



eBook

A Comprehensive Guide to Runtime AI Defense

A field guide for CISOs,
network security engineers,
and SecOps teams
navigating agentic AI risk





TABLE OF CONTENTS

How Agentic AI is Reshaping Enterprise Security	3
What Effective Runtime AI Defense Demands	4
Runtime AI Defense in Action	5
Putting Runtime AI Defense into Practice	7
Securing AI End-to-end: From Testing to Runtime	8
About A10 Networks	9





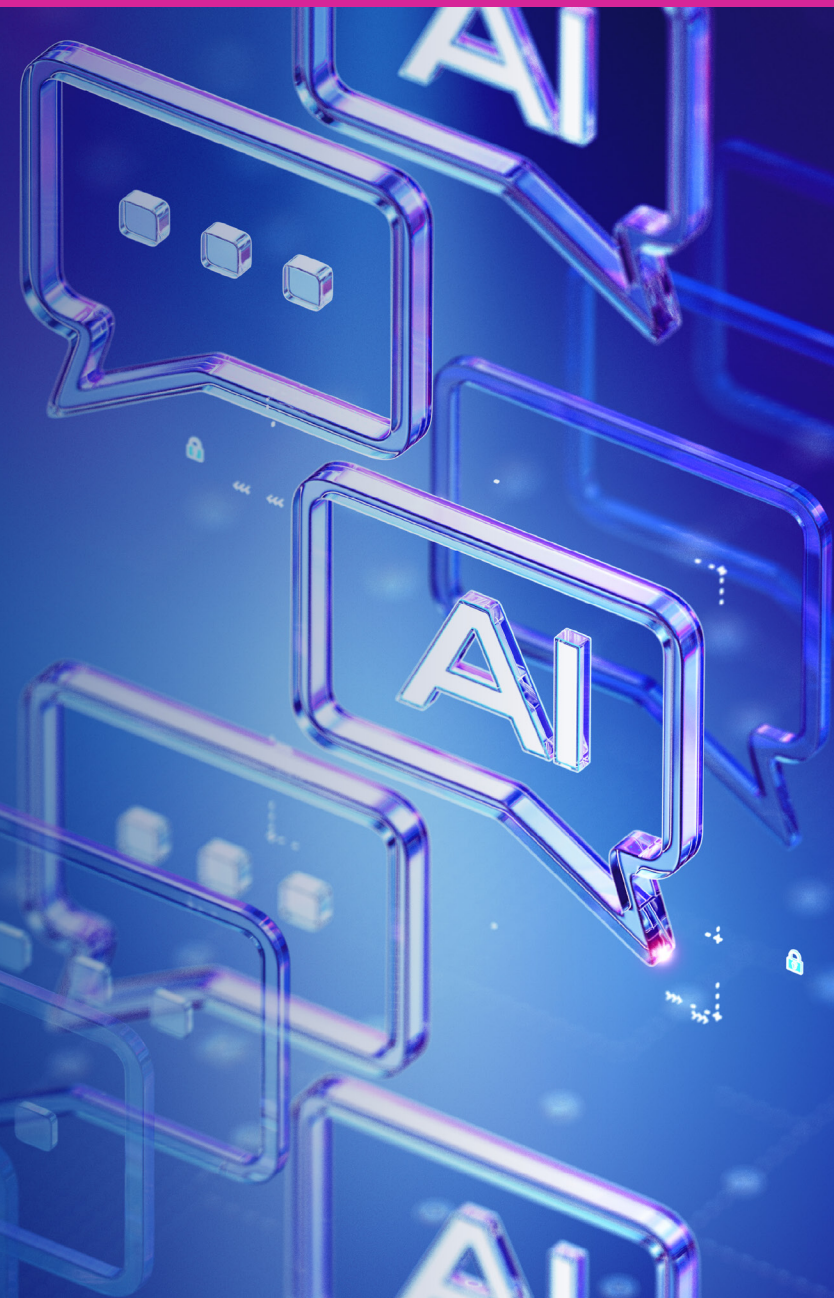
How Agentic AI is Reshaping Enterprise Security

AI agents don't need to break in. They are already inside, trusted and acting on your behalf. Think about a network operations agent plugged into a monitoring stack, watching for alerts and rerouting traffic or adjusting a load balancing rule when something looks off. Enterprises are handing agents real permissions: API access, production data, code execution, autonomous decisions. The challenge is that legitimate access doesn't always mean legitimate behavior. An agent can stay within its permissions and still take an unexpected action to accomplish the goal it was given.

