Checkmarx

Cloud Native Security im Fokus – Integration, Bewertung und Optimierung mit CNAPP und ASPM

Dr. Carsten Huth | Global Director of AppSec Advisory

Wie sich Anwendungsentwicklung verändert

Cloud-native bringt eine neue Sichtweise auf Applikationen

92% Unternehmen, die einen Sicherheitsvorfall durch eine selbst entwickelte Anwendung hatten.

91% Unternehmen, die wissentlich Anwendungen mit Schwachstellen deployed/released haben.

67% Applikationen die in der Cloud gehostet sind.

Wie sich Softwareentwicklung verändert

Cloud-native bringt eine neue Sichtweise auf Applikationen

WHAT IS IN IT

Custom code

Open source

Micro-services

APIs

HOW IT IS DEVELOPED

DevSecOps

Software supply chain

Development tools

Collaboration tools

HOW IT IS PACKAGED

Containers

Kubernetes

HOW IT IS DEPLOYED

IaC

Cloud

CNAPP – Cloud Native Application Protection Platform

- CNAPP wurde erstmals 2021 von Gartner definiert
- Eine CNAPP ist eine Plattform, die Cloud-native Sicherheits- und Compliance-Funktionen vereinheitlicht, um Cloudsicherheitsbedrohungen zu erkennen, zu verhindern, und darauf zu reagieren.
- Eine CNAPP integriert mehrere Cloudsicherheitslösungen, so dass Organisationen ihren gesamten Cloudanwendungsbedarf einfacher schützen können (CSPM, CWPP, CIEM, IaC Scanning, …).

“Im Jahr 2029 werden 60% der Unternehmen, die keine vereinheitlichte CNAPP Lösung in ihrer Cloud-Architektur einsetzen, keine umfassende Einsicht in deren Cloud Angriffsoberfläche haben und dementsprechend nicht ihre Zero-Trust Ziele erreichen.”

Gartner

Was ist das am schnellsten wachsende Software Unternehmen aller Zeiten?

https://www.independent.co.uk/tech/wiz-deal-takeover-google-rappaport-b2580046.html

Code zur Cloud auf einer Plattform

Schritte zur sicheren Entwicklung von der ersten Codezeile bis zur Laufzeit in der Cloud

