WTH is Reachability analysis

Path to 0 real critical – Shave off 90% of our vulnerabilities with reachability analysis

PHOENIX SECURITY

Francesco Cipollone, Co-Founder Phoenix Security, fc@phoenix.security

Warning revelation ahead

Warning

About Francesco

Francesco Cipollone

CEO & Co-Founder Security Phoenix, Board CSA UK

I'm a appsec passionate and have been a CISO Advisor, Cybersecurity Cloud Expert.
Speaker, Researcher and Board of Cloud security Alliance UK.
Currently we are working on interesting problem on how to link Application, Security and

@FrankSec42 | Fracipo Linkein | Email | Website | Articles | NSC42 LinkedIn

You can argue with people, but you can't argue with data

Data driven approach can help making compelling arguments

Agenda

Intro & Context

Current Scenario : 2015 to today – SLA, Critical, CVSS which one to choose

P1 - Challenges in prioritization with old metrics

P2 - Prioritizing right – reachability analysis

Reche ability analysis

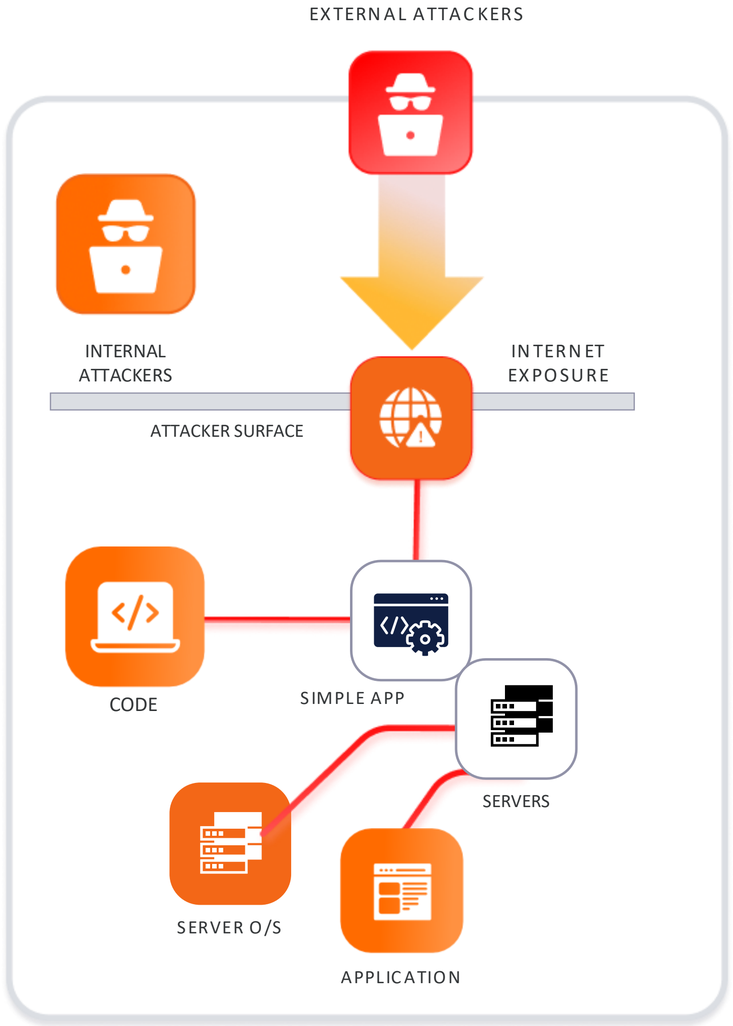
Container lineage and container throttling

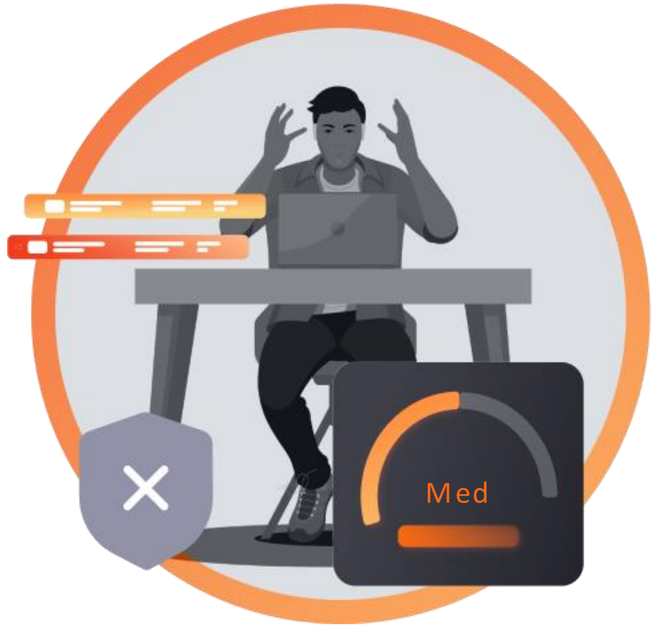
P3 – Bringing all together with risk and attribution

