves through the right kinds of tokens and certificates and cryptographic identities. But authentication needs to also be context-aware. Agent credentials should, when possible, be bound to specific tasks or domains in order to foil imposters. And behavior monitoring helps put the spotlight on any anomalies in agent behavior.

Fighting the Stealthy Threats

Just as some of the threats targeting agentic AI are like threats targeting humans, some of the same kinds of defenses are needed to find and fight these stealthy bad actors. Authentication, for example, is every bit as important in the agentic AI world as it is in human computing interfaces.

The thing is, the threats in agentic AI don't always appear as obvious malware. They often look like normal agents doing normal things — not just by stealing identities but by copying

query styles and requesting cadences. An attacker who goes down the path of prompt injection is using the power of AI itself to turbocharge the perilous possibilities, persuading the AI to make valid-seeming AI calls.


TIP

The kinds of bot management principles used for securing websites can help here. For example, strong identity controls are a must. Traffic pattern analysis should keep an eye on request frequency, chaining behavior, and the origin of requests, so unusual activity is easier to spot.

Prompts are the backbone of agentic AI workflows, so prompt-aware behavior profiling is essential. Prompt fingerprinting outlines how agents typically respond to expected prompts. If they start deviating from that normal behavior, it could mean that injection has happened.

Context boundary enforcement can be quite enlightening, too. If prompts for an agent veer outside that agent's normal domain, that can be a red flag. It's also possible to monitor the semantics to spot hidden, malicious instructions that are embedded in external content or user inputs. And if agents are required to explain the reason behind an API call, it can help shine the spotlight on goal hijacking.

IN THIS CHAPTER

- Staying on the trail of rogue agents
- Exploring API risks
- Gaining and exploiting trust
- Discovering exposures

Chapter 3
Protecting the API Supply Chain

The best way to protect against a threat is to better understand what you're dealing with. This chapter goes into greater detail on the challenges of rogue artificial intelligence (AI) agents, how chaining of application programming interfaces (APIs) can make things far riskier, how trust can be exploited in the new landscape, and how to handle exposures and threats.

Tracking the Rogue Agents

In the world of information technology (IT) security, some bad things happen on purpose, and oftentimes, things happen unintentionally that open the door to security concerns. That's certainly the case in agentic AI. Autonomous agents can be tricked in plenty of ways, and in other various ways, they can unintentionally misuse, over-permission, or expose sensitive APIs. It's vital to keep a close eye out to be sure agents are working for the good of the system, and not heading down the wrong paths.

As I share in Chapter 2, prompt injection and indirect injection can lead to unintended API calls. The output from one step becomes

viewed as trusted input to the next step, when in fact that trusted input may contain dangerous instructions. These rogue instructions may enter by way of user input or such indirect means as emails, webpages, or documents, but however they get there, they cause the agent to do the wrong thing, such as expose sensitive endpoints or data.

WARNING

Tools may be over-permissioned, and ambient authority may allow broader-than-intended access to resources — without requiring specific authorization for an action. It happens when OAuth scopes are too broad, or service tokens live longer than they should, allowing privileged actions that aren't connected to the current goal. It's also an issue when agents share service accounts — if one chain is exposed, it compromises a whole lot more. When you're talking about chained tools, a single overpowered step can magnify the damage.

Goal hijacking can happen on purpose, or drift can happen unintentionally if goals are ambiguous or too long-running. Imprecision in goals or prompts can result in overly broad queries or unfortunately reckless actions. Even when trying to be helpful, agents suffering goal drift may start exploring such things as admin or debug endpoints.

Further trouble is possible if one agent retrieves sensitive information and tags it weakly, and that raw payload is ultimately forwarded to a third-party API for a task such as summarization, thus exfiltrating data. If one tool hands over more fields than another tool needs, sensitive data can be exposed. Or, agents that have file or storage connectors may be erroneously steered to download buckets.

REMEMBER

The bottom line is, there are many ways that bots powered by large language models (LLMs) may seem to be good agents but are actually acting as bad bots. It's vital to tell them apart. Doing so requires careful attention to identity, the intent of agents as they go about their tasks, the way they behave, and the ability to track and verify what they're up to.

A good agent should carry a unique and verifiable identity each time it makes a call. A bad bot may sneak in through a shared account or reuse generic credentials.

Expect a good agent to declare its purpose and intention. What fields does it need, with what limits, or what endpoint? That

allows checks to ensure requests match the purpose. A bad bot may use parameters that don't match the goal, or may avoid declaring its intent altogether. A good agent is likely to leave a clearly articulated trail telling the story of what it's done. With a bad bot, there may be context missing, or sources that aren't verifiable, or links that are broken.

REMEMBER

When it comes to behavior, good agents employ predictable and normal-seeming steps. Endpoints and volumes are consistent; error rates are steady. Bad bots seem to be up to something that's off, whether it's a lot of listing and probing what exists, spikes in downloads, or maybe sudden shifts from read to write.

Think about what you may see when you watch people walking down the sidewalk on a busy urban street. Most are moving forward, minding their own business. Kids may be playing the old step-on-a-crack, break-your-mother's-back game, just like you'd expect a kid to do.

