SECURITY SUMMIT

Security Summit

Milano 17-18-19 marzo 2026

Clusit

ASTREA

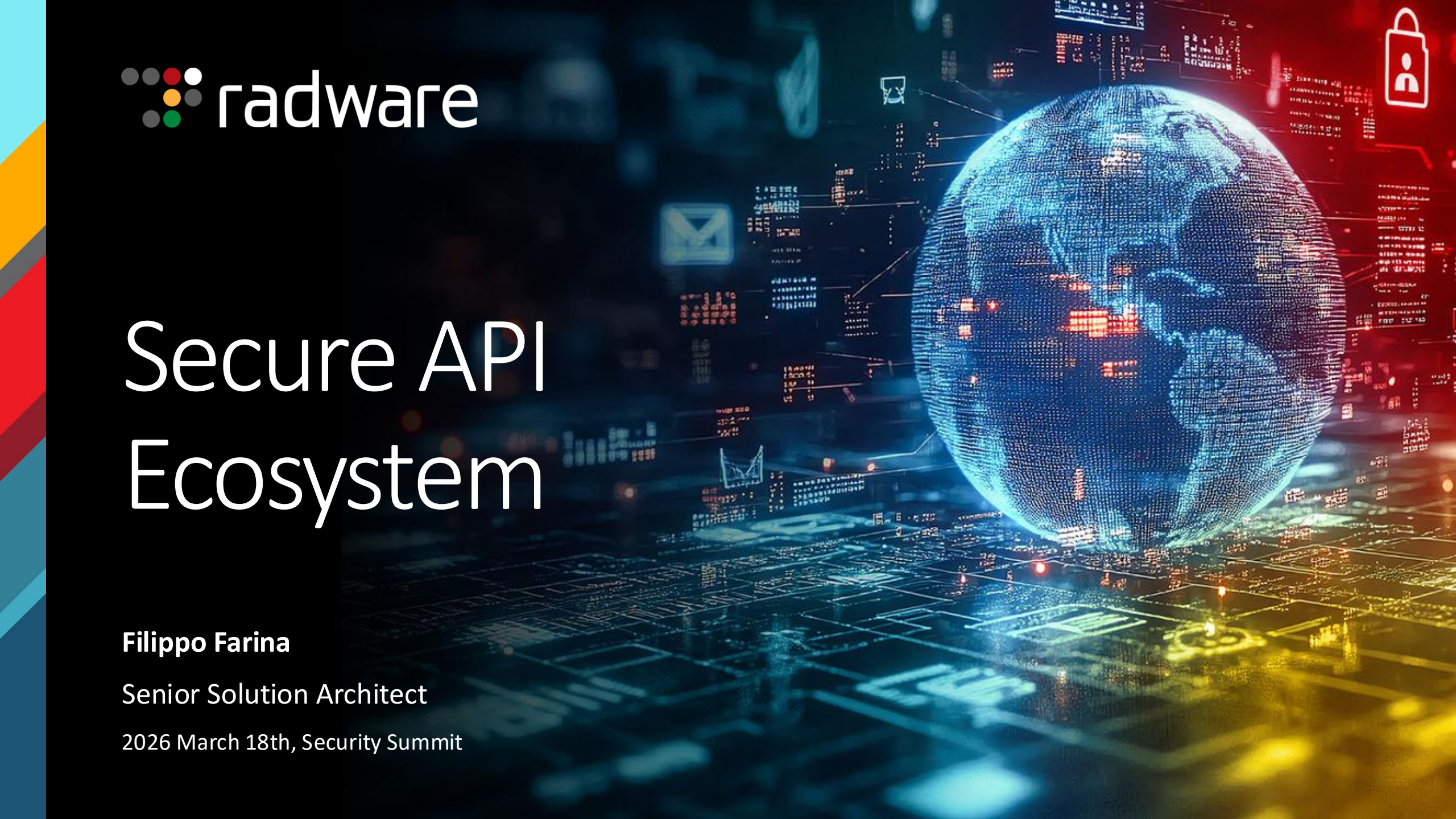
Sessione

Proteggere l’ecosistema API – AI: la nuova frontiera delle applicazioni moderne

Filippo Farina| Radware

radware

Secure API Ecosystem

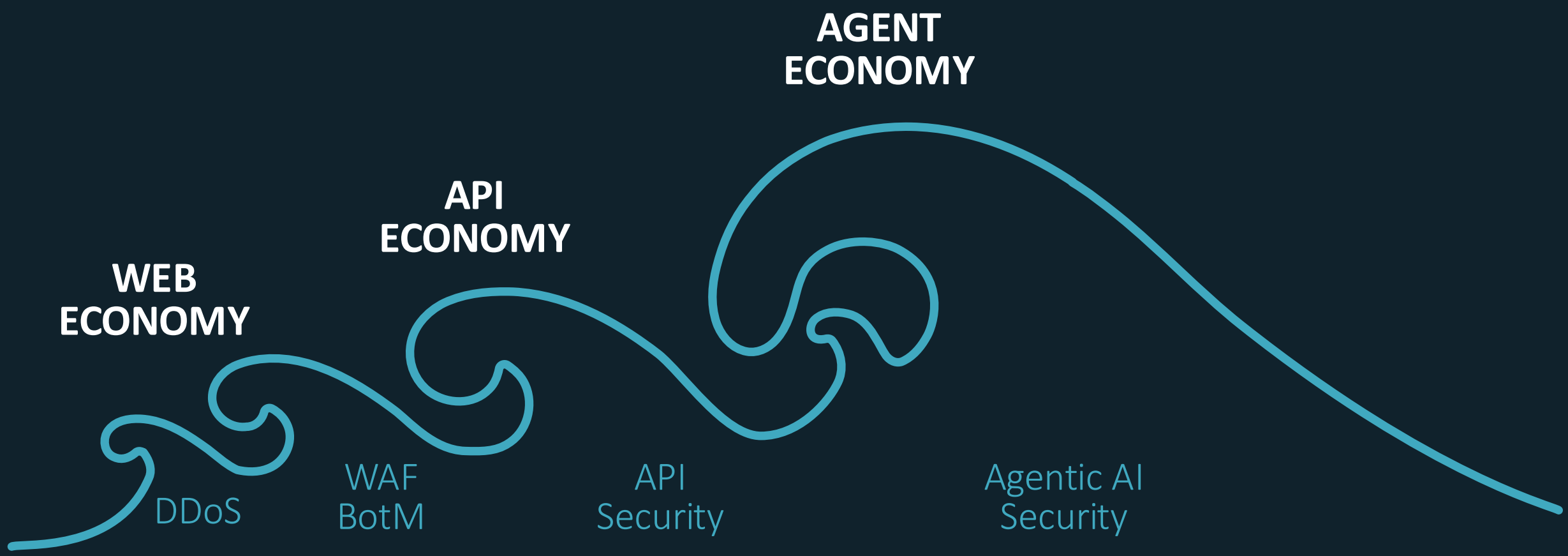


Filippo Farina

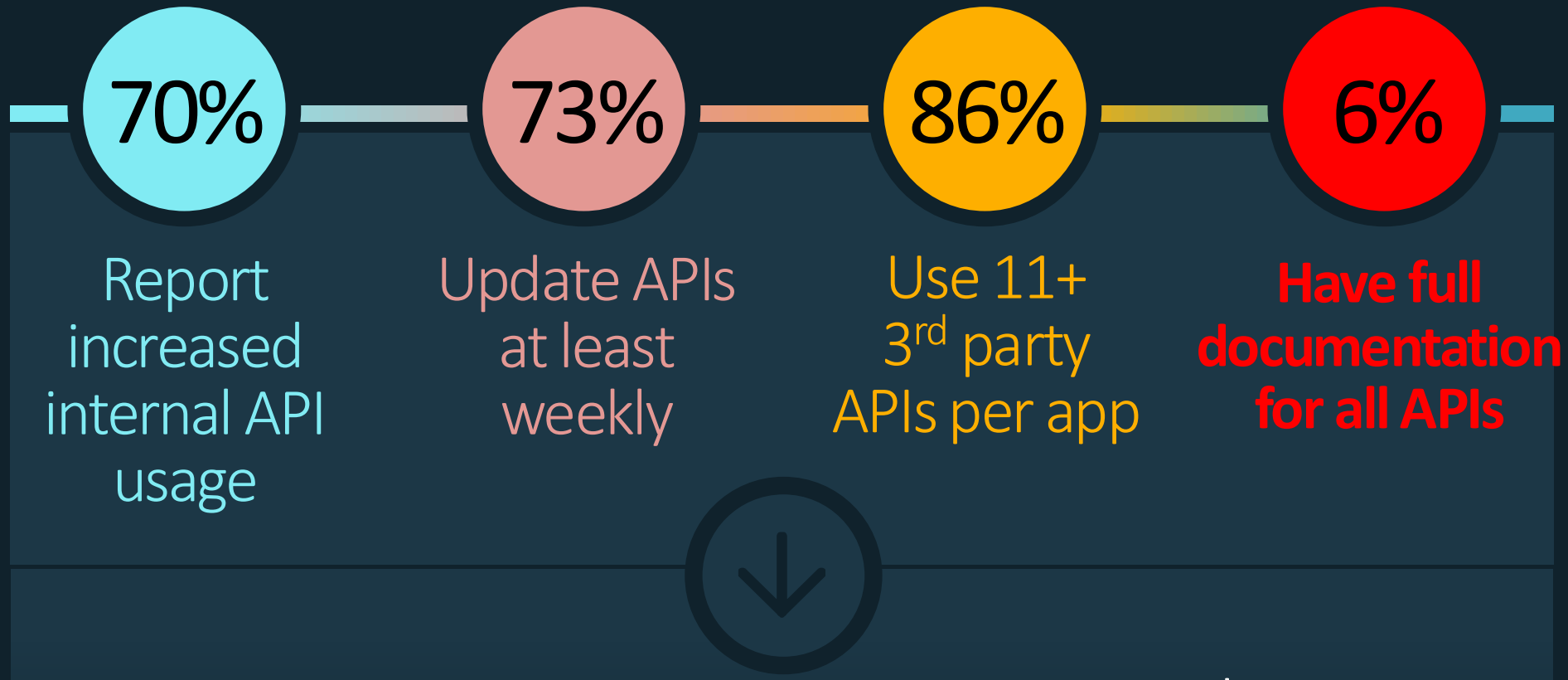
Senior Solution Architect

2026 March 18th, Security Summit

Market Evolution: Solutions Need to Adapt



API Growth Outpaces Security Controls



Frequent API changes & expanding 3rd-party integrations increase exposure to attacks, risking business continuity

Source: Radware 2025 Cyber Survey: Application Security at a Breaking Point, Jun. 2025

Radware Research: APIs & BLA Top Target for Attackers

84%

of the credential stuffing scripts explicitly target API endpoints

26%

of API-targeted configurations specifically target older version APIs

94%

Implemented advanced business logic attacks elements

- Map and monitor all your API authentication endpoints, or hackers will do it for you
- Advance BLA detection is needed: Threat actors using your apps in ways your developers did not expect