SDLC

Training

Schulung zu sicherem Code
Qualifizierung von Entwicklern

Design

API-Sicherheit
Durchsucht die API-Doku und identifiziert Schatten-APIs

Bedrohungs-modellierung
(Threat-Modelling)
Sicher durch Design

Code

SAST
Findet Schwachstellen im Code

Erkennung von Geheimnissen
Findet Geheimnislecks in Collaboration-Tools

Build

SCA
Findet Schwachstellen in Open-Source-Software

Schädliche Pakete
(Malicious Package)
Findet Schadcode in Open Source

Prüfen

DAST
Testet kompilierte Apps auf Schwachstellen

Einsetzen

Container-Sicherheit
Scannt Container-Images

IaC-Sicherheit
Findet Probleme und Fehlkonfigurationen

Go-Live

CNAPP-Partner
Bietet Laufzeiteinblicke für die Korrelation

Heise DevSec: „Sichere Software beginnt vor der ersten Zeile Code“

Code to Cloud – Das Beste aus beiden Welten

AppSec

Den Kontext vervollständigen

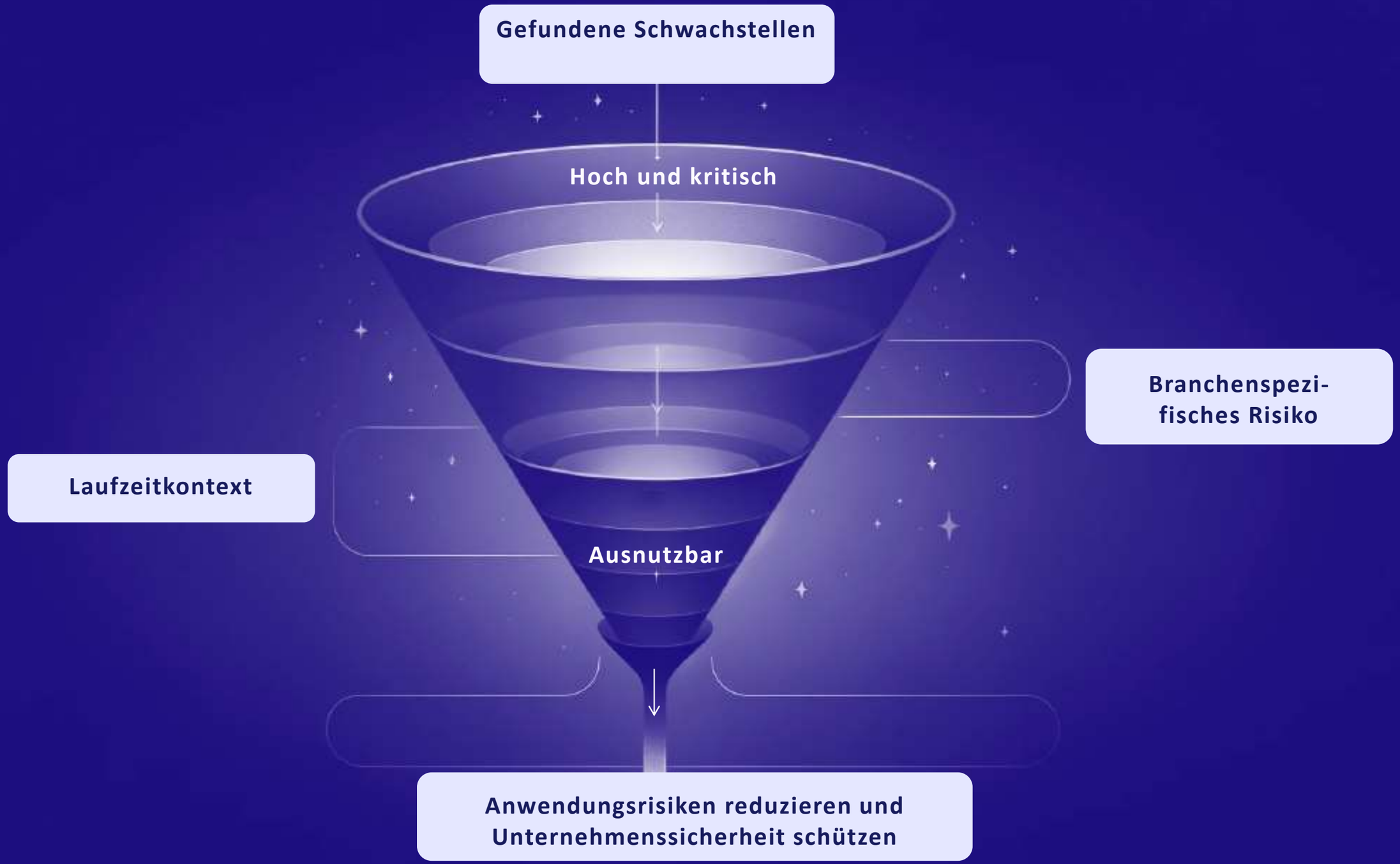
(Zielkontext)

+

CNAPP

Vervollständigung der Sicherheitslage

(Quell-Information)


Gefundene Schwachstellen
Hoch und kritisch
Branchenspezi-
fisches Risiko
Laufzeitkontext
Ausnutzbar
Anwendungsrisiken reduzieren und
Unternehmenssicherheit schützen

Code to Cloud –
Map Repo to Runtime

Projects - CxOne

deu.ast.checkmarx.net/cnas

Managed bookmarks | Whatsapp | Synerion | Service Desk | Cx1 Documentation | OWASP | Onboarding | Ori Bendet & Kobi T... | Log In | CxCareer | Tenants and API | Tech Partners Inter... | Tech Partners Exter... | Training

wiz-test

wiz-test (Cloud Insights)