Attributing the right vulnerability to the right team

Conclusion & Q&A

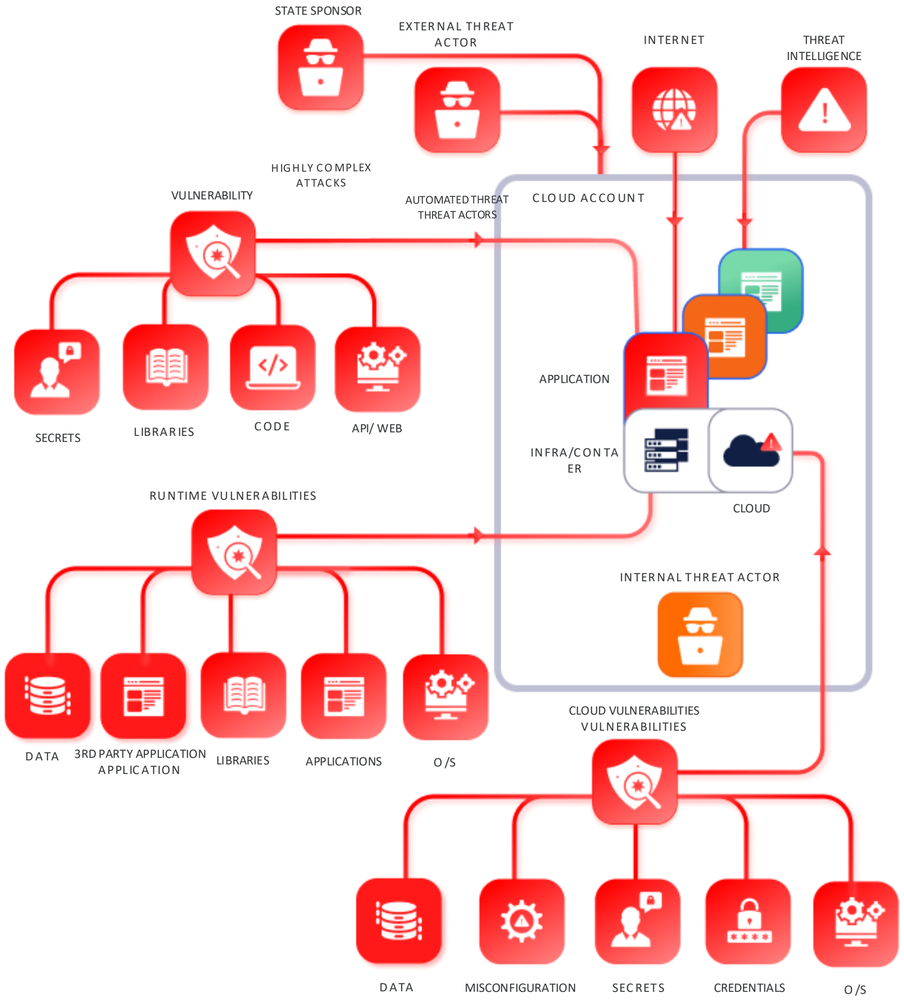
Context: In 2015 we had fewer security tools, digital software supply chain was simpler, and the attack surface was smaller, so finding fixes was trivial

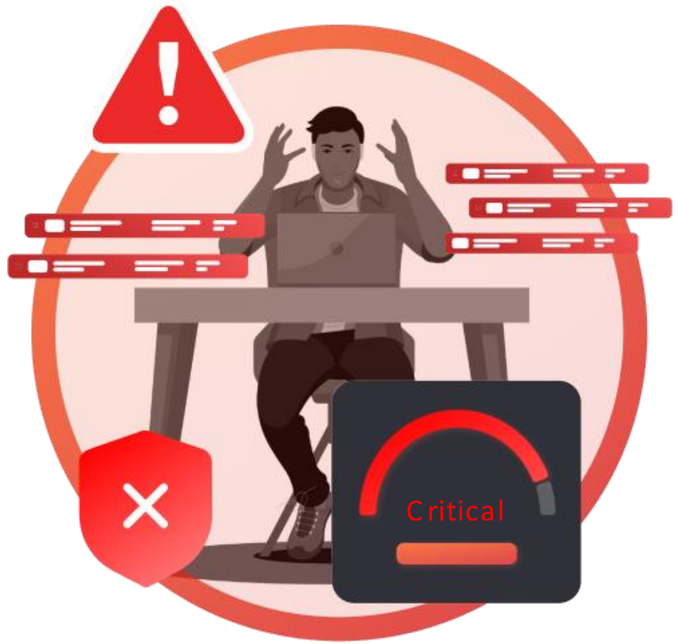




- **Total Number of CVEs:** 15 K (now 222 K+)
- **Few scanners / limited attack surface**
- **Monolithic software deployed on premises**

Context: Today it's becoming impossible to manually find which vulnerability to fix next ... when vulnerabilities are getting exploited in 3 minutes





- **Total Number of CVEs Increasing exponentially:**
  280 K (vs 6.7k in 2015)
  40K vuln last year
- **Multiple alerts all disconnected, multiple disjointed processes and reports**
- **<u>Larger</u> software attack surface built by multiple teams releasing frequently**

I feel your pain

Alert FATIGUE
No COMMON LANGUAGE
=
BURNOUT

Vulnerability growth outpaces the ability of defender to react. Automation is the only solution