Source: Radware's The Invisible Breach Report, Aug. 2025

Challenges in Maintaining API Security

Visibility

- Keeping up-to-date documentation of all APIs including 3rd party & shadow APIs
- Mapping of the application business logic.

AI-Based Attacks

- Sophisticated hackers, map applications APIs, and their business logic
- API attack traffic that looks completely legitimate

Automation (Lack of)

- Many organizations rely on manual configurations and maintenance of API documentation and security policies

Runtime Protection

- Dependency on non-real-time out-of-path posture management tools designed for DevOp teams
- No visibility or real-time mitigation capabilities for Security teams

Multi-layered Protection

- Missing protections
- Silos between security solutions
- No attack story view
- No correlation between different security solutions

API Attacks

1 Embedded Attacks

- SQL Injections
- Authentication bypass
- Exploiting outdated libraries
- Exploiting insecure endpoints

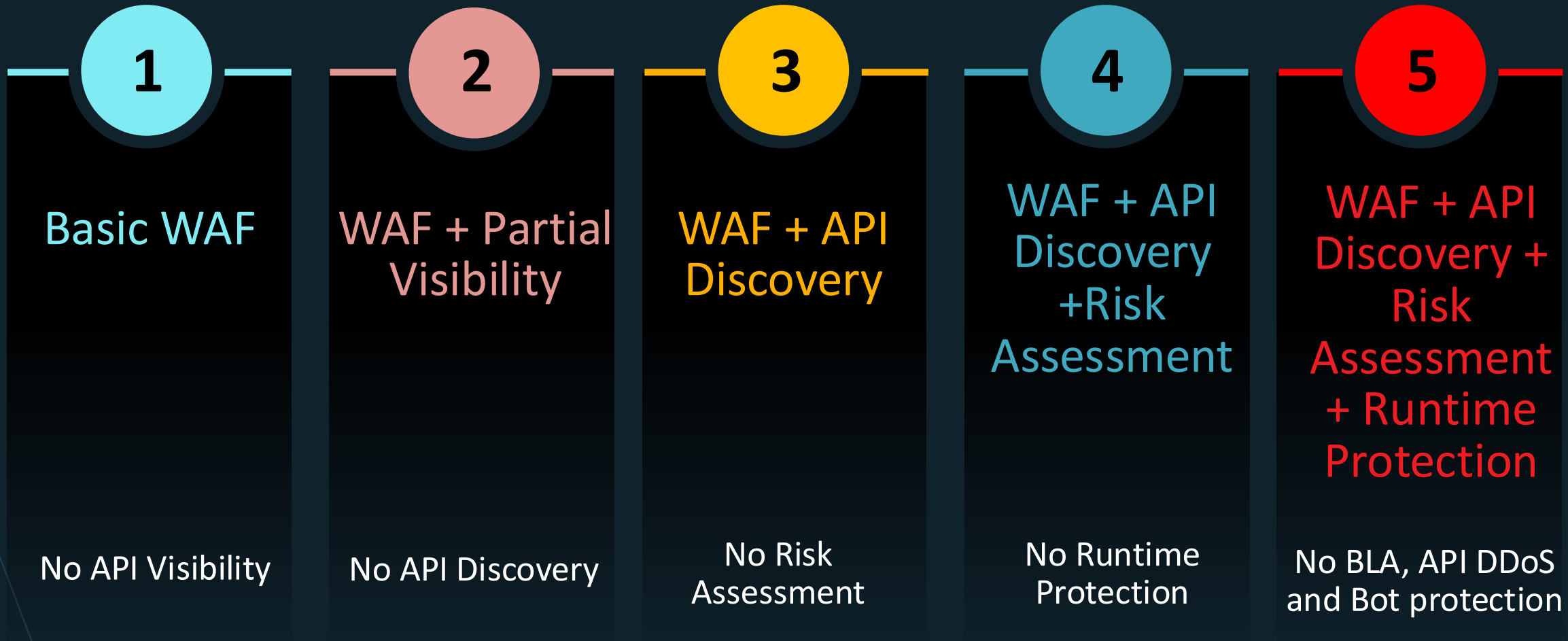
2 Business Logic Attacks

- API Call Manipulations to alter pricing
- Bypassing rate limits to scrape data
- Exploiting order workflows for fraud