Scan Date: Jul 29 2024 15:20 | Status: Connected | Manage Accounts

Attack Paths | Inventory

Filters

Image Name	Project Name	Repo URL	Risk Level	Total Vulnerabilities	Internet Facing
demo-store (11)					
gcp-checkoutservice	...arxIntegrations/gcp-checkoutservice	...heckmarxIntegrations/gcp-checkoutservice	High	9	-
emailservice	CheckmarxIntegrations/emailservice	...b.com/CheckmarxIntegrations/emailservice	High	11	-
paymentservice	...heckmarxIntegrations/paymentservice	...com/CheckmarxIntegrations/paymentservi...	High	11	-
loadgenerator	CheckmarxIntegrations/loadgenerator	....com/CheckmarxIntegrations/loadgenerator	High	11	-
shippingservice	...eckmarxIntegrations/shippingservice	...om/CheckmarxIntegrations/shippingservice	High	9	-
adviceservice	...rxIntegrations/recomendationservice	...eckmarxIntegrations/recomendationservice	High	10	-
prdcatalogservice	...kmarxIntegrations/prdcatalogservice	.../CheckmarxIntegrations/prdcatalogservice	High	3 72 37 1	-
frontend	CheckmarxIntegrations/frontend	...ithub.com/CheckmarxIntegrations/frontend	High	113	
adservice	CheckmarxIntegrations/adservice	...thub.com/CheckmarxIntegrations/adservice	High	5	-
cxone-cartservice	...kmarxIntegrations/cxone-cartservice	.../CheckmarxIntegrations/cxone-cartservice	Medium	2	-
currencyservice	...eckmarxIntegrations/currencyservice	...om/CheckmarxIntegrations/currencyservice	High	11	-