Now imagine you spot a grownup behaving that way. The person is hopping over every crack in the sidewalk, or conversely, may be intentionally stepping on each one. Or maybe doing one of those old Monty Python "Ministry of Silly Walks" routines (if you're nostalgic enough to remember the *Flying Circus*).

It gets your attention. That adult isn't behaving the way a typical adult would. Something is off, and you may want to keep an eye on that person. In a nutshell, that's one way to spot a bad bot, by knowing how an agent usually behaves so you can spot the one that isn't acting normally.

Also worth noting from a behavior perspective is the concept of business logic abuse. It's a common attack technique that appears to be a valid interaction exploiting the intended function of an app or API. The aim is to sneak past traditional detection, and it can happen by way of exploitable design or process flaws, or if the attack involves an action that developers never expected or predicted.

Business logic, for example, may dictate that a certain identity can't log into a service from California right now and from Massachusetts an hour from now. That certainly makes sense. Practically speaking, should an account be allowed to travel, and if so, how far and how fast? That's a question of both policy and business logic, and a successful attacker is one who can figure out how to abuse that logic.

TIP

A few guardrails can help ensure your team is limited to the good guys. The concept of least privilege is nothing new in the world of IT security when it comes to humans, and it's a good idea for agents and tools, too. Each tool gets only the access it needs, and only for the time it needs it.

It can also help to maintain hard allow-lists, spelling out which APIs an agent can call and what operations are allowed. Anything else is off-limits. Also, keep inputs and outputs clean, so any bad instructions are hard to hide. And keep a good handle on where data can and can't go.

Understanding the Risks

The whole power of agentic AI is its ability to get many agents autonomously working together toward a larger collective goal. API chaining is one of the ways that effective and efficient work gets done. The idea is to link together multiple API requests, so that the output of one becomes an input or parameter for another. Presto! API chaining has just completed a larger and more complex workflow.

REMEMBER

Endless potential possibilities exist when you're able to chain together a series of agentic actions. Imagine that you want to go see an upcoming show at the Sphere in Las Vegas. Agentic AI, working through complex chains of APIs, can set up the whole trip.

The system can buy the concert tickets, then check into nearby hotels for availability. It also can start exploring air travel arrangements, plus restaurants and other things to do while there. It knows the order in which to act, to avoid setting up a trip that's fantastic, except there's no place to stay or no way to get there. It's incredibly powerful stuff, but keep in mind that this chain of action is happening autonomously, without the watchful eyes of humans. It had better happen properly, without any glitches or breaches.

Think about accessing a well-protected safe-deposit box at a bank, and you bring your key, and a bank employee has a key, and through a chain of careful key turns you access your sensitive materials. Now imagine that kind of thing happening with no humans in the loop. Does that give you any jitters?

WARNING

In this autonomous setting, API chaining brings a number of inherent risks:

- **Unwarranted trust:** Imagine that the first step in the chain ingests untrusted data. What happens if the next step considers the output trusted and calls a sensitive endpoint? The trouble can cascade to all new places.
- **Expanding privileges:** Each tool has its own permissions, but the chain ultimately combines the permissions in ways that can go awry, with more privileges than is prudent. Over-permissioned agents can also be a real problem.
- **TMI across domains:** An API chain may link records from different systems in ways that collectively may reveal sensitive and impermissible insights. If an API chain in healthcare manages to connect information from medical records with customer relationship management data, that could be TMI, or too much information.
- **Taking different paths:** Upon getting errors, agents using API chains may try backup tools that carry broader access, getting around intended guardrails.
- **Telephone game gone wrong:** As an agent-to-agent chain continues to grow, information may flow from one link to the next. In some cases, the data eventually becomes less accurate than it should be, like the old elementary school telephone game in which messages shared from one kid to another to another become jumbled by the end of the chain.

TIP

How can you tell if these kinds of problems are happening? There are some red flags. For example, watch for calls to endpoints that aren't on the approved path. Or payloads that are erroneously adding fields as they move down the chain.

Let me expand a bit more on the problem of over-permissioned agents. This can happen for a number of reasons. Integrations may be given broad Open Authorization (OAuth) roles, and then chained tools end up inheriting the broadest scope. In other cases, access lives too long, with tokens or keys persisting across goals. And if multiple agents use the same pool of credentials, you can end up with over-permission issues.

Also be aware of data leakage risks. This can take a number of forms. Direct leakage may happen, for example, when a summarizer forwards raw records to an external large language model (LLM) — sensitive information may have just walked out the door.

Indirect leakage happens when sensitive fields inadvertently get carried along when they shouldn't have been, by way of intermediate payloads or caches or logs. And there's the issue of aggregation leakage, in which fields may be harmless on their own, but when they're joined across tools, they allow for sensitive profiles to be reconstructed.

TIP

A number of red flags may reveal data leakage. There may be tools that are forwarding full payloads when they really only need to send a few fields. Maybe there are memory features that persist inputs and outputs (when they really should be forgetting them). Logs with notes or identifiers that aren't redacted can also indicate potential data leaks.

As long as we're talking about risks, let's mention a couple of ways malicious actors can find their way past security. For example, you've no doubt been occasionally annoyed when you encounter a CAPTCHA. It's a challenge and response test designed to help the system know for sure that you're human. (In fact, it's an acronym that's short for Completely Automated Public Turing test to tell Computers and Humans Apart.)