#CVE
220,538 **

35% YoY increase

Most Vulnerabilities are Critical - High (58%)**

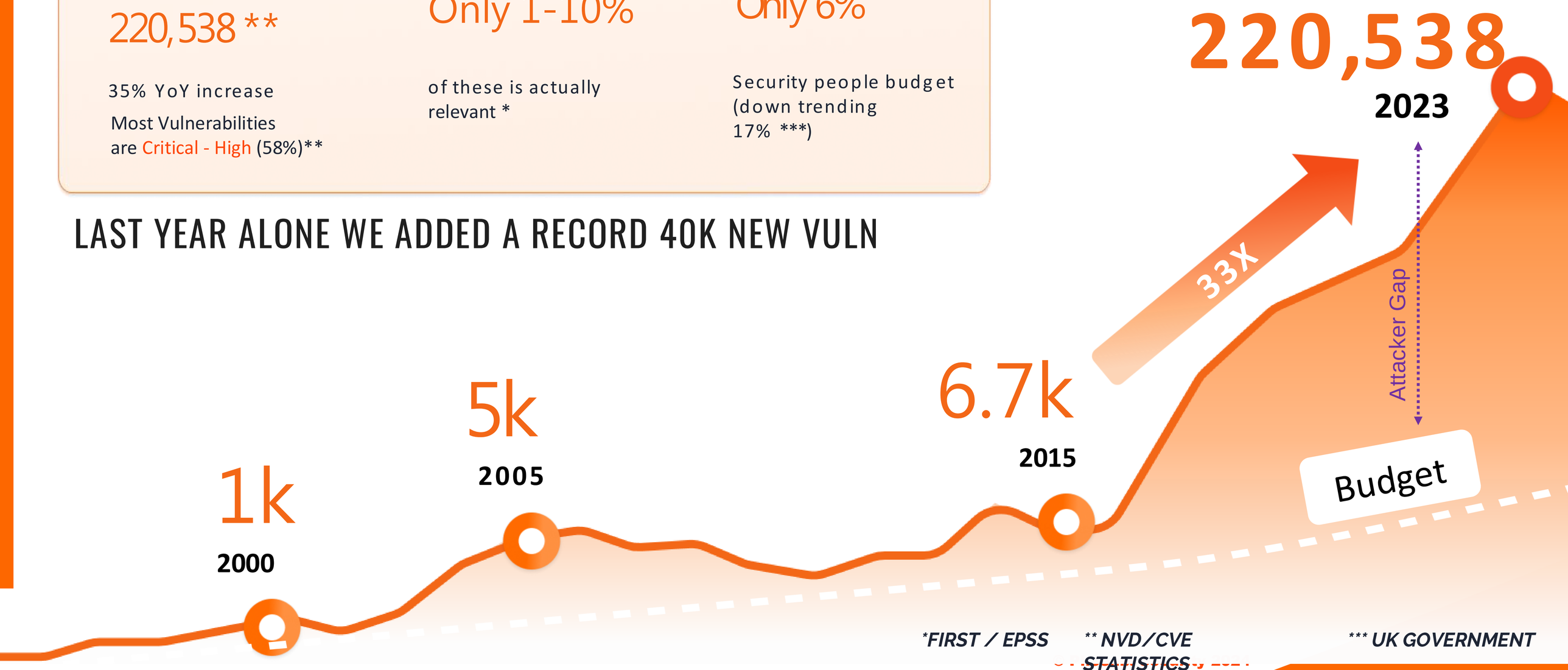
Only 1-10%

of these is actually relevant *

Only 6%

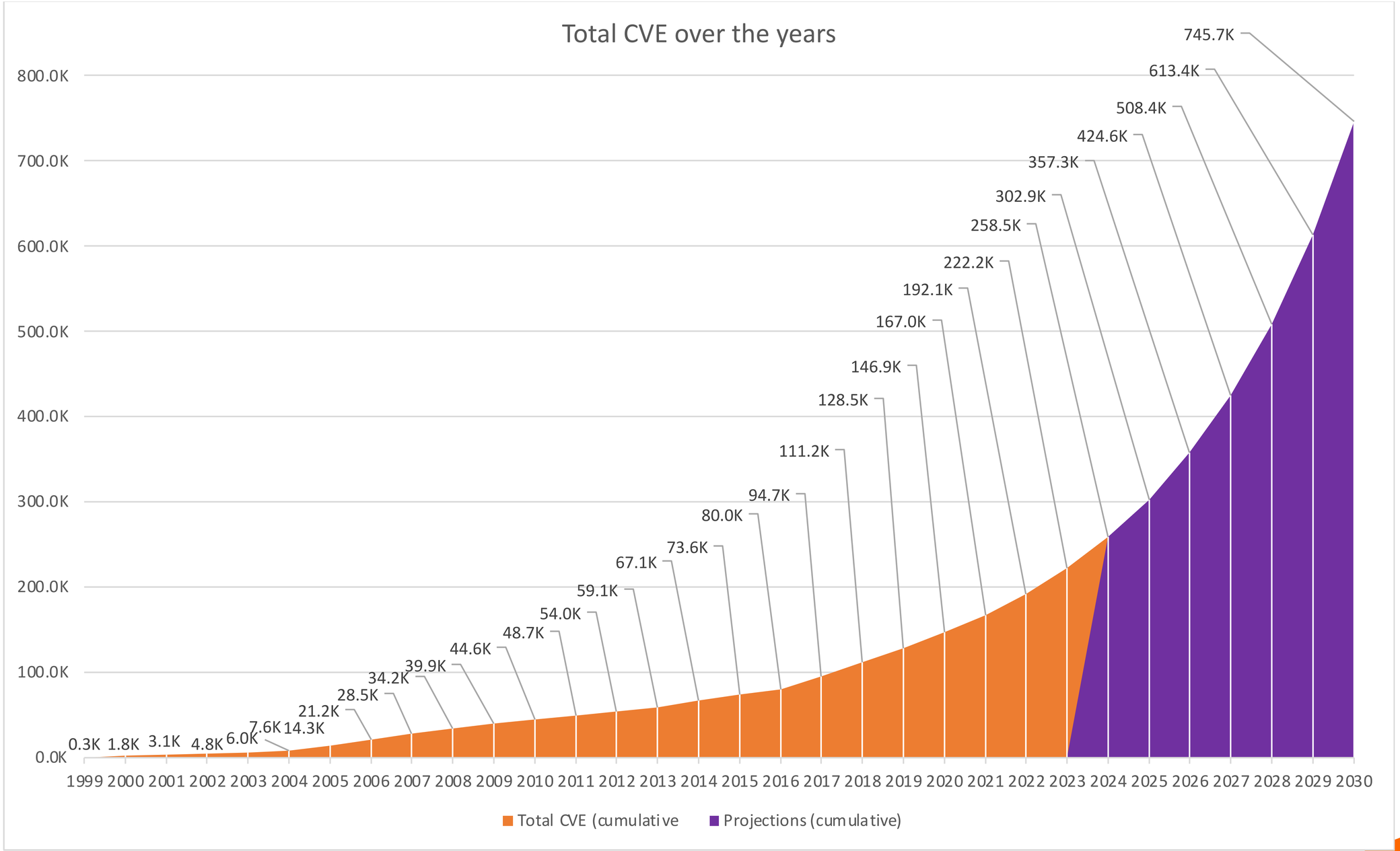
Security people budget (down trending 17% ***)

LAST YEAR ALONE WE ADDED A RECORD 40K NEW VULN



*FIRST / EPSS ** NVD/CVE STATISTICS *** UK GOVERNMENT