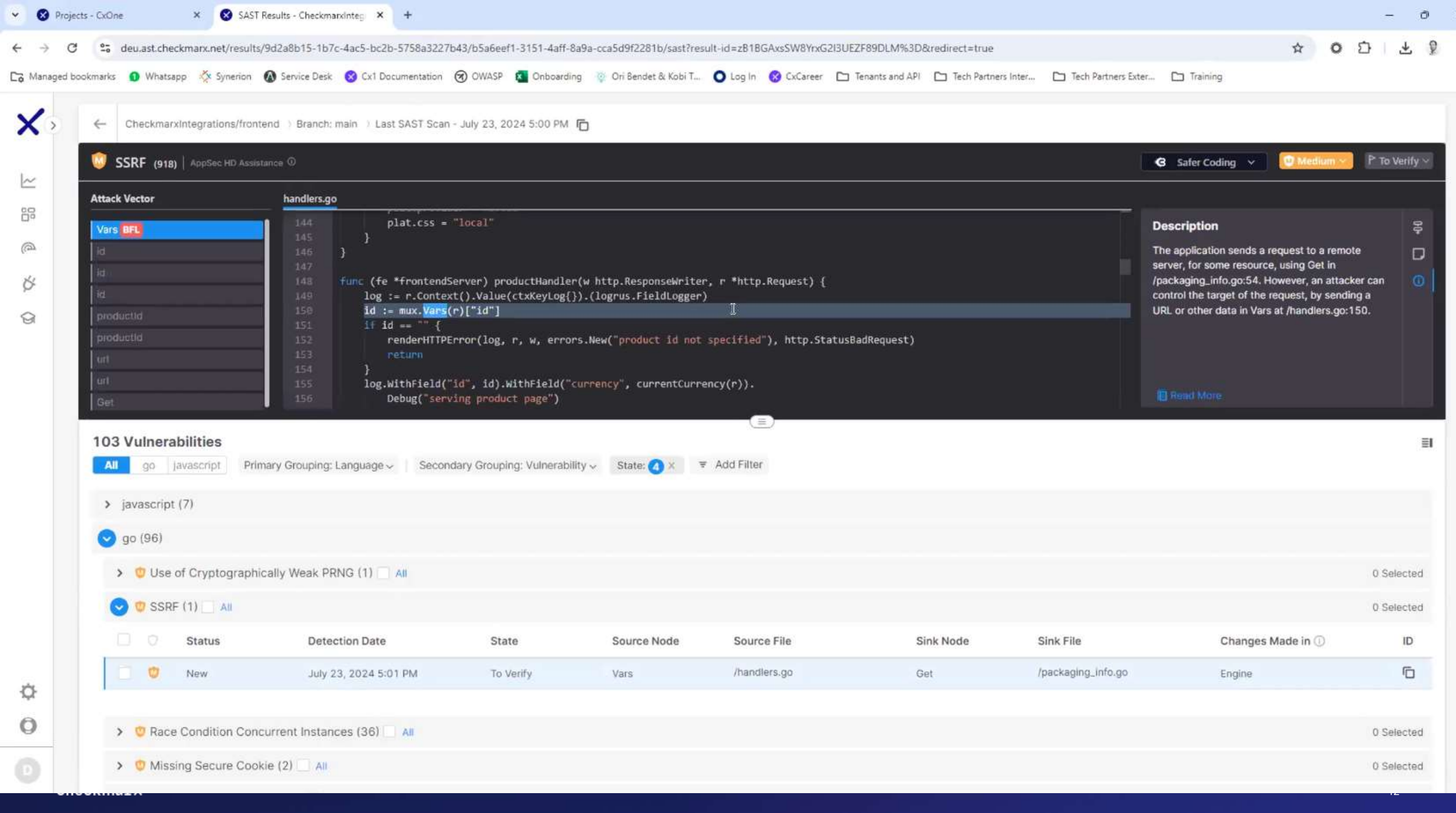
Rows 20

1

eks-load-balancer-demo (1)

traefik-ingress-demo (1)

Code to Cloud – Attack Path


Projects - CxOne
SAST Results - CheckmarxInteg
deu.ast.checkmarx.net/results/9d2a8b15-1b7c-4ac5-bc2b-5758a3227b43/b5a6eef1-3151-4aff-8a9a-cca5d9f2281b/sast?result-id=zB1BGAxsSW8YrxG2I3UEZF89DLM%3D&redirect=true
Managed bookmarks
Whatsapp
Synerion
Service Desk
Cx1 Documentation
OWASP
Onboarding
Ori Bendet & Kobi T...
Log In
CxCareer
Tenants and API
Tech Partners Inter...
Tech Partners Exter...
Training
CheckmarxIntegrations/frontend
Branch: main
Last SAST Scan - July 23, 2024 5:00 PM
SSRF (918) | AppSec HD Assistance
Safer Coding
Medium
To Verify
Attack Vector
Vars BFL
id
id
id
productId
productId
url
url
Get
handlers.go
144 plat.css = "local"
145 }
146 }
147
148 func (fe *frontendServer) productHandler(w http.ResponseWriter, r *http.Request) {
149 log := r.Context().Value(ctxKeyLog{}).(logrus.FieldLogger)
150 id := mux.Vars(r)["id"]
151 if id == "" {
152 renderHTTPError(log, r, w, errors.New("product id not specified"), http.StatusBadRequest)
153 return
154 }
155 log.WithField("id", id).WithField("currency", currentCurrency(r)).
156 Debug("serving product page")
Description
The application sends a request to a remote server, for some resource, using Get in /packaging_info.go:54. However, an attacker can control the target of the request, by sending a URL or other data in Vars at /handlers.go:150.
Read More
103 Vulnerabilities
All
go
javascript
Primary Grouping: Language
Secondary Grouping: Vulnerability
State: 4
Add Filter
javascript (7)
go (96)
Use of Cryptographically Weak PRNG (1) All
0 Selected
SSRF (1) All
0 Selected
Status
Detection Date
State
Source Node
Source File
Sink Node
Sink File
Changes Made in
ID
New
July 23, 2024 5:01 PM
To Verify
Vars
/handlers.go
Get
/packaging_info.go
Engine
Race Condition Concurrent Instances (36) All
0 Selected
Missing Secure Cookie (2) All
0 Selected

Code to Cloud – Risk Management

CloudInsights_5 Application | 7.8 High

+ Add Project

Overview | Projects | **Risk Management** | Settings

> Filters | All | Runtime

All Risks (16)

Vulnerability Name	Risk Score	Vulnerability Type	Days Open	Origin
CVE-2023-45288	7.8, High	Framework	90	Frontend View →
CVE-2024-24786	7.8, High	Framework	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend
Denial_Of_Service_Resource_Exhaustion	5.7, Medium	Infrastructure	90	Frontend

ASPM

Von der Triage von Schwachstellen zur Bewertung von Geschäftsrisiken

Priorisierung

- Risikobasiertes Anwendungsranking
- Direkte Kommunikation mit Entwicklern
- Algorithmen zur Erkennung kritischer Bedrohungen
- Auf die Bedrohungs-landschaft Reagieren