CAPTCHA is an intentional friction point, but there are two problems with it. First, the whole point of agentic AI is to get non-humans to do something for us. Won't CAPTCHA mess up that plan for progress? The bigger problem from a security perspective is that there are sophisticated AI agents that can pass this particular Turing test, so today's systems need to be secured with CAPTCHA-evasive agent requests in mind.

Okay, one more risk to talk about: bot signatures that are hidden behind legitimate agent tools. The problem arises when attackers or rogue automations work their way successfully through whitelisted tools. Calls seem to come from a trusted agent, but they really aren't. These are requests that have been laundered.

Having Trust Issues

Trust makes the world go around, and that's certainly nothing new. Gaining trust among humans is usually a gradual process built on actions and recommendations as well as nuances and gut feelings, plus the passage of time. So, what happens when we invite digital, non-human agents to step into our work and get things done? How do they develop the trust needed to get work done, and are they making the right trust decisions?

WARNING

It turns out that trust is an issue in the world of agentic AI. Agents are making decisions and judgment calls based on trust, and they're not always getting it right.

Take as an example the trust required when you allow your AI assistant to handle some duties involving your calendar. It's not as easy as simply allowing the assistant access to your schedule. It often needs to work with other AI systems to complete tasks. You can imagine the roadblocks that crop up if cooperating systems don't have enough access, but the bigger issue is that permissions are often too broad.

So, imagine you tell your AI assistant to access your calendar for today. It replies that it needs permission for the right calendar data, and you agree to allow access to today's meetings, but only today's. The AI agent, in turn, asks the Model Context Protocol (MCP) server to request today's calendar data from the calendar app.

The MCP server forwards to the calendar app a request to retrieve today's calendar data. What it gets back is all calendar data *as of today* — as opposed to the calendar data *for today.* The calendar app fully trusts the MCP server and gives it what it needs, plus a whole lot more. The MCP server then sends back only the data for today's meetings that the agent requested, and you never even realize that your entire calendar history has been accessed.

It may be a simple misunderstanding, but you've just experienced major data leakage that could be exploited. The MCP server and app server have gone overboard with over-permission, and you wouldn't even know it.

This kind of thing can happen because of the trust granted the middleman, the MCP. The MCP server ends up allowing broader access than you ever intended, and the app server misinterprets the request. Because it completely trusts the MCP server, the service doesn't double-check to be sure the request (or its understanding of the request) really matches what you consented to.

WARNING

Here are the consequences:

- There's been a privacy violation. You think you granted limited access, but broad access was granted.
- The audit trail sure looks like there was proper consent, but the scope of the consent was inadvertently expanded in a way that's not obvious to the paper trail.
- If consent boundaries aren't properly honored, there's legal risk. And there's always a reputational risk when information is inappropriately shared.
- There's also a user trust risk, if you eventually recognize that the system went beyond your very specific permission. Trust, as they say, takes a long time to build and a very short time to destroy.

The trust risks include consideration of identity without context, as well as over-permission grants and reliance on ambient authority. Transitive trust has been granted in a chained situation, and excessive data has been exposed by default.

TIP

The answer to this particular trust problem is a multilayered trust system. This scenario needed not only that initial permission grant at the outset from you; but also it needed the MCP server to validate the request before passing the data request along to the app server. And the app server also needed to validate the request before providing the data.

Any denial along the way stops the release of data and prevents the erroneous expansion of trust. It's not just checking, but double-checking and triple-checking. It's kind of like the old adage to measure twice and cut once (which goes all the way back to the Bible), although maybe I should say measure thrice and cut once.

Of course, one of the benefits of agentic AI is its ability to reach out and connect with all kinds of helpers for getting workflows

done well. That means a lot of third-party API dependencies. Third-party connections are powerful, but they also bring some risks of their own.

For starters, it means trusting a third party to maintain appropriate security. As many cautious companies have learned the hard way, a third-party issue or breach quickly becomes *your* problem and your incident. Third parties must adhere not just to best practices but also policies and regulations related to data protection, data retention, model training, and more. Third parties must be fully up to speed on such things as data residency requirements, audit needs and limitations, and breach notification terms.

Third-party API dependencies always bring risks related to availability and performance, of course. Outages or latency spikes can impact agentic AI workflows and potentially trigger retries and fallbacks, which add risks of their own.

Change risk is another third-party consideration because change on their part isn't within your scope of control — maybe not even fully within your knowledge. The consequence can be hurry-up rewrites of processes, or temporary workarounds, either of which is asking for trouble.

WARNING

And then there are the supply chain risks that third parties can bring — it's also true in manufacturing, retailing, and any business that needs a supply of something. A third party may quietly rely on a fourth or fifth party, and while your circle of trust includes the third party, what about the fourth or fifth?

Detecting Exposures

If you're wondering whether the rewards of agentic AI are worth the risks, fear not! I mention a few Hollywood movies, and like Hollywood, I much prefer a happy ending. The secret to writing that happy ending is really knowing your characters and anticipating all of the plot twists well in advance. That way you won't be startled when the killer pops up in the backseat.

