Traceable AI
Technical Brief

TRACEABLE

The API Security Challenge

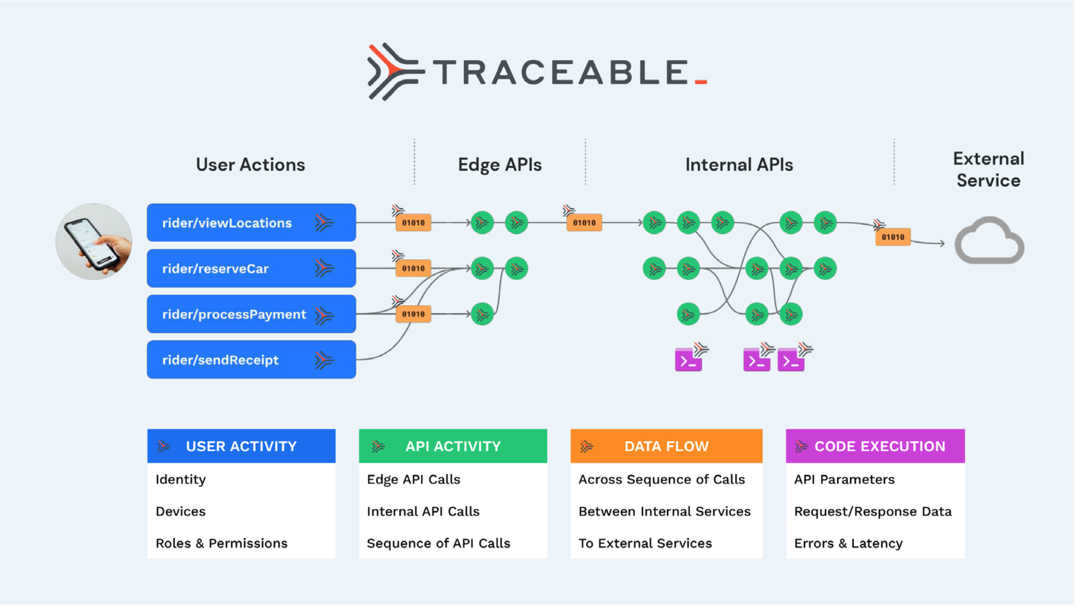
Applications have changed. The last several years have seen the rise of applications built on microservices architecture, deployed to scale on enterprise-grade cloud platforms. Their ability to elastically scale to match user demand has revolutionized the way applications are written and deployed. The connective tissue that enables microservices to work in tandem are APIs. APIs are the lifeblood that power these microservices-driven applications. With the growth of microservices applications growing each year, cybercriminals are changing their attack vectors to take advantage of the growing API attack surface that is emerging.

Organizations simply do not have the proper security tools to protect their expanding API attack surface. Existing application security tools that rely on signatures built on regular expressions to catch exploits generate a high number of false positives. The widespread use of APIs that power today's business success is getting blocked by traditional security solutions while allowing malicious cyberattacks to pass through to exploit API applications and exfiltrate sensitive delta. Modern API-driven applications move too fast, releasing new features while inadvertently releasing API vulnerabilities and business logic flaws. Existing security tools such as WAFs, RASP, and API gateways simply do not move fast enough to adapt to the speed of API application development and their security needs.

Traceable AI: A new vision for API Security

Traceable AI has built a new API Security application platform from the ground up, focused on protecting fast-moving API-driven applications that power today's enterprise-grade businesses. Built on an application observability platform, Traceable AI is able to offer API Security that moves in tandem with the API-driven applications. Powered by machine learning, Traceable AI security platform is able to collect data from user-driven transactions as they flow through an API-driven application. All collected data is stored within the Traceable AI platform, where machine learning is used to piece together the application's business logic into a logistical model. As the application adapts and changes over time, the ML model adapts along with it, learning new changes in normal application behavior.