3 API DDoS Attacks

- HTTP floods of API-based apps, Mobile Apps and Web-API hybrid apps

API Security – Maturity Journey



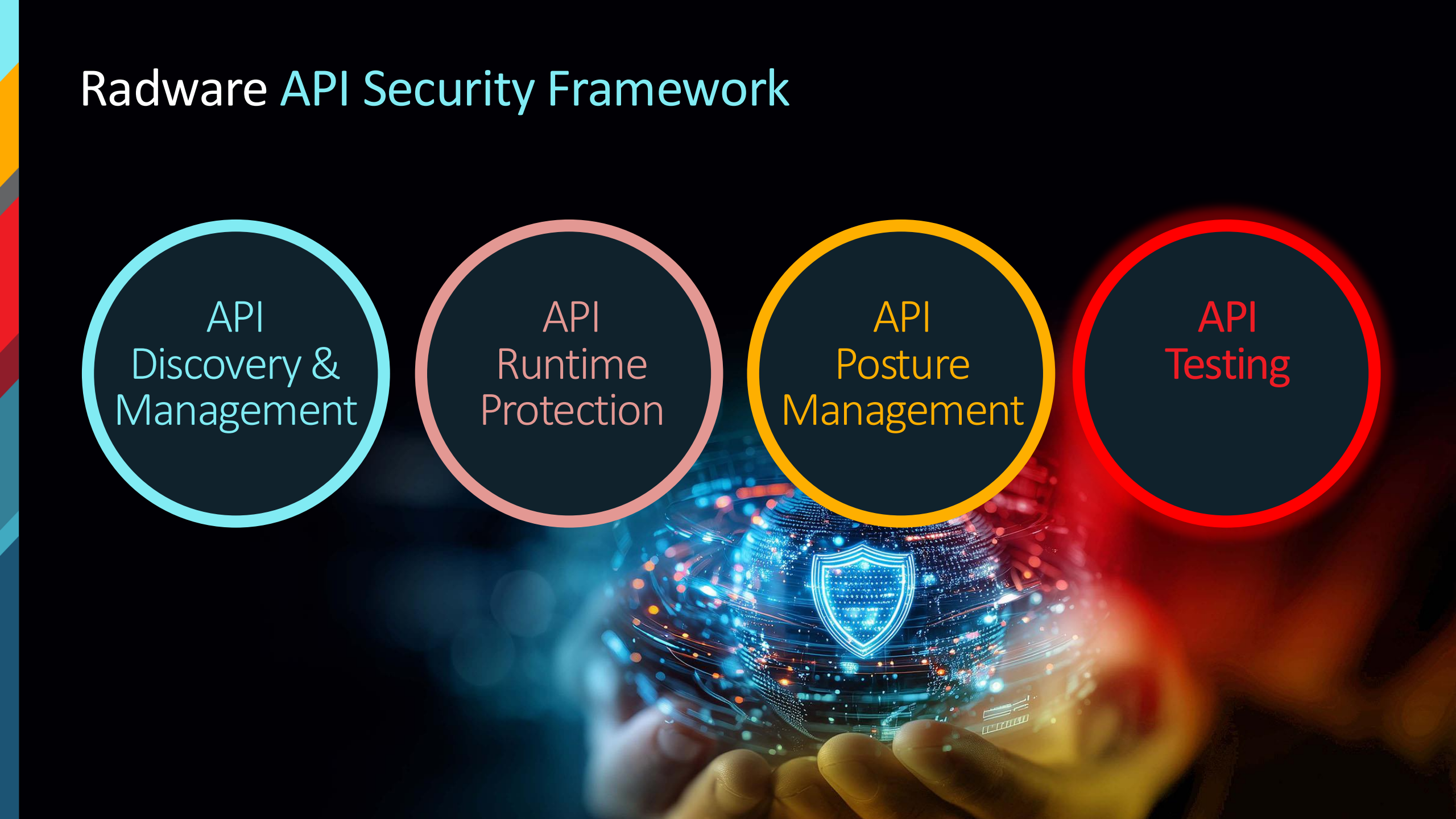
Radware API Security Framework

API Discovery & Management

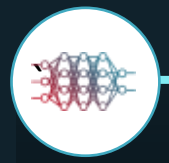
API Runtime Protection

API Posture Management

API Testing



Radware API Security Strategy: Runtime-First Protection



How we See

Inline & SecurePath

- Observes **real production traffic**
- No code changes
- No traffic mirroring
- No blind spots

We display what is used, not what was declared

How we Act

- **Business Logic Attack detection**
- OWASP API Top 10 coverage
- **Automated enforcement**

We stop real attacks as they happen

How we Understand

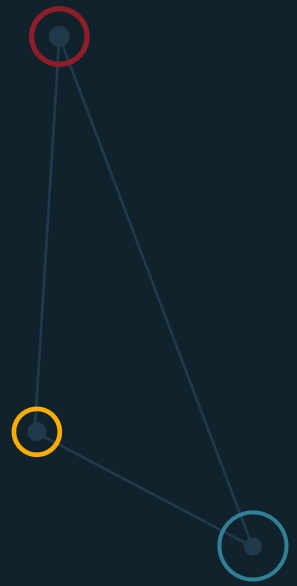
- **Continuous risk assessment**
- Rolling risk view
- Prioritized, contextual risks
- Recommendation on Risks

We detect real risk, not theoretical exposure

radware

API Discovery & Management

“You can’t secure what you don’t know exists”



API Discovery: The Foundation of API Security

Why it matters

What it does

- Discovers APIs from real traffic
- Identifies endpoints, param…
- Tracks changes over time
- Exposes shadow / zombie APIs

How it works

Drill Down in API Visibility

Business-Critical API Identification

Which APIs really matter

Performance Degradation as Early Warning

When performance becomes a risk signal

Bad API Usage & Abuse Detection

When errors reveal abuse

Security Abuse Visibility (OWASP API Signals)