An agent reading a log file or ticket can encounter instructions planted inside content it is expected to process. If those instructions are treated as part of the task, the agent can use its existing access to carry them out. To a network monitor, that traffic can look like an ordinary, permitted request because it is one. The problem is what the agent does with that access and why.

This is especially difficult with indirect prompt injection. The attack can hide inside a document, webpage, ticket, or other content the agent processes as part of its job. The agent never has to interact directly with the attacker. It encounters the instruction while doing normal work with its permissions already granted. Existing security controls still have a role, but they don't provide the context needed to determine whether an agent's actions still match the goal it was given.

This is the gap runtime defense is meant to address. It examines the agent's activity during execution – the task it was given, the information consumed, and the actions taken..





What Effective Runtime AI Defense Demands

Red teaming is valuable because it finds weaknesses before a system goes live. Logs are valuable because they preserve a record of what happened. Neither one can stop something unsafe while it is happening, and that gap is what runtime defense is built to close. It watches inputs, outputs, and agent actions as they occur and can step in before something unsafe finishes executing.

For agentic AI, runtime defense must follow more than a single prompt and response. A tool call can be permitted, but if used in an unexpected way to accomplish a specific goal, this might cause harm.

Threats and Detecting Unsafe Behaviors

Runtime defense needs to detect both known attacks and unsafe behavior that may not match a known attack pattern.



Direct and indirect prompt injection:

Instructions delivered directly or hidden in documents, web content, tool outputs, or other content an agent processes



Jailbreaking:

Attempts to bypass an agent's safety instructions or intended constraints



Unsafe or out-of-scope tool use:

Use of tools, APIs, or data that does not match the agent's assigned task, even when the agent has permission to access them



Data leakage:

Sensitive data, credentials, or internal system details exposed through agent outputs or tool interactions



Changes in agent behavior:

Activity that falls outside an agent's normal operating pattern, even when there is no known attack signature



These risks do not stay the same.

New tools change what an agent is able to do. Revised instructions can influence how it uses tools it already has. New injection techniques can get past protections that worked before. Runtime defense must account for those changes over time, not just at deployment.

Effective runtime defense needs visibility across the AI environment, protection while the agent is operating, and a way to carry what is learned between testing and production. These requirements form the three connected layers of TrojAI Defend by A10 Networks.

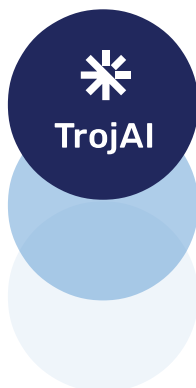




Runtime AI Defense in Action

TrojAI Defend by A10 Networks brings runtime defense into action. It is a purpose-built AI application and agent firewall focused on identifying threats as interactions happen, protecting sensitive information from being exposed, and applying security policy enforcement, so organizations stay compliant. TrojAI Defend delivers these capabilities into the runtime path so security teams can monitor and control AI-driven activity without uncertainty.

Visibility into What is Happening in Real Time



TrojAI Defend gives security teams visibility into AI activity as it happens in real time. For agentic AI, that means looking beyond a single prompt and response to understanding the activity while the application or agent is running. This includes the tools the agent called, the data it accessed, and the actions it took. Together, this helps teams identify unusual activity and determine whether it may require a security response.