In this case, I'm talking about such approaches as API risk assessment and the detection of sensitive data exposures (hopefully before anyone can exploit them). These are keys to catching dangerous behaviors before they spread.

REMEMBER

The first step in API risk assessment is truly knowing all of the APIs that you're relying on. Every last one of them. And the process of discovery means more than just consulting someone's list of APIs. Just because someone says this is the list of APIs, that doesn't mean that really is all of the APIs you have in your environment. You have to go looking.

If you're thinking "you can't protect what you don't know about" is certainly not new wisdom, you're right. Discovering everything you need to protect is a time-honored security concept, an oldie-but-goody from well before the days of agentic AI.

Discovering and inventorying APIs is the first step. Assessing and prioritizing them is essential, too. You need to understand the impact each API exerts and whether it connects with sensitive data. And you need to gauge the likelihood that it could be a problem, which includes noting what exactly it can do, and whether it employs authentication and authorization.

TIP

So think about these kinds of things:

- What's the criticality? Does this API access public data or protected information?
- Can it just read or can it also write or delete or export?
- What's the blast radius from a business perspective? Can it expose regulated data? Change prices? Move money?
- What's the reach? Internal-only? The Internet? Partners?
- What's the strength of authentication and authorization? Does it exist at all? Is it short-lived and purpose-bound, or more open-ended?
- What's the potential for over-permission issues?
- How volatile is it? Are there fast-moving versions and upgrades?

Runtime detection is how you spot dangerous behaviors quickly. You need to know the who, what, how, and why of proper API behavior. That way you can be instantly aware if you notice a wrong identity, a questionable intent, irregular behavior, or access to information or endpoints that are out of scope.

IN THIS CHAPTER

- Building compliance into agentic AI
- Understanding the risks related to regulations
- Ensuring agents are in compliance

Chapter 4
Securing Sensitive and Regulated Workflows

Agentic artificial intelligence (AI) workflows autonomously interact with data and systems, getting things done through orchestration layers that call application programming interfaces (APIs). This is all happening at the intersection of automation and regulation.

This chapter explores why compliance needs to always be at the center of that intersection, what kinds of regulatory risks may be triggered in agentic AI operations, and how to keep AI agents working on the right side of the law.

Keeping Compliance in Mind

So many fantastic things can happen with the help of AI. One California company developed a platform that can gather information about a specific individual and provide an output based on that info — in particular, an avatar that can even hold conversations. The hope is to uplift the emotional well-being of users through a virtual friend.

Italy didn't think the idea was entirely friendly, though. One particular concern was that an under-18 user from Italy accessed the service and entered personal information, which government officials said ran afoul of privacy regulations. The incident ended up costing the company more than $5 million in fines, and that was related to operations in just one country.

WARNING

AI operations, including agentic AI systems, have the potential to clash with a whole host of regulatory requirements and prohibitions. They can have trouble with rules specifically related to AI and the training of large language models (LLMs). But they can also run afoul of many other kinds of rules related to protecting the security of data as well as the privacy of individuals. In the Italy case, the complaint had less to do with AI and more to do with the age of the user in question.

Either way, when you think about it, the patchwork of regulatory regimes across geographies and industries is a real challenge for companies hoping to turn a profit. If you lose a few million dollars in just a single country, and have to either shut down access there or redevelop your product to live within the rules, what happens as you also try to do business in the country next door? What company can afford to navigate the minefields?

Just one more reason to have agentic AI that is fully up to the task. Agentic workflows intersect all the time with regulatory requirements, which means lots of careful consideration and real-time monitoring and enforcement. As the California company found by doing business with just one underage customer in Italy, it takes a single glitch to get into costly trouble.

REMEMBER

Real-time enforcement is always important, but particularly so when it comes to AI-powered operations.

- **Agents are acting at top speeds.** That's why you employ them, of course, but by the time compliance folks are able to review logs, some sort of misstep may have already happened, and a rule has been violated or sensitive data may have already found its way out the door.
- **Agents work hand-in-hand.** Dynamic API chaining creates workflows that are far bigger than just one tool, which can spread trouble far and wide. And shadow data flows may head in unanticipated directions and could create accidental regulatory violations.

- **Agents must be held accountable.** Regulatory regimes are all about accountability and auditability, meaning you need a trail of activity. Real-time enforcement offers proof that actions have always been compliant.
- **Agents need to be monitored.** Risk prevention is best, and risk containment is far better than cleaning up widespread messes. Real-time monitoring helps ensure that small variations don't escalate into big breaches.

Agentic AI systems aren't just sitting around thinking and analyzing — they're actually taking actions that may involve sensitive data, financial transactions, and decision-making that can have legal implications. They are likely to handle protected health information (PHI) or personally identifiable information (PII) or financial records.

REMEMBER

That means compliance needs to be a live operational requirement, happening all the time in real time. It's all about policies, but it's far more than just a policy document.

Outlining Regulatory Risks

Just as you need to be fully aware of all the APIs you're trying to watch and protect, you also need to be fully up-to-speed on the rules that you are expected to follow. When it comes to data privacy and protection regulations, perhaps the best known is the General Data Protection Regulation (GDPR) of the European Union (EU). It puts strict limits on how data is used, maintained, accounted for, and deleted within Europe and involving Europeans.