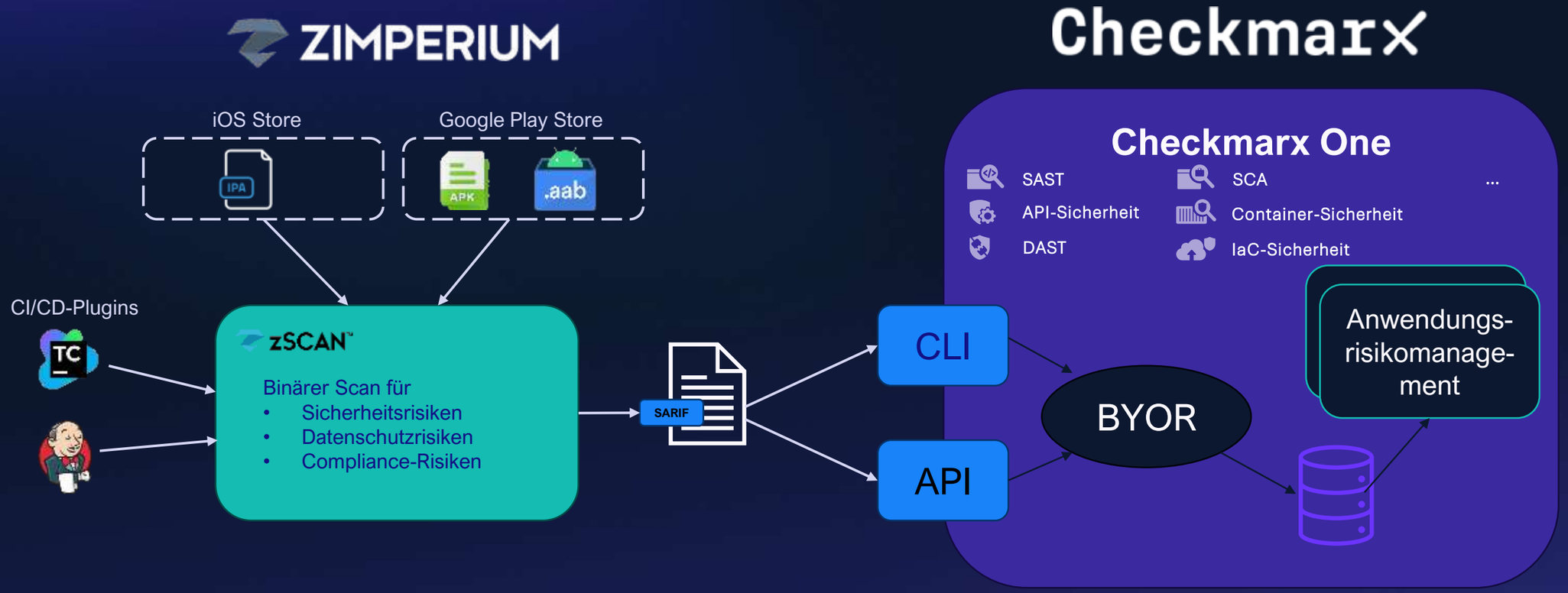
Abdeckung

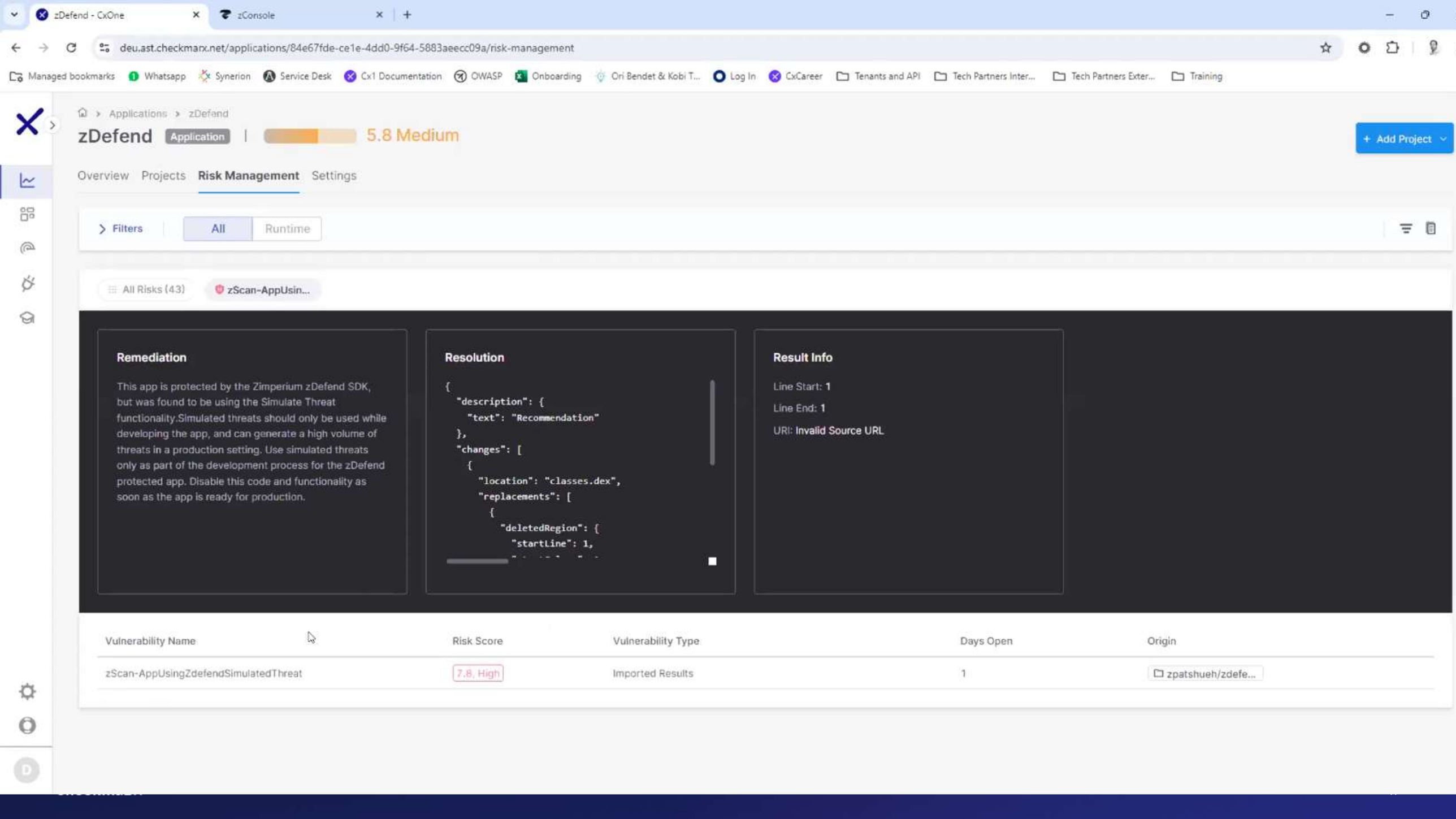
- Umfassend
- Gesamter Anwendungslebenszyklus
- Nahtlose Integration über Plattformen hinweg
- Zentralisierte Verwaltung und Überwachung

Skalierbar

- Richtlinienmanagement (policy management) zur Einhaltung von Industriestandards
- Anpassbare Richtlinien (policies)
- Transparentes Reporting und Analyse

Beispielablauf „Bring Your Own Results“ (BYOR)




zDefend - CxOne
zConsole
deu.ast.checkmarx.net/applications/84e67fde-ce1e-4dd0-9f64-5883aeecc09a/risk-management
Managed bookmarks
Whatsapp
Synerion
Service Desk
Cx1 Documentation
OWASP
Onboarding
Ori Bendet & Kobi T...
Log In
CxCareer
Tenants and API
Tech Partners Inter...
Tech Partners Exter...
Training
Applications > zDefend
zDefend
Application
5.8 Medium
+ Add Project
Overview
Projects
Risk Management
Settings
Filters
All
Runtime
All Risks (43)
zScan-AppUsin...
Remediation
This app is protected by the Zimperium zDefend SDK, but was found to be using the Simulate Threat functionality.Simulated threats should only be used while developing the app, and can generate a high volume of threats in a production setting. Use simulated threats only as part of the development process for the zDefend protected app. Disable this code and functionality as soon as the app is ready for production.
Resolution
{
"description": {
"text": "Recommendation"
},
"changes": [
{
"location": "classes.dex",
"replacements": [
{
"deletedRegion": {
"startLine": 1,
Result Info
Line Start: 1
Line End: 1
URI: Invalid Source URL
Vulnerability Name
Risk Score
Vulnerability Type
Days Open
Origin
zScan-AppUsingZdefendSimulatedThreat
7.8, High
Imported Results
1
zpatshueh/zdefe...