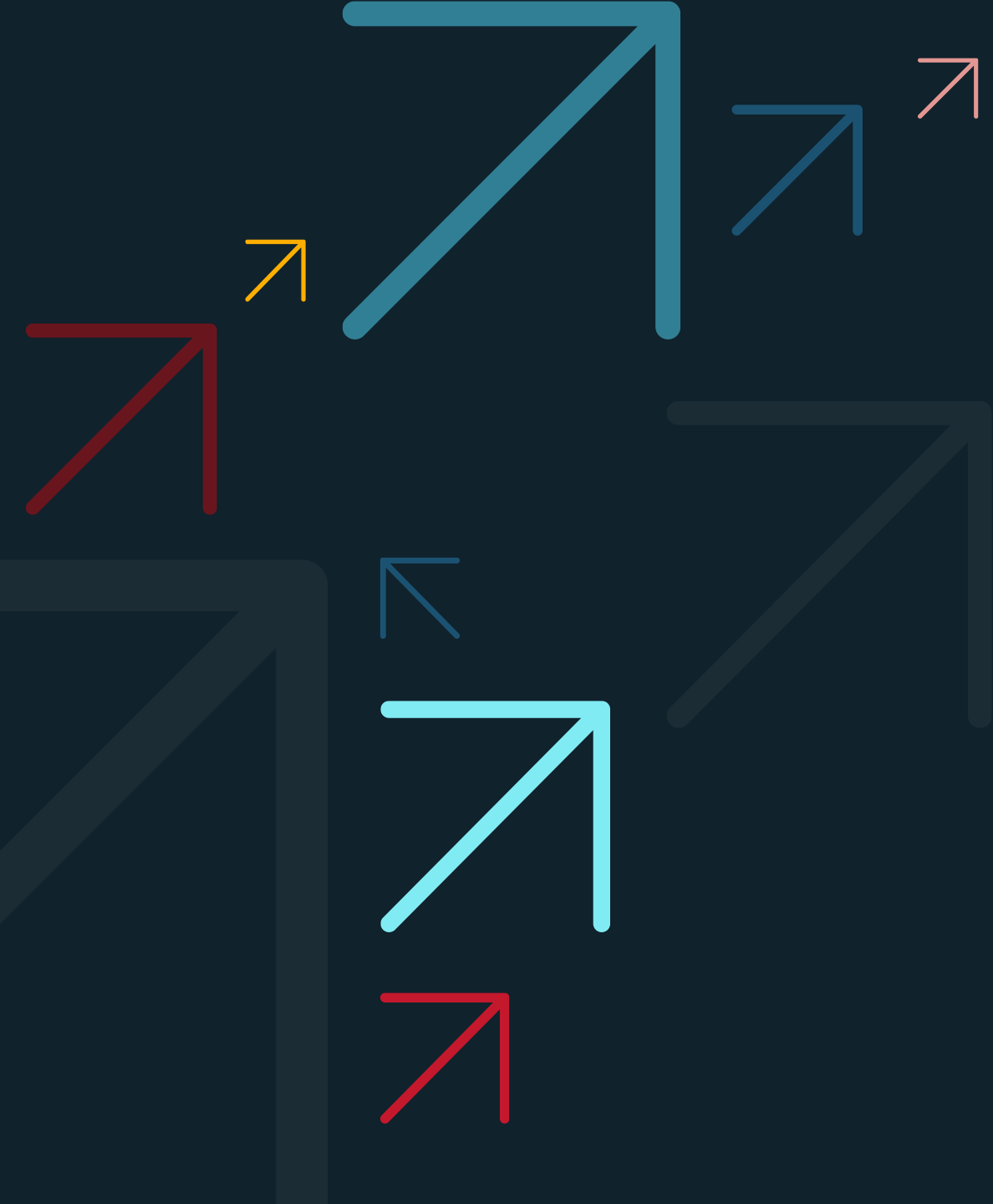
Traffic looks legitimate, but isn't

Application Workflow Visibility

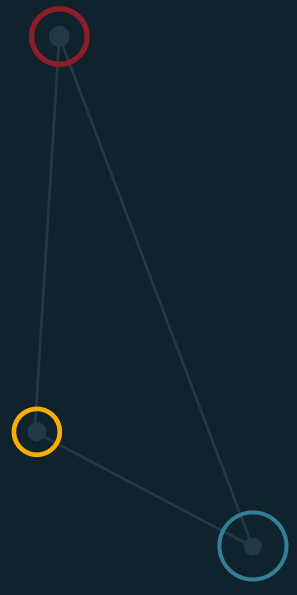
How an application is used / misused

radware

API Posture Management



“We don’t guess what an API is - we observe what it does.”



Runtime Posture Management

Bridge between development, intent, and production reality.

- Continuous assessment of **real API traffic in production**
- Based on **observed behavior**, not assumptions
- Covers application **security policy, configuration, exposure, and usage risks**
- Drives **protection decisions and risk prioritization**

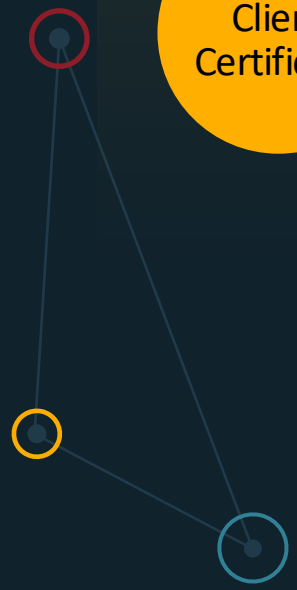
Runtime API Posture — Types of Security Controls

Each control type is continuously validated using **observed production traffic**.

radware

API Runtime Protection

API Runtime Protection



AUTHENTICATION

- Token Authz Control
- Client Certificate

API ABUSE MITIGATION

- Quota Limit (REST)
- Query Limit (GraphQL)
- Global Rate Limit

NEGATIVE SECURITY

- Malicious Payload Protection
- Data Leakage Prevention
- API Bot Protection

POSITIVE SECURITY

- GraphQL & REST
- **Full** Schema Enforcement
  Documented & Undocumented
- **Business Logic Protection**

VOLUMETRIC ATTACK

- API L7 DDoS Protection

SECURITY ADD-ONS

- LLM Firewall
- EAAF
- Threat Intelligence

Business Logic Attacks (BLA) Explained

What Is it?

When you use an app, you follow a set of expected steps—this is called business logic.

API business logic attacks exploit these legitimate workflows to achieve unauthorized actions without triggering traditional security alarms.

Why It Matters?

- Traditional security often miss it
- Attackers rely on it to bypass protections

Example: Sneaky Checkout Discount

1. E-commerce site offers discount only to first-time users
2. Attacker signs up, gets the discount & notices the discount logic is triggered by a specific API call
3. Attacker replay the API call - getting the discount again & again

✔ They didn't hack the system

✘ They just abused the intended flow -> business logic attack

Why Traditional Cannot Block BLA Attacks

Expected Business Conditions

- Same User
- Same Beneficiary
- Same Amount
- Valid Time Window
- One Confirmation per Validation

Step 1

GET /accounts/123

Valid & Expected

- Validate schema
- Validate authentication
- No signature match

Normal Account Access

Step 2

POST /transfer/validate

Valid & Expected

- Validation token created
- Short-lived state
- One-time use expected

Transfer pre-checks completed

Step 3

POST /transfer/confirm

Valid Request; Invalid Intent

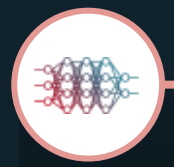
- Executed too many times
- No recent validation
- Reused validation token
- Amount mismatch

What Traditional Security Sees

✓ Validate Schema ✓ Validate Authentication ✓ No Malicious Payload

The request is **valid**. The context is **not**.