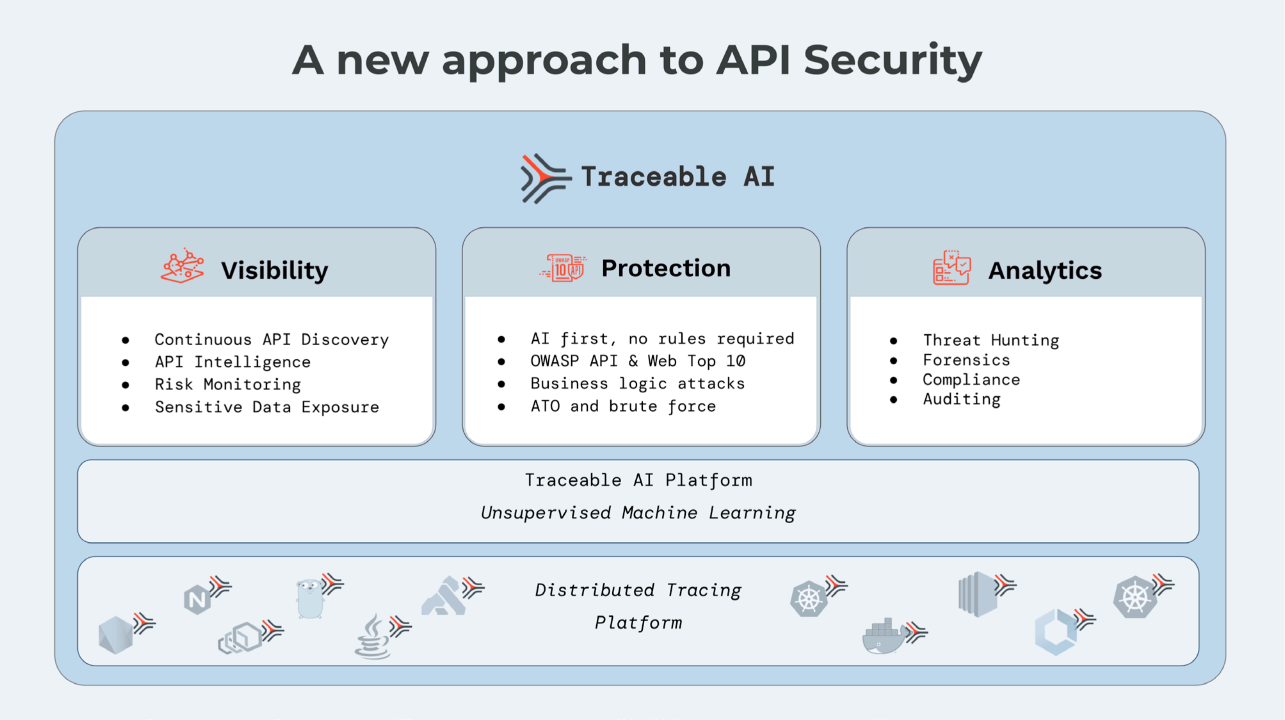
Figure 1: End-to-end visibility from user to data



The machine learning model for the application lays the foundation to accurately determine and separate legitimate(normal) from malicious user actions. As users start to access applications, Traceable AI will start to learn how users access the application, learning the user-intent of the application, as users work through an application. As more users access the application, Traceable AI will continue to build a very accurate statistical model of the application. What results is that Traceable AI is able to very accurately discern malicious use throughout the application. For malicious users that perform actions that are out of step with a normal user's sequences of actions with an application, Traceable AI will be able to immediately surface these deviations and determine their intent as malicious.

What results is a security platform that provides visibility, protection, and deep analytics for your mission-critical applications. Built on a distributed tracing platform, it collects traces from all touchpoints from across an application, storing for deep analysis through powerful machine learning algorithms that provide deep security insights on how your application intrinsically works and adapts over time, ensuring that your application security moves in lock-step with your application evolution.

Figure 2: Traceable AI security platform key benefits



A new approach to API Security

Traceable AI

Visibility

- Continuous API Discovery
- API Intelligence
- Risk Monitoring
- Sensitive Data Exposure

Protection

- AI first, no rules required
- OWASP API & Web Top 10
- Business logic attacks
- ATO and brute force

Analytics

- Threat Hunting
- Forensics
- Compliance
- Auditing

Traceable AI Platform

Unsupervised Machine Learning

Distributed Tracing Platform

How it works

Traceable AI security platform is split into three parts:

1. Instrumentation
2. Observability platform
3. Machine Models

Instrumentation

In order for Traceable AI to obtain insights about your application, customers can choose from the following two options:

1. Internal Instrumentation
2. External Instrumentation

Internal Instrumentation

With "internal instrumentation" customers deploy Traceable AI agents inside their application. Traceable AI currently supports Python, Go and Java languages. The in-app agent would be deploying during run-time and when it is installed will copy all transaction details(request/response) and send them to the Traceable AI for storage and analysis. This level of instrumentation enables customers to receive the full benefits of the Traceable AI platform that including API discovery, threat discovery, and real-time blocking.

External Instrumentation

With "external instrumentation" customers can deploy the Traceable AI agent external to their application in the underlying infrastructure. The options to deploy in the underlying infrastructure can range from deployment.

1. External API Endpoint
2. Internal Microservices

Deployment touchpoints: Where does it protect?

1) External API Endpoint

On an external API Endpoint, customers can enable Traceable AI out-of-band that mirrors user-driven traffic from the endpoint. All copied traffic is sent to Traceable AI for storage and analysis.

2) Internal Microservices

For internal microservices that operate on the Kubernetes platform, customers can deploy an agent in a sidecar container on their Kubernetes POD. The external agent can then capture all incoming traffic that is directed to the microservice and sent it to Traceable AI for storage and analysis.