The Race to a Million vulnerabilities...not that far away



Market – More code than ever, malicious code generator accelerate exploitation time to 3 minutes

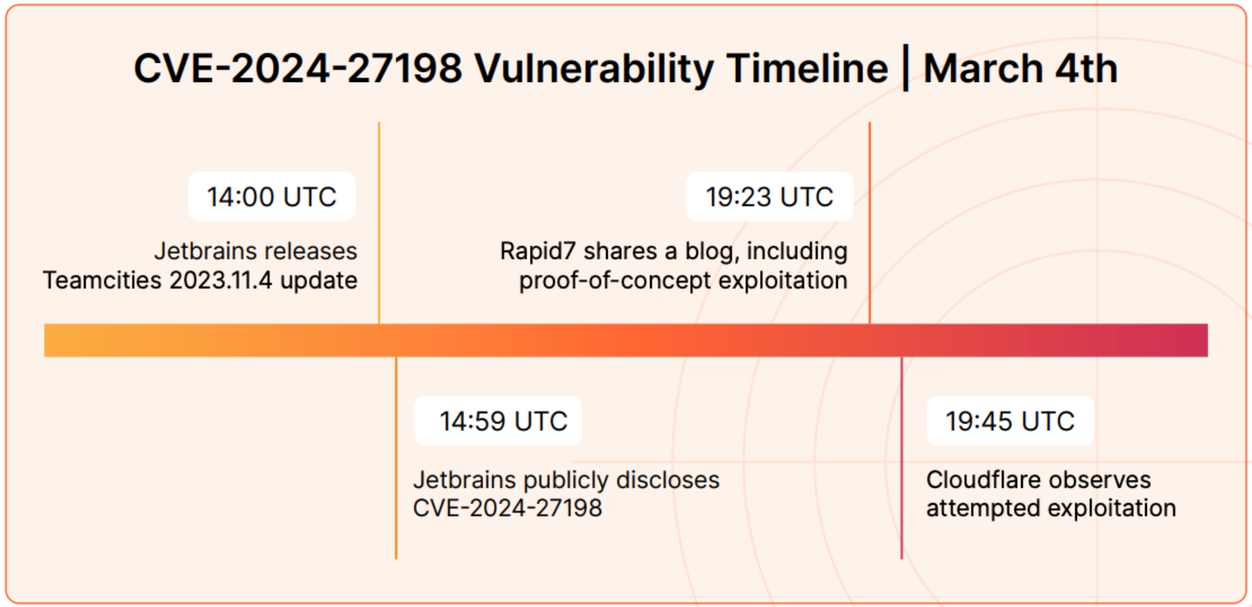
> Data from GitHub reveals that "41% of all code right now is AI generated," Mostaque remarked. More interestingly,
>
> GitHub CTO