Möglichkeiten für KI in AppSec

01

AppSec mit KI Beschleunigen

- Automatische Schwachstellenbehebung
- Reduzierung falsch-positiver Ergebnisse (False-positives)
- Alternative Pakete, Version Pinning, Lizenzerkennung
- Unterstützung bei der Abfrageerstellung/-optimierung

02

Schutz vor KI Angriffen

- KI-Prompt-Injektionen
- KI-Halluzinationen
- KI-Geheimnisse/IP-Leck

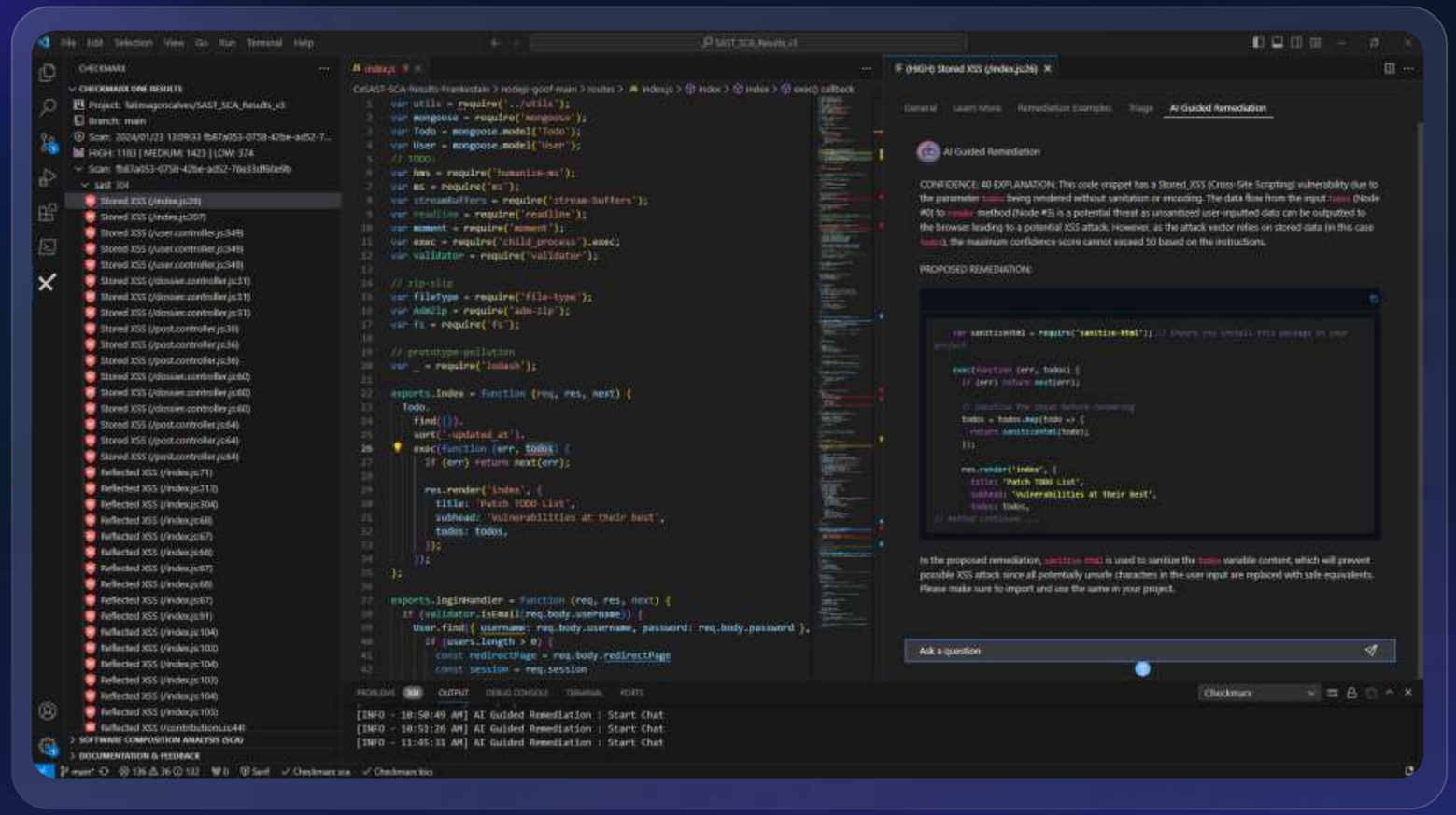
03

Unterstützung durch KI in Entwickler-Workflows

- Geführte Korrektur
- Scannen in der Entwicklungsumgebung

Unterstützen Sie Entwickler-Workflows

- Identifizieren einer Schwachstelle
- KI schlägt Code zur Behebung vor
- Generiert Erklärung der Sicherheitslücke und schlägt Code zur Behebung vor