The most impactful similar laws in America are the California Consumer Privacy Act and the newer California Privacy Rights Act. All these regulatory regimes spell out individual rights that must be respected, including by AI agents.

Money makes the world go around, so no surprise there are lots of regulatory systems focused on financial matters. Examples include the rulemaking of the Payment Card Industry Security Standards Council, the Securities and Exchange Commission and the Financial Industry Regulatory Authority in America, MiFID II in the EU, and the Financial Conduct Authority in the United Kingdom.

Europe also has Digital Operational Resilience Act (DORA), which just took full effect in 2025 and deals with the digital resilience of financial institutions and their information technology providers. Wherever the jurisdiction, financial regulations tend to focus on accuracy and accountability of financial records and transactions, fraud prevention, and the safety and resilience of financial data.

As for health care regulations, probably the most feared in America is the Health Insurance Portability and Accountability Act (HIPAA). It's all about what can and can't be done with medical records and PHI. The key is protecting from unauthorized access or disclosure, but it's easy to be tripped up by HIPAA. There are plenty of other industry-specific regulations, so agentic AI may come under the microscope of those in certain cases.

REMEMBER

But especially burdensome and tricky are the regulatory regimes, including those outlined above, that create the need for region-aware data access. For example, various data localization laws mean that agentic AI can't just route data through APIs without considering what jurisdictions they are in or where the data will travel.

For example, data transfers that involve locations in China or Russia may run into regulatory issues. And not only does the GDPR focus on where geographically the activity is taking place, it also concerns itself with who is part of the transaction or activity. If a person in the EU is involved, the GDPR applies, even if all data resides and computing happens in America or somewhere else outside of the EU.

Keeping Agents in Line

Compliance is vital in every part of the organization, from financial considerations, to the way employees are hired and managed, to the way buildings are designed and maintained. Companies of all sizes have people in charge of compliance, often whole departments full of people.

And so, of course, agentic AI is no different from anything else. Compliance is both vital and complicated. And because of the nature of AI, compliance errors can multiply into disasters with remarkable speed. Autonomous agents can certainly raise new

concerns, but it also stands to reason that matters of compliance can benefit from some aspects of automation.

REMEMBER

A compliance rules engine is a powerful approach for getting a handle on compliance in the world of agentic AI. Consider that it is referred to as an "engine." This isn't just a rulebook or a scorecard — it's going places and doing things and driving compliance.

A compliance rules engine is a bit like a traffic controller. It handles policy enforcement duties in real time, keeping watch over API calls and data requests to be sure they're in line with both internal policies and external laws and regulations.

The rules engine has agentic powers of its own. It can make decisions on the fly, and take actions to protect systems and data and the organization as a whole. Its policy and rule enforcement can take a variety of contextual factors into account, including user roles, geographies, the systems involved, and the types of data. It can, for example, notice that PHI is being handled, and automatically restrict agent access unless the request comes from a covered entity that is acting in a proper role.

This kind of automated governance keeps the organization audit-ready while cutting the need to manually review every action. And it's vital because it helps keep pace with the speed of AI. After all, without proper controls, an agent could quickly and efficiently expose restricted patient records, or perhaps transfer the personal data of EU residents outside approved jurisdictions, or maybe execute financial transactions that go beyond regulatory limits.

This kind of action-oriented, automated compliance won't work well if data is not properly classified. Compliance rules can only be enforced if a compliance engine knows what every piece of data is — its type and its sensitivity, whether it includes PHI or PII or financial data, or perhaps export-controlled information.

Proper data classification also helps tremendously with auditability. When each agentic action is tied to classification metadata, it's far easier to prove that the data was handled properly.

And beyond all that, this approach boosts efficiency. Compliance work has always been necessary, but some may have viewed it as a necessary evil, perceiving it as a roadblock to getting things done as quickly and effectively as possible. With proper data

classification in place and a compliance engine driving enforcement, agents can get their work done quickly while steering clear of dangerous compliance gaps.

REMEMBER

It should go without saying that strong compliance enforcement is good business. Establishing guardrails in real time helps prevent violations before they ever happen, or mitigates them before they can get worse. That can avoid or reduce fines and reputational damage. And with verifiable logs in hand, auditors can see strong evidence of due diligence and control, which is exactly what they want to see.

IN THIS CHAPTER

- Building your adaptive defenses
- Monitoring the whole workflow
- Staying current on policies
- Plugging into security ops
- Putting together a unified solution

Chapter 5
Building Agent-Resilient API Architectures

This is the second-to-last chapter, and I confess that this book has been a bit of a horror story up to this point (if you've been reading in chapter order). You may have seen that Warning icon quite a few times because agentic artificial intelligence (AI) has the potential to take risk and crank it up to 11.

Spoiler alert: You're near the Hollywood ending. This chapter explains that you can have the benefits of agentic AI along with all the security you need. Read on to learn about building an adaptive defense for application programming interfaces (APIs), keeping an eye on the whole workflow and the latest threats and policies, joining forces with the rest of your security program, and building a powerful, unified solution.