State of Malicious underground LLM to develop malicious code*

Table 1: *Malla* services and details

Name	Price	Functionality			w/wo voucher copy	Infrastructure
		Malicious code	Phishing email	Scam site		
CodeGPT [11]	10 βytes*	●	○	◐	No	Jailbreak prompts
MakerGPT [49]	10 βytes*	●	○	◐	No	Jailbreak prompts
FraudGPT [30]	€90/month	●	●	●	No	-
WormGPT [79, 80, 83]	€109/month	●	●	◐	No	-
XXXGPT [28, 61, 84]	$90/month	●	○	○	Yes	Jailbreak prompts
WolfGPT [77, 78]	$150	●	●	●	No	Uncensored LLM
Evil-GPT [26]	$10	●	●	●	No	Uncensored LLM
DarkBERT [16, 17]	$90/month	●	●	○	No	-
DarkBARD [14, 15]	$80/month	◐	◐	○	No	-
BadGPT [2, 3]	$120/month	◐	◐	◐	No	Censored LLM
BLACKHATGPT [4–6]	$199/month	●	○	○	No	-
EscapeGPT [23]	$64.98/month	●	◐	◐	No	Uncensored LLM
FreedomGPT [32, 33]	$10/100 messages	●	◐	◐	Yes	Uncensored LLM
DarkGPT [18, 19]	$0.78/50 messages	●	◐	◐	Yes	Uncensored LLM

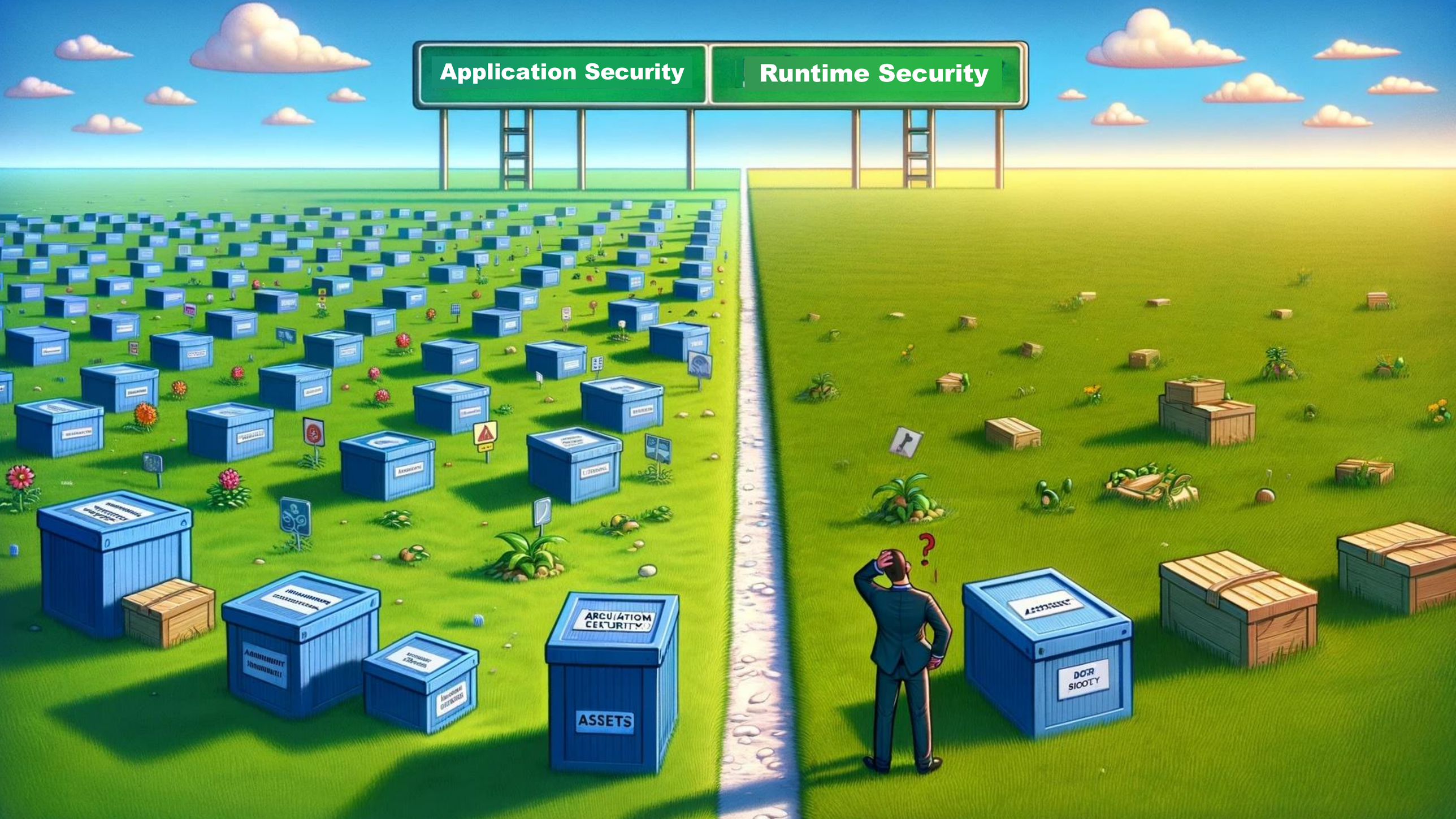
* βytes is the forum token of `hackforums.net`; ◐ indicates implicit mention.