Runtime Protection – Business Logic Protection



Continuous Learning of Business Logic

- **Context Analysis** of all API transactions
- From **real time transactions**
- From all users / clients
- **Adapts to API updates** and changes

Automatic Policy Generation

- Policies that can **identify business logic vulnerabilities** exploitations
- Creation of business logic rules security policies, that ensures **lowest false positive rate**
- Continuously optimized

Real Time Detection & Mitigation

- **Detect, Block and Alerts on business logic attacks – as they occur**
- Highly **accurate mitigation**
- Precise identification and blocking of **bad actors** (not just their IPs)

Unparalleled **real-time detection** and **immediate mitigation** of business logic attacks

Business Logic Attack Rules Examples

Sequencing

Parameter Cardinality

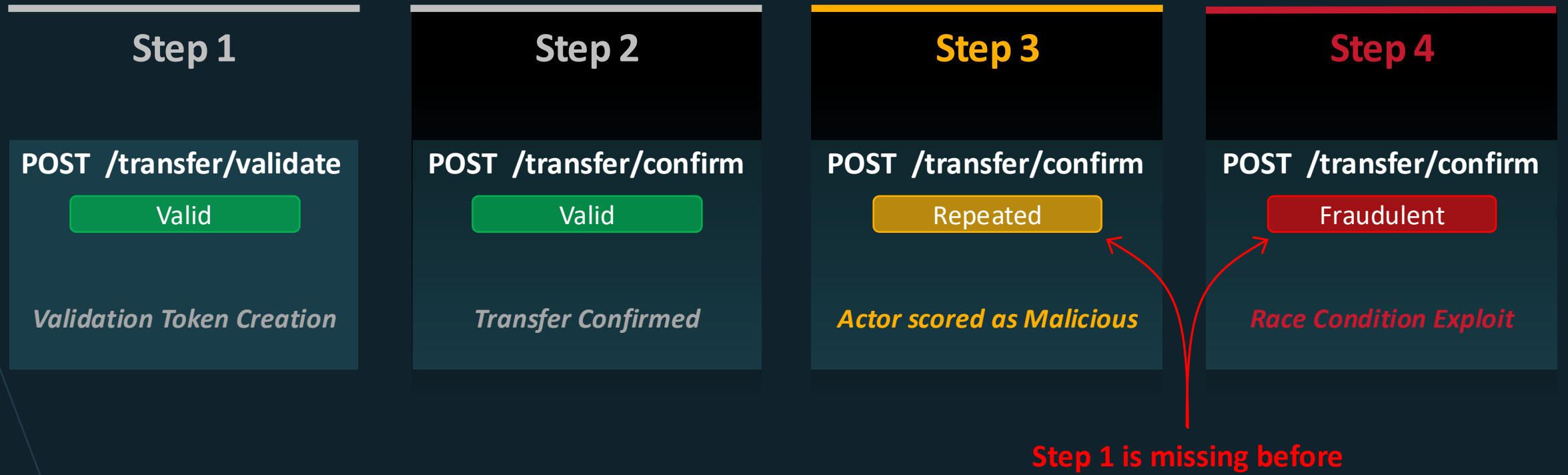
Parameter Allowed

Sequencing

OWASP Top 1 (BOLA)

OWASP Top 6

Ensures certain APIs are called in a specific order, based on observed workflows

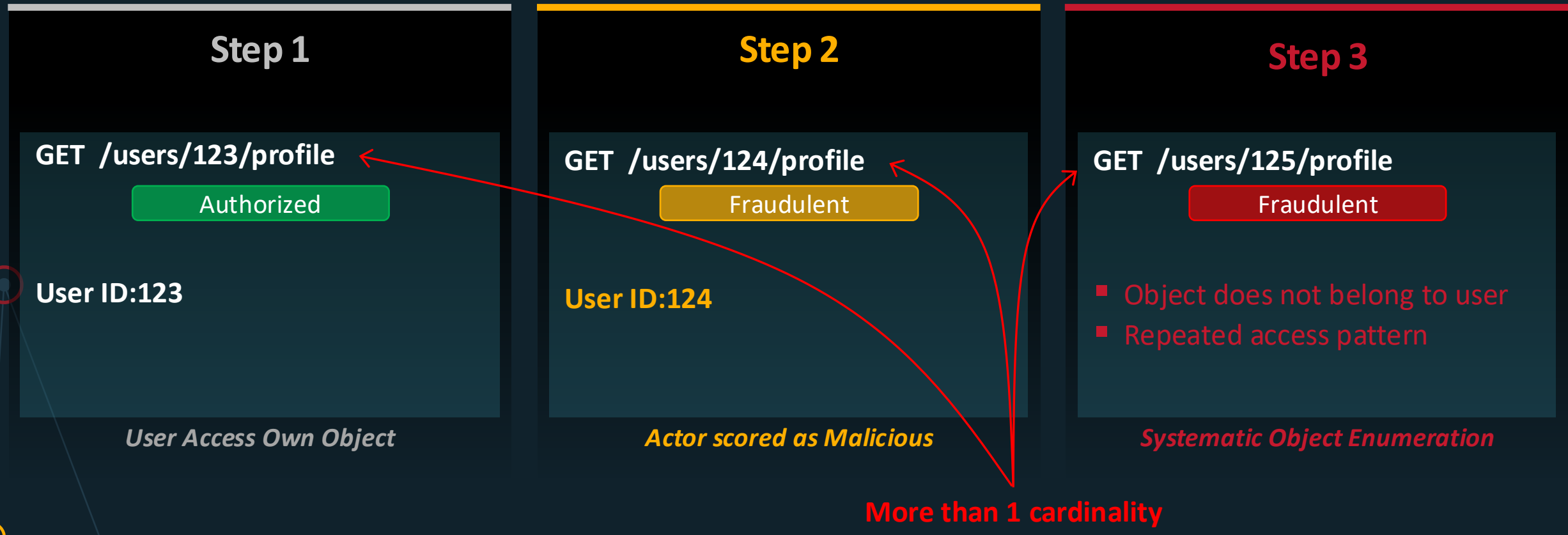


The workflow is abused. Not the API.

Cardinality

OWASP Top 1 (BOLA)

Ensures certain APIs are called in a specific order, based on observed workflows



Authorization is **correct per request**. ***Broken across objects***.

Parameter Allowed

OWASP Top 3 (BFLA)

Defines the full set of allowed parameter names for a specific API endpoint

Step 1

GET /accounts/123

Expected

Response fields:

- **id**
- **name**
- **balance**

Standard Response

Step 2

GET /accounts/123?include=details

Is it Valid?

Response fields expanded:

- **id**
- **name**
- **balance**
- **accountType**
- **createdAt**

More Data Returned

Step 3

GET /accounts/123

Malicious exposure

Response includes:

- Role
- riskFlag
- MarketingScore

× *Server-Side state changed*
× *Sensitive Fields Returned*
× *No Role Justification*

Authorization is **valid**. Data exposure is **excessive**.