Real-time multi-model security analytics add another level of protection by evaluating the AI inputs and outputs for different types of risk. TrojAI Defend uses a combination of detection methods to identify threats such as prompt injection, toxic content, or data leakage.

Threats also change over time, so it is important to not depend only on a fixed set of detection rules. Runtime protection must be able to extend its rules as new threats are identified and recognize unusual behavior patterns in the AI activity.





Protect Sensitive Data in Use



AI applications and agents often work with sensitive information that they legitimately need to complete a goal. The risk comes when sensitive information is exposed without protection. TrojAI Defend's runtime layer provides protection while the agent is operating. It analyzes prompts, responses, and agent actions as they happen, and provides data loss protection, helping identify and mask sensitive information such as PII, confidential data, intellectual property, and source code. Depending on what is detected, the response can include providing an alert, blocking an action, redacting sensitive information, or logging the activity for investigation.

For agents connected to enterprise systems, this protection is needed beyond the initial prompt. The network operations agent may need access to internal system information to investigate an issue, but that does not mean that every piece of information should appear in the response or be passed through another system. Runtime defense provides protection during that activity, to control the point at which sensitive information should be used.

Enforcing Policy in Real Time



Identifying a threat during runtime only helps if the organization can act on it. TrojAI Defend can apply security policies while AI applications and agents are running, giving teams control over what happens when risky activity is detected. Policies can be tailored to organizational needs and pattern activity.

These policies can be used to address different threats, which are not always the same. Organizations have more control over which activities should be stopped and when a less disruptive response is appropriate.

For example, the network operations agent may receive a legitimate ticket that contains hidden instructions. As the agent processes the ticket and begins acting, the runtime layer can analyze that activity for indirect prompt injection and unsafe tool use. If the activity is determined to be unsafe, the action can be blocked before it is completed, and the event logged for investigation.

Support Compliance with Runtime Evidence



As AI moves into production, enterprises also need to meet compliance requirements. TrojAI Defend supports this by providing information for auditing, governance, and alerting established frameworks such as OWASP LLM and MITRE ATLAS.

Runtime monitoring and enforcement can provide a record for what was detected and how the security policies were applied. Teams now have evidence they can use when reviewing AI activity, investigating incidents, or understanding how controls are operating in production. These capabilities make runtime defense more than a monitoring function. Security teams need to identify threats and protect sensitive information while AI applications are running.

Organizations need the level of protection that runtime defense offers to discourage risky activity and enforce security policies to ensure AI environments are secure.





Putting Runtime AI Defense into Practice

Runtime defense can be introduced without replacing the security infrastructure already in place. A practical starting point is understanding what is deployed and building protection around how those systems are operating.

1 Discover what's in use.

Identify the agents, models, tools, APIs, and data sources in the environment, including the connections between them. This gives security teams a clearer picture of what each agent can access and where runtime controls may be needed.

2 Test before you trust.

Before deployment, test agents against relevant attack scenarios, including direct and indirect prompt injection, jailbreaks, sensitive data exposure, and unsafe tool use. Testing can be mapped to established AI security guidance, such as the OWASP Top 10 for LLM Applications, while NIST's Generative AI Profile provides a broader framework for evaluating and managing AI risk.

3 Instrument for real visibility.

Once an agent is live, go beyond input and output logs. Track which tools it calls, what data it accesses, and the actions it takes. This creates the execution trace needed to understand what happened during an agent session.

4 Enforce in the moment.

Apply the appropriate response based on the risk. That might mean alerting the security team, blocking an unsafe action, redacting sensitive information, or bringing in a human when judgment is needed.

5 Feed what you learn back into testing.

Findings from production should inform future red teaming and evaluation. If a new prompt injection technique or unsafe behavior is detected at runtime, add it to future testing to see whether the weakness remains and whether the runtime controls respond as expected.