Creating an Adaptive Defense

Elsewhere in the book, I reminisce about the long-ago days when a moat provided great perimeter security. Dig a wide ditch around the castle, fill it with water, then maybe throw in a few alligators. You're safe for the long haul — until your adversary learns to fly over the moat on a dragon, and then you had better adapt your defense.

REMEMBER

When it comes to securing agentic AI, you need an adaptive defense from the get-go. No telling whether they'll be coming at you tomorrow with dragons or missiles or teleportation — you have to be ready for anything, ready to adapt. Here are some strategies:

- **Getting the behavioral baseline:** For a while now, companies setting up login and authentication systems have employed CAPTCHA or similar controls to keep the bots out. Now, not only are there AI agents that are able to fool those systems, but also many companies are actually *welcoming* bots to come do agentic business. That means your security must be able to understand benign bot behavior in order to recognize when bad agents are knocking on the door. Behavior patterns can be spotted, and responses written into policies.
- **Tapping intelligence:** You're not in this alone. Everyone is dealing with the same kinds of villains, so there is helpful information out there through intelligence feeds and other sources. Behavioral analytics of your agentic AI system can watch for known issues and behaviors, spot compromised connectors, notice traffic coming in from internet protocol ranges known to be troublesome, flag the latest iffy prompts, and watch for other always-changing threats.
- **Employing automation:** Build a defense that delivers real-time risk scores based on company-specific parameters, industry norms, regulatory requirements, the pacing of logins, and the like. Set up dynamic policy enforcement, and tap into the power of good AI to optimize detection workflows.
- **Hardwiring resilience:** Find the blind spots by collecting information from sources such as your web app firewall (WAF) and your API gateway and layering it into your application layer. Use machine learning to detect and identify known traffic patterns, to get a better handle on such issues as credential stuffing.

Watching the Big Picture

Agentic AI is an orchestrated system, not just an individual API call but whole agent-enabled workflows. Securing agentic AI, thus, requires carefully monitoring those entire workflows, not just calls.

That means workflow-level visibility, being able to link a sequence of calls into a coherent narrative. Your security must be able to understand not just agent behavior, but agent intent.

For example, Chapter 3 envisions an agentic scenario able to plan and book a whole trip to see a show in Vegas, including air travel to get there, a hotel room, and dining reservations. All the systems involved need to recognize what's normal customer booking activities such as those, as opposed to a malicious bot going in and nabbing all of a restaurant's open reservations to sell on the black market.


REMEMBER

Agentic AI security best practices include session stitching, tracking tokens, watching device signals, employing geolocation, and tuning into time-based patterns to detect anomalous chains. Remember that agents are able to act more quickly and more broadly than humans, so observing their execution flow is critical for real-time validation.

Keeping Policy Current

As mentioned near the start of this chapter, things change. Like really quickly. Any agentic AI defense that was up-to-date as of a few minutes ago might be outdated a few minutes from now. True security requires dynamic and context-aware policy tuning, using threat signals and behavior models.

To begin with, static WAF rules just can't keep up with the constant evolution of agents. Everything is always changing, from prompts to payloads to query structures. Static rules can result in a lot of missed threats, with new attack forms slipping through. False positives can be a problem, too.

As an example of how agentic AI can evolve, imagine that a WAF rule has been established to block anything coming in faster than 100 requests a minute. Seems prudent enough. But now imagine that a bad bot figures that out, and learns how to slip through 96 requests a minute. Now you need a new rule. If your adversary is smart and dynamic, your defense had better be, too.

There are many ways your defenses can also evolve to keep up with the changing times. You need a system that is continuously noticing and flagging new agent behaviors. Behavioral models should

be able to learn as they go, with threat signals used to reprioritize enforcement actions. Aim to achieve policy tuning through low-code workflows or automated triggers based on detection confidence. Ultimately, you must keep detection capabilities current without always requiring additional rules.

Integrating With Security Ops

It certainly takes a village to keep an organization's information technology operations safe and secure. Your efforts aimed at securing agentic AI are just one piece of a much bigger picture, and it is essential that those efforts be in alignment with and in communication with that bigger security ops picture.


TIP

To take the Unified Application Protection (UAP) platform from Cequence Security as an example, API risks feed directly into the organization's platforms handling security information and event management (SIEM) and security orchestration, automation, and response (SOAR).

If there's a spike in failed login attempts across all of the endpoints, UAP will detect that and signal it into the SIEM as a correlated event from identity access management logs. The SOAR system should have auto triggers in place. The offending IP addresses get blocked, there can be crosschecks to see whether that IP is part of a known botnet feed, and the threat hunters can be called in. Many automated hands can thus make quick work of that security threat.

And as mentioned elsewhere, not only does agentic AI bring your enterprise some amazing new capabilities, it also supercharges the capabilities of those who would do harm. The powerful capabilities of agent chaining come to mind.

In one scary scenario, a malicious large language model (LLM) could use one API to enumerate account identification information, while a second automatically tries password attempts, and a third digs for personal information. You'll want to have all of the appropriate alerts integrated into SIEM and SOAR in order to effectively spot and deal with that.

Which brings us to threat detection, investigation, and response (TDIR). Ensuring effective TDIR at runtime is absolutely critical, especially for API channels that carry sensitive information.

