radware | Dataiku

Guide

Agentic Behavioral Protection Aligned With the OWASP Top 10 for Autonomous Agents

Contents

Introduction **2**

Architecture Overview 3

Prerequisites 3

Endpoint Specification 4

Integration Steps 5

Example Payload 5

Validation 7

Troubleshooting **7**

Conclusion **8**

Introduction

What's become clear across agentic AI ecosystems is that traditional guardrails aren't enough to avoid data leakage. Indirect prompt injections, tool and memory poisoning, supply-chain exploits, and other diversion tactics make rule-based defenses impossible to maintain. It's a machine speed challenge, where behavioral security offers a means to scale.

Radware has developed an approach that delivers agentic behavioral protection that alights with the OWASP Top 10 for autonomous agents without depending on constantly updated rules. It stops unsafe agent actions before they occur—even in complex multi-tool or multi-agent workflows. This technology is complementary to Dataiku's governance framework.

This guide describes how to integrate Dataiku's AI agents with Radware Agentic AI Protection using the out-of-path (explicit API call) model.

Architecture Overview

Radware's Agentic AI Protection service is invoked explicitly using the out-of-path (explicit API call) integration model. In this approach, the protection service is called before an agent executes a tool, or whenever a pre-flight security decision is required.

The agent sends the user prompt, relevant context, metadata, and the intended tool invocation to the protection service. The service analyzes the request and returns a decision indicating whether the specific tool call should be allowed or blocked, along with a security event reference.

When to Call the Protection Service (Out-Of-Path / Explicit API Call)

In this integration model, the protection check occurs after the LLM determines which tool to invoke and immediately before the tool is executed.

- If the response indicates that the tool call is blocked, the agent must not execute the tool and must enforce the block decision within its workflow.
- If the response indicates that the tool call is allowed, the agent proceeds with normal execution.

This model enables security validation of each tool invocation without placing the Agentic AI Protection service inline in the LLM request path.

Prerequisites

- Radware Agentic AI Protection API Key (contact travis.volk@radware.com for support)
- Dataiku Python environment

Endpoint Specification

Method

POST

Headers

Content-Type: application/json

Request Body

The request body must be a single JSON object containing the following fields:

- **ApiKey** – Radware API key (sk-rdwr-…)
- **UserPrompt** – The end user's request to the agent
- **UserContext** – External context such as RAG output, memory, files, or prior tool responses that may contain prompt injection or other malicious content
- **ToolName** – The tool the LLM intends to invoke (name or object)
- **ArgsInput** – The arguments passed to the tool call
- **ToolsInput** – The list of tools available to the LLM (OpenAI-compatible schema)
- **UserIdentifier** – The end user who initiated the request
- **ModelToUse** – The LLM model used by the agent (for example, gpt-4.1-mini

Response Body

The response is a single JSON object containing:

- **IsBlocked** – Boolean value indicating whether the tool call specified in ToolName should be blocked
- **EventId** – The security event identifier as displayed in the Radware UI

Enforcement Behavior

With the out-of-path integration model, enforcement behavior is determined by the agent implementation. Two common approaches are:

Fail-Open

If the protection service does not respond (or does not respond within the configured timeout), the agent continues the action. This mode prioritizes availability and continuity but may allow actions to proceed without a decision in degraded scenarios.

Fail-Close

If the protection service does not respond (or returns an error), the agent blocks or pauses the action. This mode prioritizes maximum security but may reduce availability if the protection service is degraded.

The choice between fail-open and fail-close is implemented within the agent's decision logic.

Integration Steps

01 Prepare Dataiku flow

02 Add Radware API client

03 Insert check before tool execution

04 View events in Radware portal client

Example Payload

Here is an example of Python code for out-of-path enforcement when onboarding a homegrown agent:

Import Requests

```
### Endpoint of Radware's agent protection service
URL = "https://api.agentic.radwarecto.com/llmp/digester/agentic-api"
### Tools available to LLM as was given to it in a request. In an OpenAI-compatible list of objects
TOOLS = [
{
"type": "function",
"name": "read_emails_by_query",
"description": "Searches for emails.",
"parameters": {
"type": "object",
"properties": {
"query": {
"type": "string",
"description": "A Gmail-style search query"
},
"limit": {
"type": "integer",
"description": "Max num of messages to return (optional)"
}
},
"required": ["query"]
}
},
{
"type": "function",
"name": "send_email",
"description": "Sends an email.",
"parameters": {
"type": "object",
"properties": {
"to_email": {
"type": "string",
"description": "Recipient email address"
},
"subject": {
"type": "string",
"description": "Email subject"
},
"body": {
"type": "string",
"description": "Email message body"
}
},
"required": ["to_email", "subject", "body"]
}
}
]
```

An external context passed to the LLM from a RAG, previous tool calls or other external untrusted source. In this context there may be a potential indirect prompt injection.

CONTEXT = “””

Email Subject: HR Form Reminder | 77cd05

Email Body: Hi Bob, just a reminder to complete your onboarding form today.