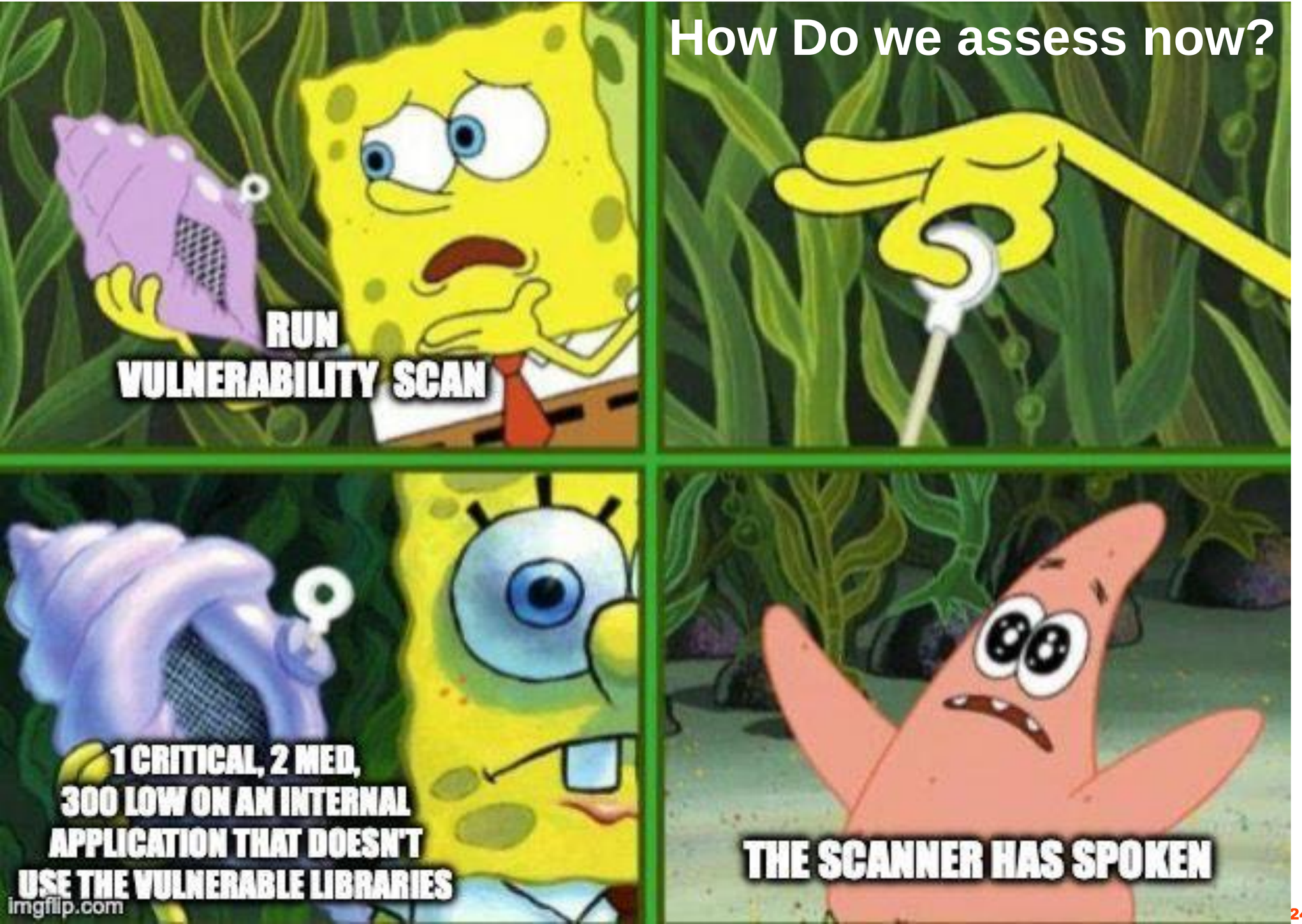


← 3 Minutes** →

*https://arxiv.org/abs/2401.03315

**https://blog.cloudflare.com/application-security-report-2024-update/


Application Security
Runtime Security
ASSETS


How Do we assess now?
RUN VULNERABILITY SCAN
1 CRITICAL, 2 MED, 300 LOW ON AN INTERNAL APPLICATION THAT DOESN'T USE THE VULNERABLE LIBRARIES
THE SCANNER HAS SPOKEN
imgflip.com