TIP

Finally, application and API defense needs to be fully up to speed on the latest threats and attack concepts. The Open Web Application Security Project (OWASP) top ten list is certainly great information, but it's not enough. You'll want to move beyond that to embrace newer patterns such as agent chaining, API enumeration (when an attacker pokes around to discover an API's endpoints and resources and other parameters), and synthetic identities (a type of fraud that builds fake identities by stirring together real and fabricated data).

Keeping it Unified

The rest of this book has painted a picture of agentic AI as a growing powerhouse of possibilities as well as a complex landscape of connected applications and resources that make these powerful things happen. That, in turn, creates a broad and tempting attack surface to defend, requiring high visibility and an orchestra's worth of defensive approaches.

The prospect of agentic AI security may thus seem daunting, but the good news is a unified approach is both the most effective way to go and the most straightforward to implement and operate. You can turn insights into action at scale through UAP and threat detection and response — and it's achievable today.

REMEMBER

That's a big relief, because API activities have been going through the roof, just in the past couple of years. For many organizations, it seems clear that if you don't get on the agentic AI train right now, you'll be left behind. That fear of missing out in some cases leads to unhealthy haste and even a dangerous lack of carefulness. We're here to tell you, you really can safely board this accelerating train without throwing caution to the wind.

For agentic AI security, I talk about the must-haves in other chapters. As you seek tools to facilitate unified protection, here are things you want to bring together:

- **API discovery and inventory:** Find a tool that continuously discovers APIs that are internal and external, as well as part of third-party resources. You need a complete runtime API inventory of everything everywhere, including edge,

infrastructure, gateway, and hosting providers. Your API inventory can't be based on just a list someone made — it requires broad and continuous discovery, integrated with your infrastructure.

- **API threat detection and prevention:** Your system needs to always be on the hunt for threats, comparing your API landscape with the best available intelligence on known attack behaviors and potential concerns. Tap into machine-learning-powered threat detection that integrates with third-party defensive solutions such as WAFs and API gateways. And keep an eye on the network, not just apps, for the best visibility.
- **Bot defense:** Secure agentic AI means both defending your good bots and protecting against bad bots. This work takes many forms. It can include detecting and blocking AI bot activities such as scraping, preventing AI-enhanced business logic abuse, and protecting against such threats as credential stuffing. Your defenses should be able to spot API coding errors that could lead to data loss or other compromise. A key part of bot defense is being able to discern the difference between legitimate and malicious automation.
- **Runtime API posture management:** Security must be a primary consideration at all points of the life cycle, and that includes runtime posture management. Continuous checks can, for example, reveal sensitive fields containing protected data that shouldn't be part of the conversation. You need to know which APIs may be exposing that data, and be able to automatically mask the data.
- **Compliance and AI governance integrations:** After you have a comprehensive view of the API attack surface and the various risks you're trying to protect against, you can also verify conformance to specifications and compliance with all governance considerations. That includes internal governance as well as the kinds of external regulatory requirements we discussed in Chapter 4. Some are specific to AI, but even the many that aren't AI-specific have very serious implications in the world of agentic AI.

IN THIS CHAPTER

- Discovering, monitoring, and acting
- Involving the humans
- Protecting and defending APIs

Chapter 6
Ten Tips to Secure Agentic AI

The previous chapters go into depth about the powers and perils of agentic artificial intelligence (AI) and the vulnerabilities introduced by the way it takes action through application programming interfaces (APIs). In this chapter, you discover the ten essential steps to secure your agents.

Discovering and Inventorying

You can't secure what you can't see. What you need is total visibility into all agents that exist in your ecosystem. That means more than just seeing them — it's vital to fully understand what APIs they call, as well as what permissions they hold. It means not taking the word of somebody's notes taken on a whiteboard or memorialized in a spreadsheet. It means active ongoing discovery, because the landscape is likely bigger than you think, and it's always changing.

This if, of course, not just knowledge for the sake of knowledge. The community will mature much faster and become more secure when you broadly share discovery frameworks and lessons learned.

Treating APIs as the Frontline

AI agents aren't just sitting there, watching and thinking and creating. They're doing things, taking real, concrete actions. Agents live and breathe APIs, because APIs are their means for connecting with the apps involved in taking those actions.

APIs are how agents do all the good they do, but they can also open the door to doing the wrong things, either accidentally or by allowing malicious actors to gain powers they're not supposed to have. So, protect all APIs that you discovered in that last step with authentication, encryption, input validation, and rate limiting.

Lessons learned should also be lessons shared. With that in mind, report patterns of API abuse back to peers so defenses evolve collectively.

Preventing Prompt Injection

Prompt injection is a con, tricking agents into taking actions beyond or counter to their intended scope. Like the mind control depicted in so many suspense films (and comedies, too), prompt injection delivers instructions that steer agents off the right path.


TIP

It's essential to sanitize and filter inputs so you can find and block hidden instructions that would otherwise hijack agent behavior. Share your findings by documenting new injection tactics and contributing to shared repositories of defenses.

Enforcing Least Privilege

No one — human or agent — should be allowed any privileges beyond what they need in order to do what they're supposed to do. It also must be possible to immediately revoke access to agents that are misbehaving.