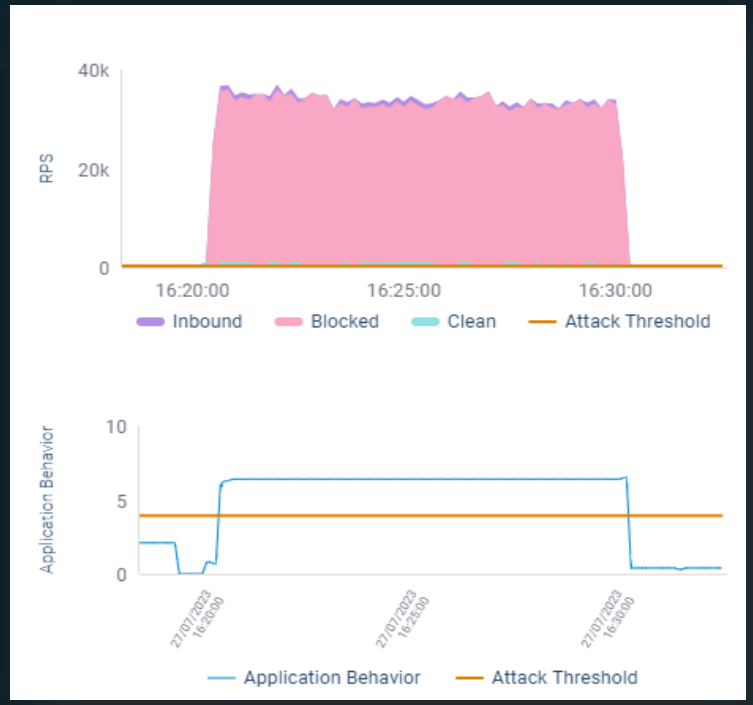
Business Logic Attack Rules

- Sequencing
- Parameter Cardinality
- Parameter Allowed
- Error Rate / Ratio
- Unique Out of Catalog Error Rate / Ratio
- Unique Out of Catalog Rate / Ratio
- Parameter Value
- Identical Parameters
- Parameter Mandatory
- Parameter Type
- Parameter Array
- Resource Suffix Rate

Industry Leading L7 DDoS Protection

Detection

Combines both **rate-based** (RPS) and **Rate invariant** (Applicative Behavior) baselines



Mitigation

Automated, real-time signatures tailored to attack characteristics

Latest Real Time Signature

HTTP Method = GET, HEAD, ST AND
Number of Query Arguments = 0 AND
Number of Cookies = 0, 4 AND
Number of Standard Headers = 10, 9 AND
Number of Non-Standard Headers = 3, 4 AND
Header 'cache*' exist AND
Header 'pragma' exist AND
Header 'sec-*' exist AND
Header 'upgrade-insecure-requests' exist AND
Header 'via' exist

OWASP API Security Top 10

Business Logic Protection

API1:2023

Broken Object Level Authorization

Improper access controls allowing users to view or modify other users' data through API endpoints.

Example: Changing user ID in API request to access another user's profile information.

Business Logic Protection

API3:2023

Broken Object Property Level Authorization

Exposure of sensitive object properties through API responses without proper authorization checks.

Example: User role field exposed in API response allowing unauthorized modification.

Business Logic Protection

API5:2023

Broken Function Level Authorization

Users able to access API functions beyond their authorized permission level.

Example: Regular user accessing admin-only API endpoints.

Token Authz Control

ATO Protection

API2:2023

Broken Authentication

Flaws in authentication mechanisms that allow unauthorized access to API endpoints.

Example: Weak token validation allowing expired tokens to remain valid.

Quota Limit

Strong Schema Enforcmt.

API4:2023

Unrestricted Resource Consumption

Lack of proper rate limiting or resource quotas leading to potential DoS attacks.

Example: Unlimited API requests causing server overload.

OWASP API Security Top 10

Token Authz Control

Business Logic Protection

API6:2023

Unrestricted Access to Sensitive Business Flows

Critical business processes exposed without proper controls or rate limiting.

Example: Automated bulk purchases bypassing inventory controls.

Strong Schema Enforcmt.

Signature based Protection

API7:2023

Server Side Request Forgery

API endpoints vulnerable to malicious requests targeting internal systems.

Example: API fetching data from user-provided URLs without validation.

Strong Schema Enforcmt.

Signature based Protection

API8:2023

Security Misconfiguration

Improper security settings in API infrastructure, frameworks, or dependencies.

Example: Exposed error messages revealing sensitive system information.

API Discovery

Business Logic Protection

API9:2023

Improper Inventory Management

Lack of proper documentation and monitoring of API endpoints and versions.

Example: Deprecated API versions remaining active with known vulnerabilities.

Strong Schema Enforcmt.