The Vulnerability Cycle

Step 1 – Overload Dev

Step 2 – Pray they catch that 1 vulnerability

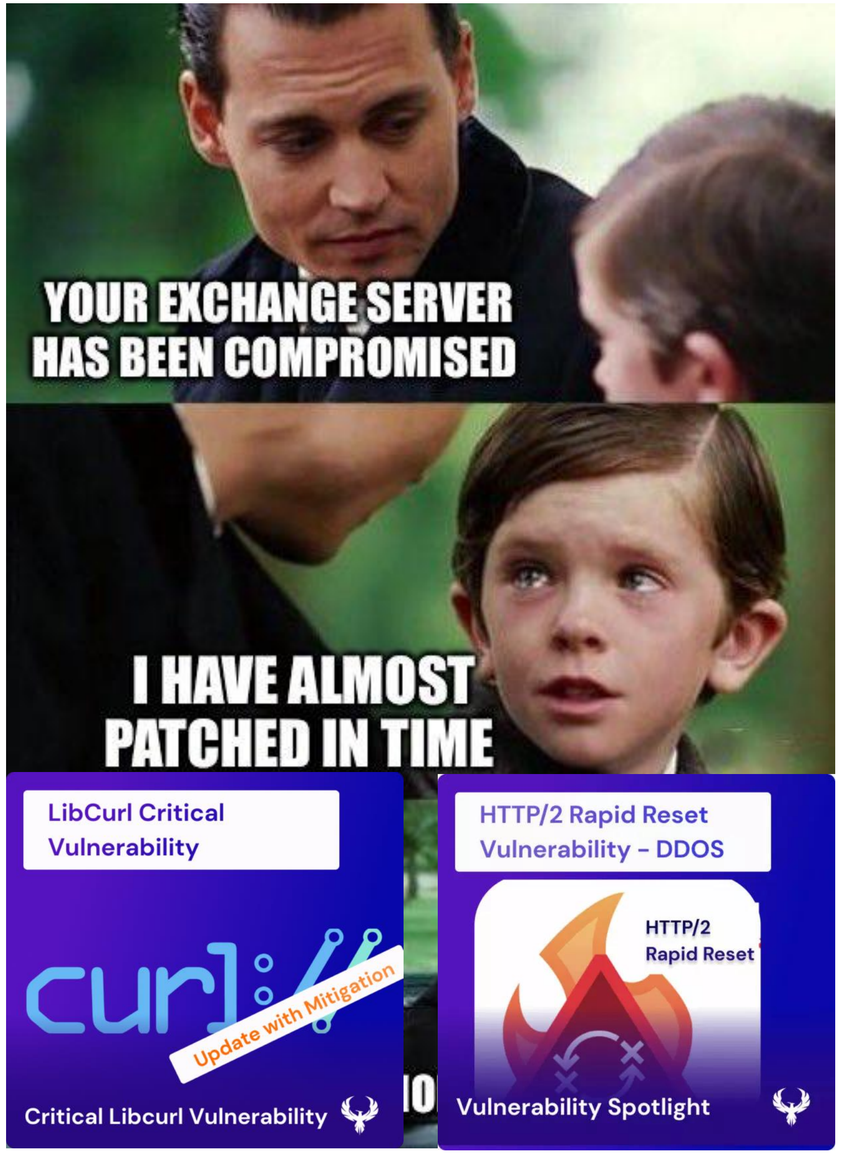
Step 3 – That 1 vulnerability get compromised