Your team must scope tokens and keys tightly, giving agents the bare minimum of access, and ensuring that revocation is instant. The best practices when it comes to identity and privilege are no secret with regard to humans. So, advocate for the same best practices around agent identity and privilege.

Keeping Humans in the Loop

This is a matter of both visibility and control. If you don't have insights into what's going on, you won't know what's going wrong. Beyond that, as powerful as agents can be, there are some decisions that really need to be left to the humans.

TIP

For sensitive workflows — such as financial moves, data deletion, and compliance actions — always insert human checkpoints. That keeps humans aware of the more impactful actions about to be taken, and allows them to give either a thumbs up or a thumbs down. This should be the norm across industries, both to strengthen security and also to ensure that agentic AI remains trusted.

Isolating and Securing Multi-Agent Ecosystems

The more agents you have as part of your agentic AI workflow, the more you can get done. But the more complex it gets, the more risks you've created.

That's why it's vital to sandbox agents, to isolate them from the potential damage they might do before you fully understand their behavior. It's why you must monitor inter-agent communications, and also prevent lateral movement. And to put in yet another plug for sharing the wisdom, don't forget to publish reference architectures so others can design safer orchestrations.

Validating Outputs and Mitigating Hallucinations

By now it is no secret that AI doesn't always get things right. What could be worse than acting upon mistaken understandings?

REMEMBER

For a truly safe and secure agentic AI system, you must require evidence or provenance checks for agent conclusions. If there are hallucinations, you need to spot them before an agent acts on them. Work with the research community to improve shared methods for testing accuracy and truthfulness of AI outputs.

Red Teaming, Monitoring, and Disclosing Responsibly

Security should never be just a thought exercise. It requires continuous prudence and action, and the ability to see things from the viewpoint of the bad guys. Ongoing red teaming is a part of that effort, checking out the agentic AI workflow as a simulated adversary and looking for weaknesses.


TIP

As part of your monitoring, you must continuously test for prompt injection, data poisoning, and model evasion. Don't forget that everyone gains when wisdom is shared — whatever you learn about threats and risks, share it responsibly with the wider community to harden defenses across the ecosystem.

Defending Against Supply-Chain and Connector Risk

Agentic AI's ability to reach across a broader supply chain of possibilities is a both huge strength and major vulnerability. It means that someone else's problem can quickly become yours.

With that in mind, carefully vet external APIs, models, and plugins before integration. Support all of the efforts that can help reduce hidden risks of working hand-in-hand with outsiders, including community software bill of materials standards, model cards, and transparency initiatives.

Planning Governance and Actions

Proper governance outlines what is intended, what is required, and what needs to happen if things get out of line. It's essential to define policies and escalation paths. And just as your building probably displays maps of emergency exit routes, you must fully think through in advance the means for emergency shutdowns in case of rogue behavior.

As with all of the steps above, working as a community is essential. Participate in shaping open standards such as Model Context Protocol (MCP) so access to data isn't dictated only by the largest platforms.

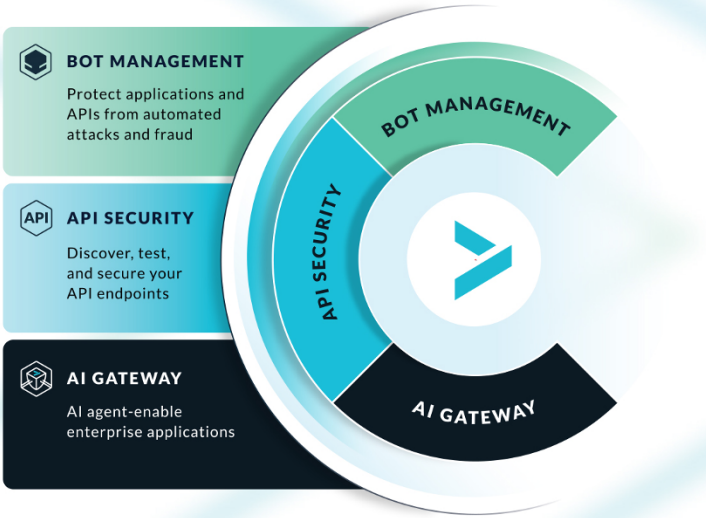
CEQUENCE

Application, API, & AI Security

Protect What Connects You.

Cequence protects the applications and APIs that organizations depend on from attacks, business logic abuse, and fraud, while unlocking the promise of agentic AI productivity.

Visit **www.cequence.ai** to learn more.



Gain visibility into all APIs with agentic AI

This book is your introduction to operating in the new world of agentic AI. You discover how agentic AI functions, how it relies on APIs, how it makes connections to apps and third-party resources and important data, and why it's vital to secure all APIs. You're made aware of the threats that prey on agentic AI, the potential consequences, compliance considerations, and detailed steps to take to build secure agentic AI with agent-resilient API architectures.

Inside...

- Stand against invisible intruders
- Protect the API supply chain
- Secure sensitive and regulated workflows
- Build agent-resilient API architectures
- Ten tips to secure agentic AI

Steve Kaelble is the author of many books in the *For Dummies* series, and his writing has also been published in magazines, newspapers, and corporate annual reports. When not immersed in the *For Dummies* world or writing articles, he engages in healthcare communications.

ISBN: 978-1-394-37963-7
Not For Resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.