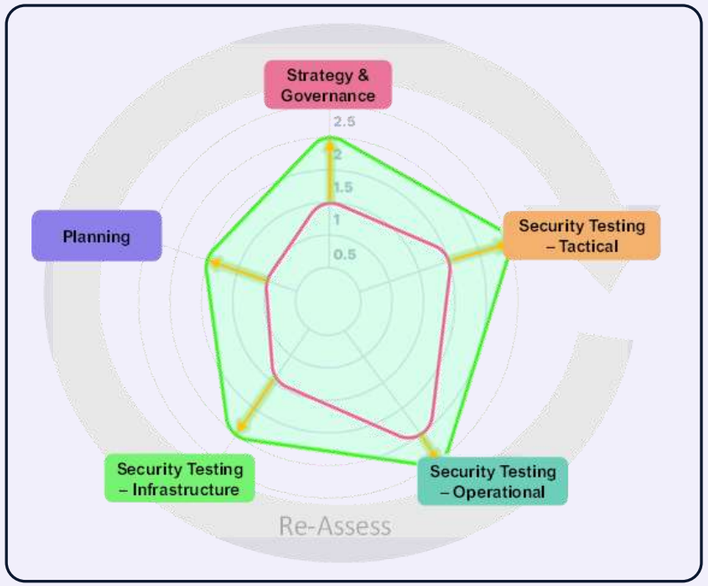


AppSec Programm Reifegrad

“AppSec is a Journey”

AppSec-Programmmethodik und -bewertung (APMA)

Bewerten

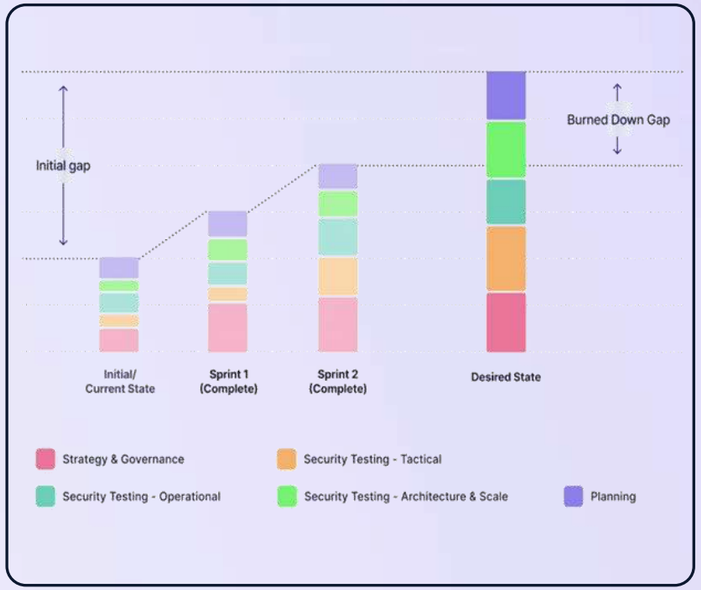


Bewertung und Lückenanalyse liefern quantifizierbare und umsetzbare Ergebnisse von Branchenexperten.

Planen

Gemeinsam überprüfen, beraten und analysieren, um einen umsetzbaren Verbesserungsplan zu erstellen.

Verbessern



Durch die agile Umsetzung anhand können Verbesserungen bestätigt und eine fortwährende Neubewertung durchgeführt werden.

Conclusion

- **Code to Cloud und CNAPP sind wichtig für Cloud Native Entwicklung**
  - Checkmarx unterstützt den Code to Cloud Fluss End-to-End

- **Holistisches Management von Applikationssicherheit ist von zentraler Bedeutung**
  - Priorisierung/Risk Management
  - Korrelation, Gesamten Entwicklungslebenszyklus abdecken

- **Developer Experience**
  - KI unterstütztes Beheben von Schwachstellen, Schutz vor KI-Angriffen
  - Ergebnisse in der Arbeitsumgebung bearbeiten

- **„AppSec is a Journey“** – Wir helfen mit Reifegradmodell und agilen AppSec-Programmen

Checkmarx

Securing the Applications Driving Our World