However, with external instrumentation, there are some limitations that customers receive in terms of the fidelity of data quality and threat protection.

Internal vs External Instrumentation Tradeoffs

Deployment options between Internal and External instrumentations do come with tradeoffs and benefits. With "internal" instrumentation, Traceable AI is able to provide a higher level of capabilities that are not offered through external instrumentation deployments. The following are the key benefits offered by an internal deployment:

1. API discovery
2. Sensitive data Tracking
3. Application Flow mapping
4. Threat Detection based on business logic & API parameters
5. Vulnerability Detection
6. In-line (real-time) threat blocking of cyber-attacks.

The following is a description of the tradeoffs with features that come with each deployment option when protecting your applications.

Traceable AI Levels of Deployment

	Traffic Mirroring	Gateway / Proxy	Side-car	In-app Agent
External API Endpoints	✓	✓		✓
Internal Micro-services			✓	✓

The following is a feature chart that comes with each Traceable AI deployment option.

Levels of Feature Enablement

	Traffic Mirroring	Gateway / Proxy	Side-car	In-app Agent
Agent				✓
API Intelligence	✓	✓	✓	✓
Threat Detection	✓	✓	✓	✓
Vulnerability Detection	Partial	Partial	Partial	✓
WAF				✓
Real-Time Blocking (Inline)				✓
Application Flow Tracking				✓

Observability Platform: Traceable AI

Traceable AI is built on an observability platform that enables a central location to consolidate application traffic and performed advanced analysis not possible from within the application. A rich context-rich data lake of transaction data enables Traceable AI to generate critical insights into the application. Traceable AI offers customers the option to redact sensitive data before it is stored within Traceable AI, in order to ensure privacy and compliance requirements. Traceable AI enables multi-tenancy, ensuring that each customer's information is secure and only viewable by each customer.

The key advantage of Traceable AI is that all request/response is copied from the instrumentation and sent to the Traceable AI platform for storage and analysis. Traceable AI enables multi-tenancy, where each customer data is isolated from other customers. There are options to redact sensitive data before it is stored within Traceable AI in order to ensure privacy and compliance with regulatory mandates.

Machine Model Development

Built upon an observability platform that collects trace data from across all touchpoints of the application, Traceable AI is able to utilize user-driven traffic to collect trace data from each transaction as it flows through the application and builds out machine learning models. During the learning phase, Traceable AI will build out from user-driven traffic a logistical model of the application utilizing machine learning and tracing technology. The logistical model builds out the following for each application:

1. Business Logic
2. API Parameter

Business Logic

The business logic model that is developed from user-driven traffic is a representation of how users "expect" the application to operate as they perform their activity. For example, a user might log in to a retail application, search for a number of products, click to add them to their shopping cart, pay by credit card, and then log out. Each of these steps in the user journey is executed within the application by an API Endpoint. There will be one API for login, another for searching, another API to add products to the shopping cart, and an API to pay before login out. All these steps are captured in the business logic model that is formed. When malicious users start to access the application, deviations will be immediately surfaced as they try to probe and exploit flaws in the business logic of the application that are outside of the business logic model.

API Parameter

The API parameter definition file is constructed by observing all requests and responses captured from user traffic.

This can include the following:

1. Authentication
2. Response type (text vs numeric)
3. Length of response type
4. User-type
5. Header type (POST, PUT, etc)
6. Content-Type

The primary use of obtaining the API parameter file is to use it as a validation security tool to prevent anomalous deviations in data input that seek to exploit vulnerabilities within the application. For example, an API endpoint is designed so that the user can input a maximum of 255 characters but a recent change in the code can enable a cybercriminal to send an unlimited number of characters that could potentially crash the API Endpoint.

Protecting against Sophisticated Attacks

These models lay the security foundation that enables Traceable AI to protect against the most sophisticated API attacks. By creating a baseline model of application behavior, it can immediately surface unusual or anomalous behavior that could potentially be a malicious cyber-attack. This enables Traceable AI to lay a powerful foundation to protect against the exploitation of API vulnerabilities.

Unusual behavior such as enumerating through customer-ids can reveal API vulnerabilities such as BOLA that enable malicious users to access API resources(such as viewing bank statements or personal photos) that they are not authorized to access. The business logic model can understand the normal and proper sequence of user behavior through the life of a session. Slight deviations uncover business logic flaws and API vulnerabilities that are not possible with an agentless security solution.

About us.

Traceable AI was founded by third-time entrepreneur Jyoti Bansal and Sanjay Nagaraj. Bansal and Nagaraj saw the massive adoption of cloud-native architectures firsthand during their time at AppDynamics and founded Traceable AI as a result to protect applications from next-generation attacks.

Traceable AI applies the power of machine learning and distributed tracing to understand the DNA of the application, how it is changing, and where there are anomalies in order to detect and block threats, making businesses more secure and resilient.

Traceable.ai

TRACEABLE