Signature based Protection

API10:2023

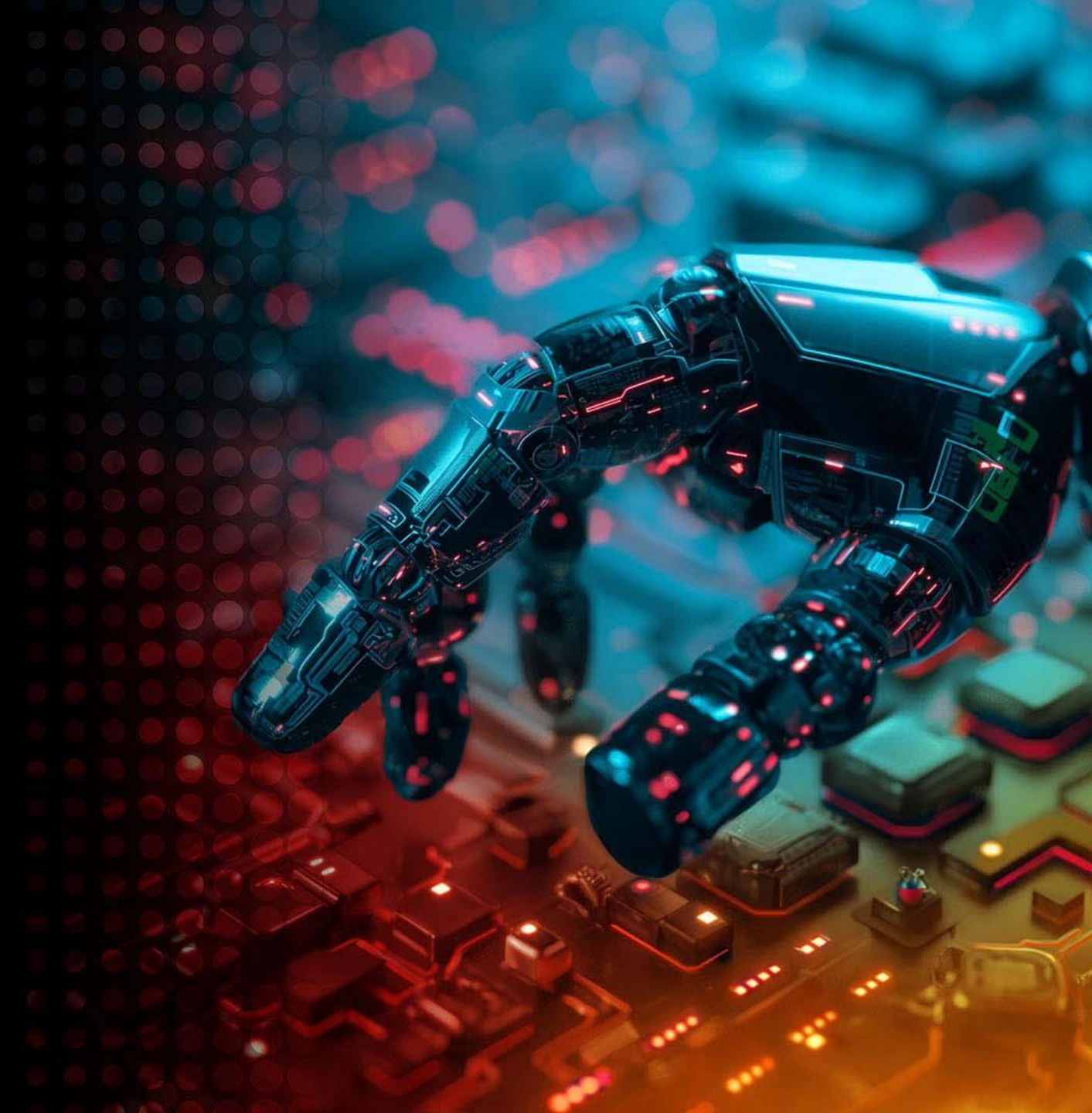
Unsafe Consumption of APIs

Insufficient validation of data received from external APIs leading to security issues.

Example: Blindly trusting and processing data from third-party APIs.

radware

Agentic AI Protection

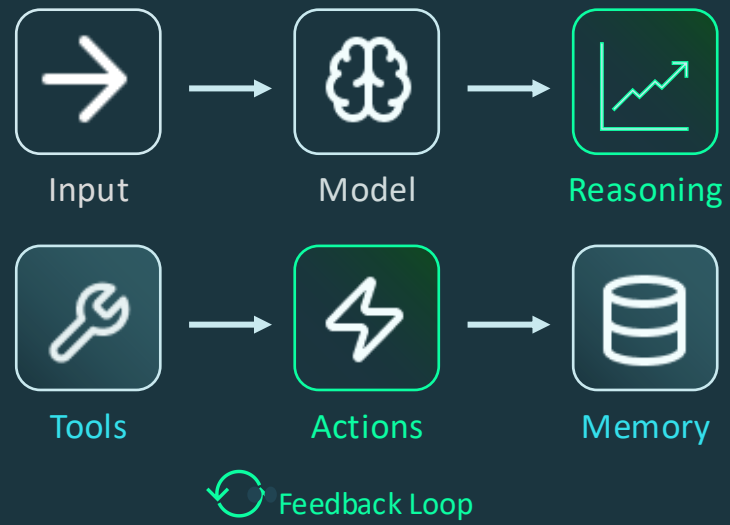


Chatbot vs. Agentic AI - Key Difference

TRADITIONAL CHATBOT

Static. No tools, no memory, no autonomous actions.

AGENTIC AI SYSTEM

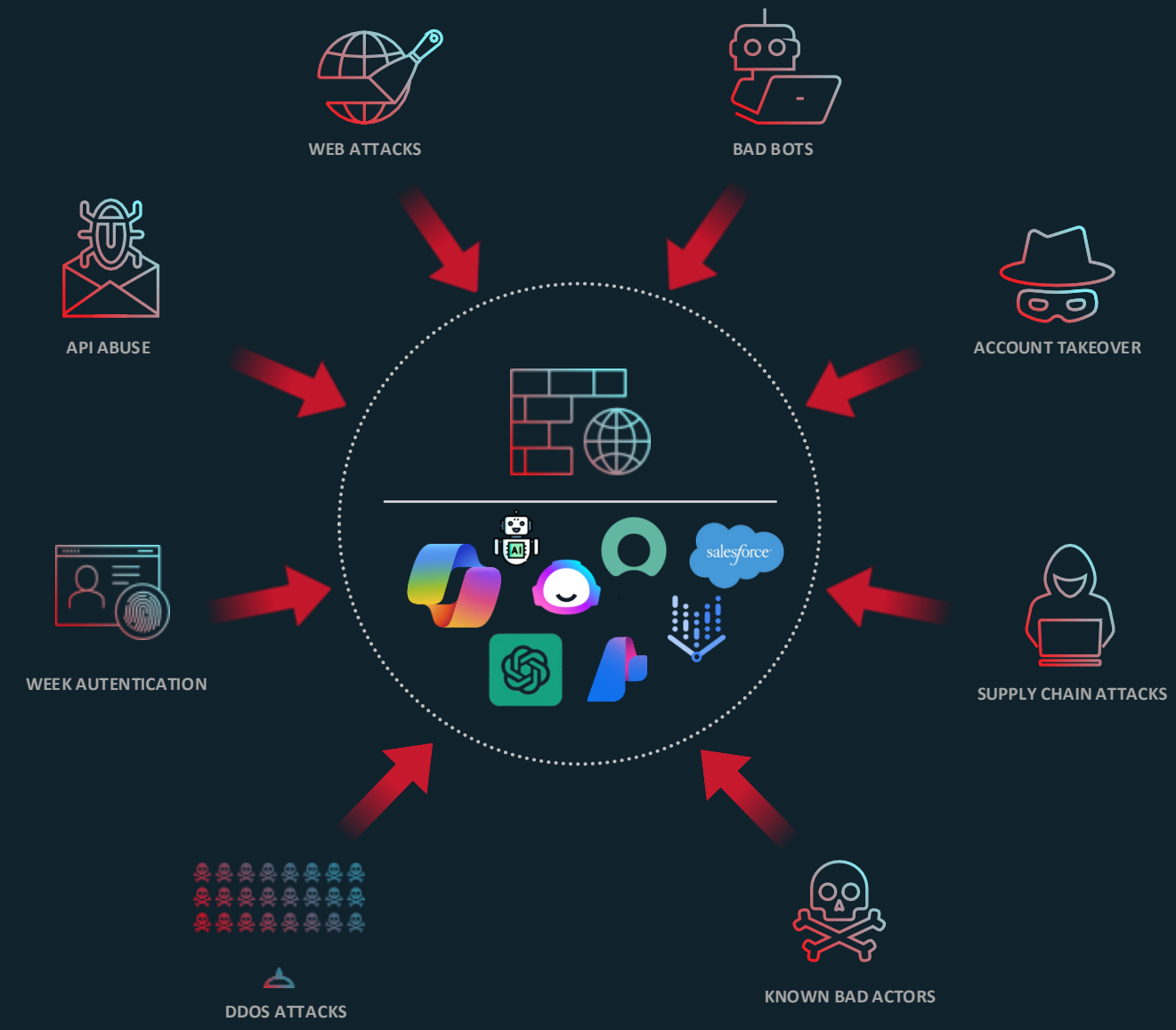


Dynamic, tool-using, stateful, capable of multi-step reasoning and real-world actions.

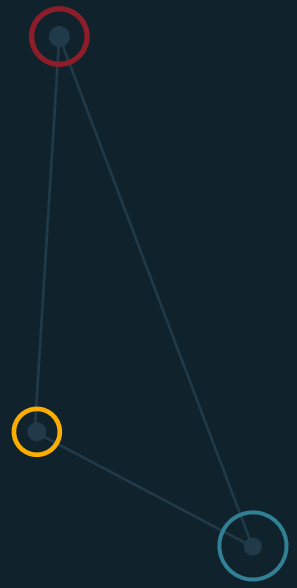
Chatbots respond. Agentic AI decides, acts, and learns.