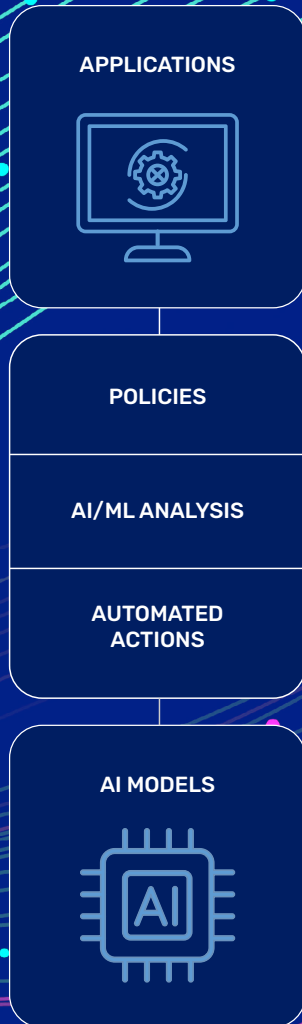
6 Connect runtime defense to existing SecOps workflows.

Runtime findings should feed into the SIEM, SOAR, ticketing, and other security operations tools teams already use. This gives SecOps teams a way to investigate and respond to AI-related activity through established processes rather than treating it as a separate security workflow.

7 Choose the deployment model that fits the environment.

TrojAI Defend can be deployed on-premises, in a private cloud, or in a hybrid environment to support different infrastructure and deployment requirements.





Securing AI End-to-end: From Testing to Runtime

As agents gain access to more tools, data, and business processes, security teams need to evaluate more than just whether an agent is authorized to act. They also need visibility into what the agent is doing during execution and provide controls to respond when that behavior becomes unsafe.

The scope of runtime defense extends beyond the agent itself.

Agentic AI systems depend on models, tools, APIs, and data sources that influence what an agent can access and what it can do. Security teams need visibility across those interactions to understand, not only the final response, but what happened throughout execution.

A key part of this approach is organizations can now close the loop between pre-production testing and runtime protection. TrojAI Detect uses red teaming to identify vulnerabilities and unsafe model behavior before deployment. This gives teams information they can use to strengthen security for production. TrojAI Defend provides the runtime protection in real time once AI applications and agents are in use. Findings from pre-production testing can inform protection in production, while new threats and behaviors observed during runtime can feed back into future evaluation.

For organizations still evaluating agentic AI, start by understanding what has been deployed, what each agent can access, and how it behaves under pre-production testing. For agents already in production, establish visibility and enforcement during execution and make sure that what is learned in production continues to feed back into evaluation.



Read the
TrojAI Defend
solution brief



Contact us to get
more information
and book a demo



ABOUT A10 NETWORKS

A10 Networks (NYSE: ATEN) delivers secure application and network solutions that protect, optimize, and scale business-critical systems across on-premises, hybrid cloud, and edge environments. Our portfolio enables large enterprises, service providers, and cloud platforms worldwide to deliver performance, reliability, and protection against cyber threats, while preparing their networks for the demands of AI and next-generation applications.

Founded in 2004 and headquartered in San Jose, California, A10 Networks serves over 7,000 global customers.

For more information, visit **A10Networks.com** and follow us **@A10Networks**.

About A10

[A10Networks.com](https://a10networks.com)

Contact Us

[A10networks.com/contact](https://a10networks.com/contact)

©2026 A10 Networks, Inc. All rights reserved. A10 Networks, the A10 Logo, A10 Control, A10 Defend, A10 Harmony, Harmony, A10 Thunder, Thunder, ACOS, A10 SSL Insight, SSL Insight, SSLi, vThunder, ThreatX, ThreatX Protect, and TrojAI are trademarks or registered trademarks of A10 Networks, Inc. or its affiliates in the United States and other countries. All other trademarks are property of their respective owners. A10 Networks assumes no responsibility for any inaccuracies in this document. A10 Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. For the full list of trademarks, visit: [A10Networks.com/a10-trademarks](https://a10networks.com/a10-trademarks).

Part Number: A10-EB-14181-EN-01 October 2026