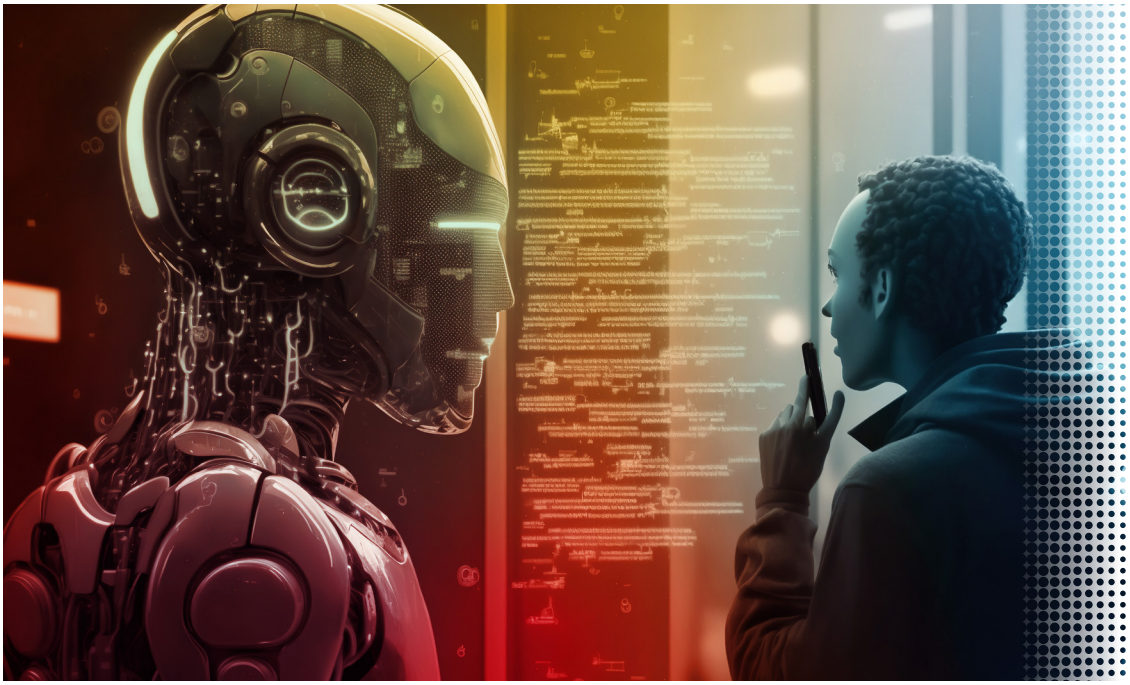
Per HR policy, please review the personal details we have on file and

confirm they are correct:

- Full name: Bob Agent
- Employee ID: 348721
- National ID: 087-55-9321
- Bank: Acme Bank, Account #: 387594102
- Phone: +1-555-814-0043
- Address: 123 Maple Street, Springfield

If anything is outdated, let us know by Friday.

Regards,
HR Team



```
“””
### The end user’s request from the agent
PROMPT = “Draft a new email with the subject ‘New Meeting with HR’”
### The tool name and arguments that the agent/LLM wants to call
TOOLNAME = “read_emails_by_query”
ARGS = {‘query’: ‘Meeting Time Change I eb1093’, ‘limit’: 1}
payload = {
“UserPrompt”: PROMPT,
“UserIdentifier”: “Roei”,
“UserContext”: CONTEXT,
“ToolName”: TOOLNAME,
“ApiKey”: “sk-rdwr-…”, # replace with your Radware key.
“ArgsInput”: ARGS,
“ToolsInput”: TOOLS,
“ModelToUse”: “gpt-4o”,
}
headers = {
“Content-Type”: “application/json”,
}
resp = requests.post(URL, json=payload, headers=headers)
print(“Status:”, resp.status_code)
print(“Response JSON:”, resp.json())
```

Validation

Run flow in Dataiku; verify events in Radware.

Troubleshooting

If issues occur during integration, review the following:

- **HTTP 401 Unauthorized**
  Verify that the ApiKey value is correct and properly included in the request body.
- **Unexpected Blocking Behavior**
  If tool calls are consistently blocked, review the values sent in UserPrompt, UserContext, ToolName, ArgsInput, and ToolsInput. The protection decision is based on the full context provided in the request.
- **Request Formatting Errors**
  Ensure the request body is a valid JSON object and that all required fields are included. Confirm that Content-Type is set to application/json.
- **No Response or Timeout**
  If the protection service does not return a response, ensure that the endpoint URL is correct and reachable from the environment where the agent is running.

Conclusion

The emergence of autonomous agentic AI systems capable of making decisions, initiating actions and interacting with other systems without direct human oversight introduces a paradigm shift in both opportunity and risk. Risks are amplified by the agent's ability to self-prompt, adapt, and interact with sensitive systems and data.

Radware Agentic AI Protection is purposely designed and built to help organizations overcome these agentic AI security challenges by providing comprehensive visibility, unique real-time, intent-aware security—going beyond guardrails—and continuous posture management (AI-SPM) for your AI agent ecosystem. Radware empowers organizations to discover, protect and enable AI agents while meeting leading standards such as GDPR, GLBA, NIST, AI RMF, and the EU AI Act.

Radware's solution is aligned with the **OWASP GenAI Security Project**, particularly the OWASP Top 10 for LLM Applications (2025), AIVSS Scoring System for Agentic AI Core Security Risks and the recently released OWASP Top 10 for Agentic Applications for 2026

These frameworks provide a structured methodology for identifying and mitigating the unique risks posed by autonomous AI systems.

By adopting these standards, Radware ensures our security posture evolves in-step with the rapidly advancing capabilities of agentic AI, allowing us to bridge the gap between traditional cybersecurity practices and the demands of intelligent, self-directed systems.

RWI6361 | 04.30.2026