Old Risk On New Technology Leads to New Security Challenges

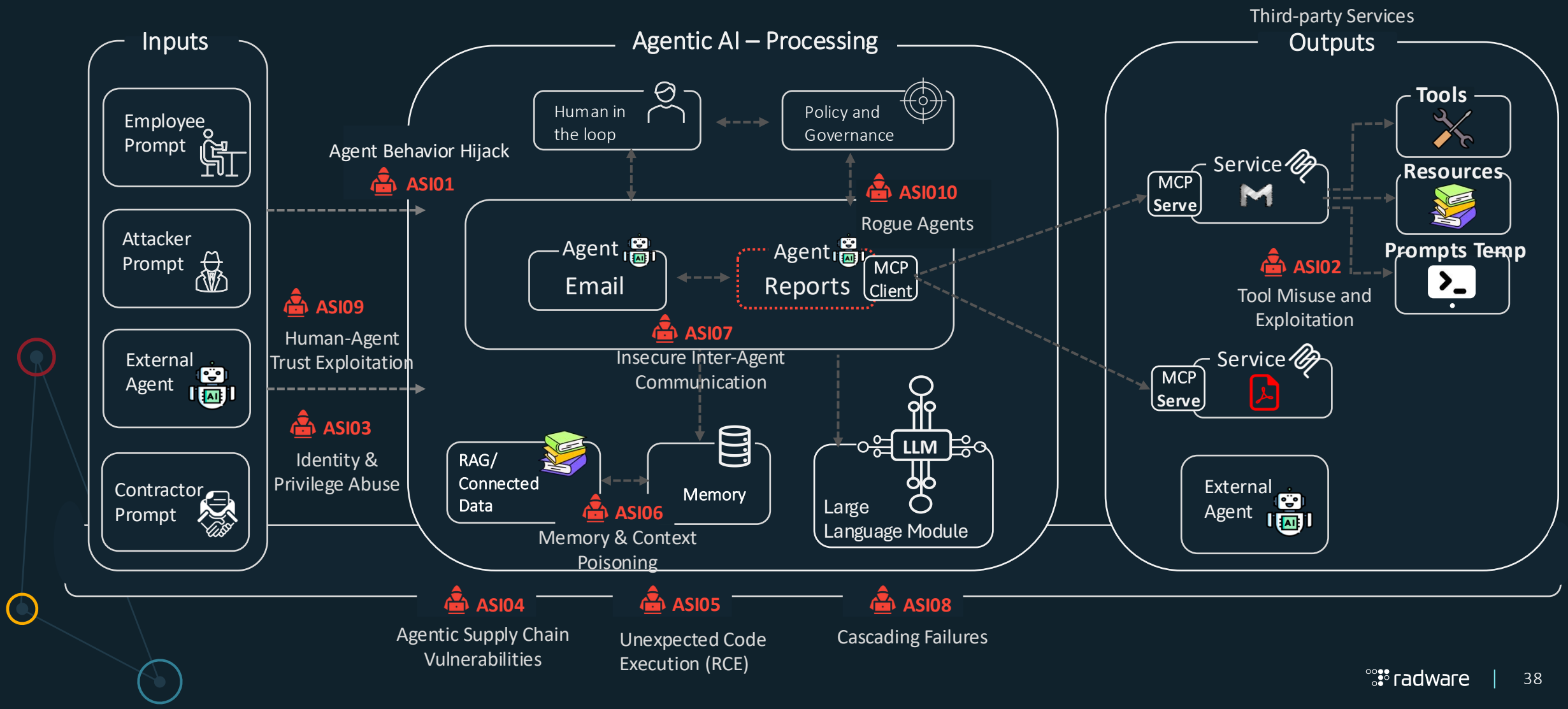
Old Threats In a New AI Form



“Traditional AI risks were about outputs. Agentic AI risks are about actions”

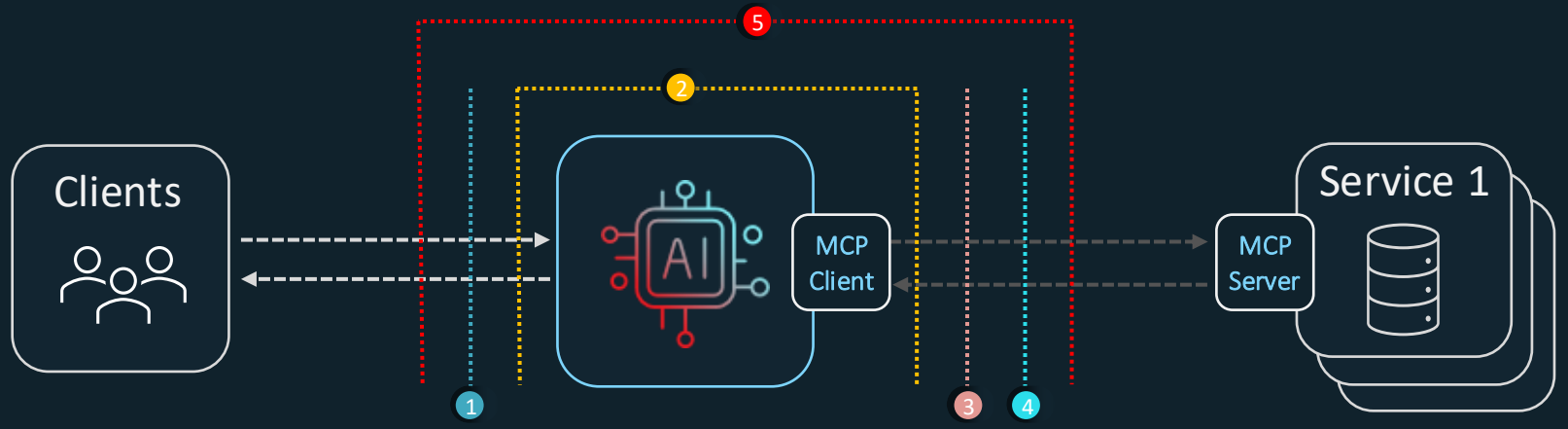


OWASP Top10 for Agentic Applications



Protect Your AI Apps & Agents

Adopt a Multi Layered Approach to Agentic AI Protection



LLM Firewall

Controls usage validates prompts, & prevents injection or data leaks

Behavior Guardrails

Monitors tool calls and blocks goal-divergent actions

Secure Agent Access

Restricts system entry to authorized agents only

Agent & Tool Discovery

Maps all agents and their accessible tools

Operational Analytics

Tracks agent activity, model usage, and tool interactions

Our Vision

Web & API Economy

POWERED by AI

Cloud Security Platform *powered by Radware EPIC-AI*	App & API Security Cloud DDoS Protection Web DDoS Protection
Cloud-Augmented **Application Delivery** & Security	Alteon Protect GEL
Industry Leading Network & Application **DDoS Protection**	DefensePro X w/ Cyber Controller

Agent Economy

PROTECT AI

Protecting LLM Models & AI Agents

LLM Firewall

Agentic AI Protection

SERVE AI

Enable Secure Agent Traffic

Bot & Agent Trust Management

Thank you!

radware