Step 4 – Shocked Executive, we asked security to be secure

Step 5 – Overload Team some more with latest buzzword scanner

Bonus – Executive mention do security > Security replies fix with SLA

How do we address this problem



The question we try to answer NOW

HOW MANY problems have we addressed and how quickly

Questions we should be answering

WHO does WHAT where and how IMPORTANT is it

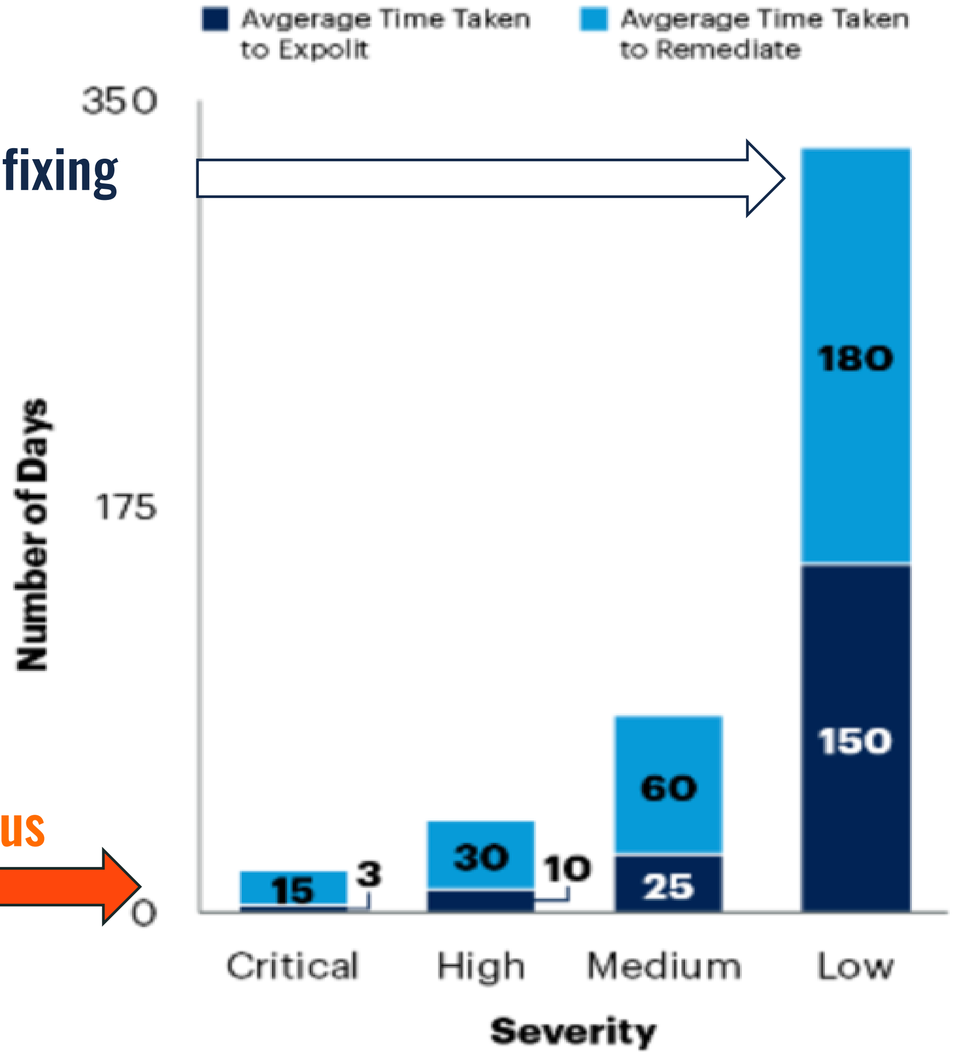
But really... is it raceable

Part 1 - Identify what to fix first IS COMPLEX

WE ARE FIXING SLOWER THAN ATTACKERS

How fast are you fixing

Where you should focus

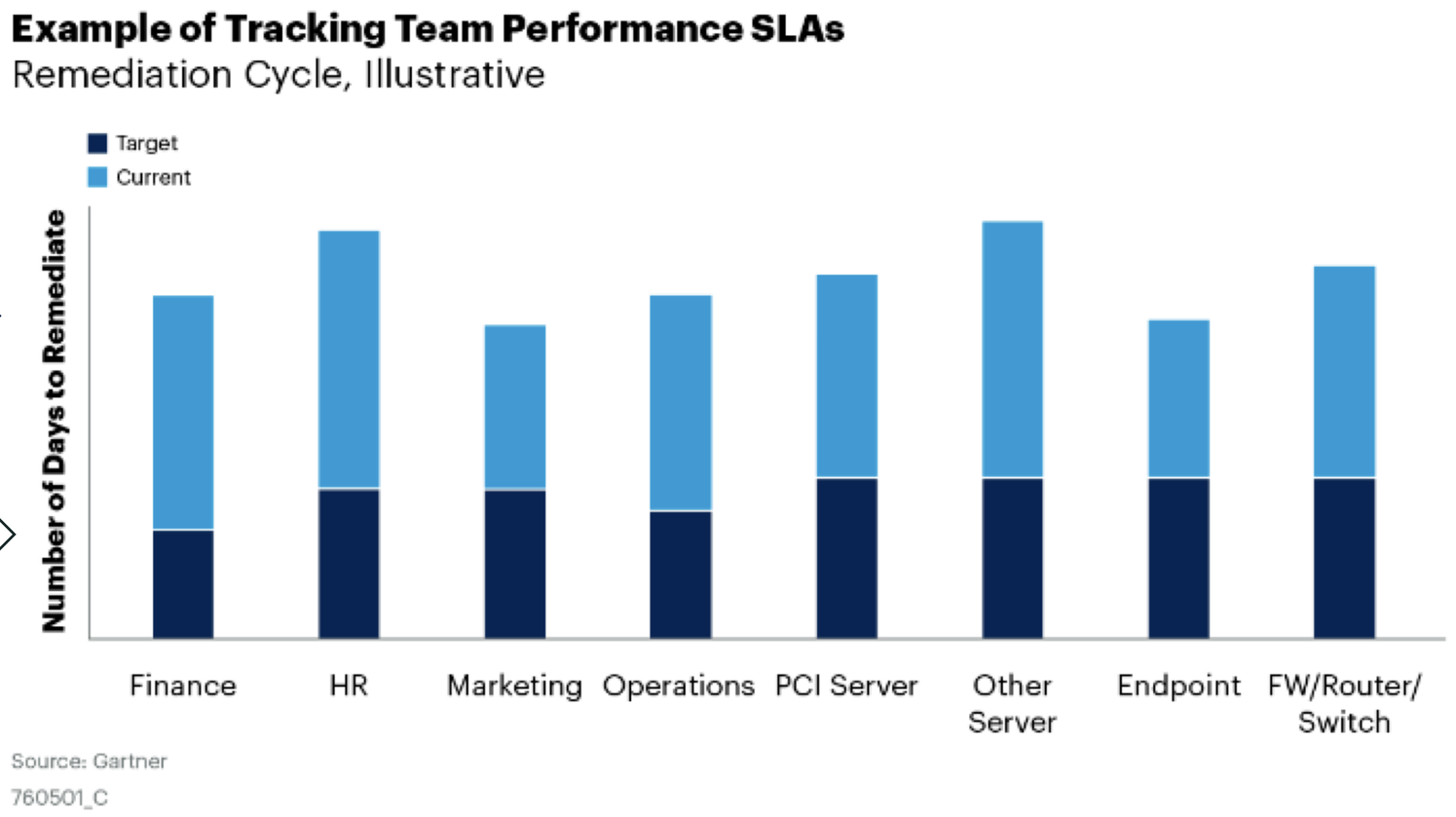


Source: Gartner
760501_C

Using SLA is proven 100% unreachable objective

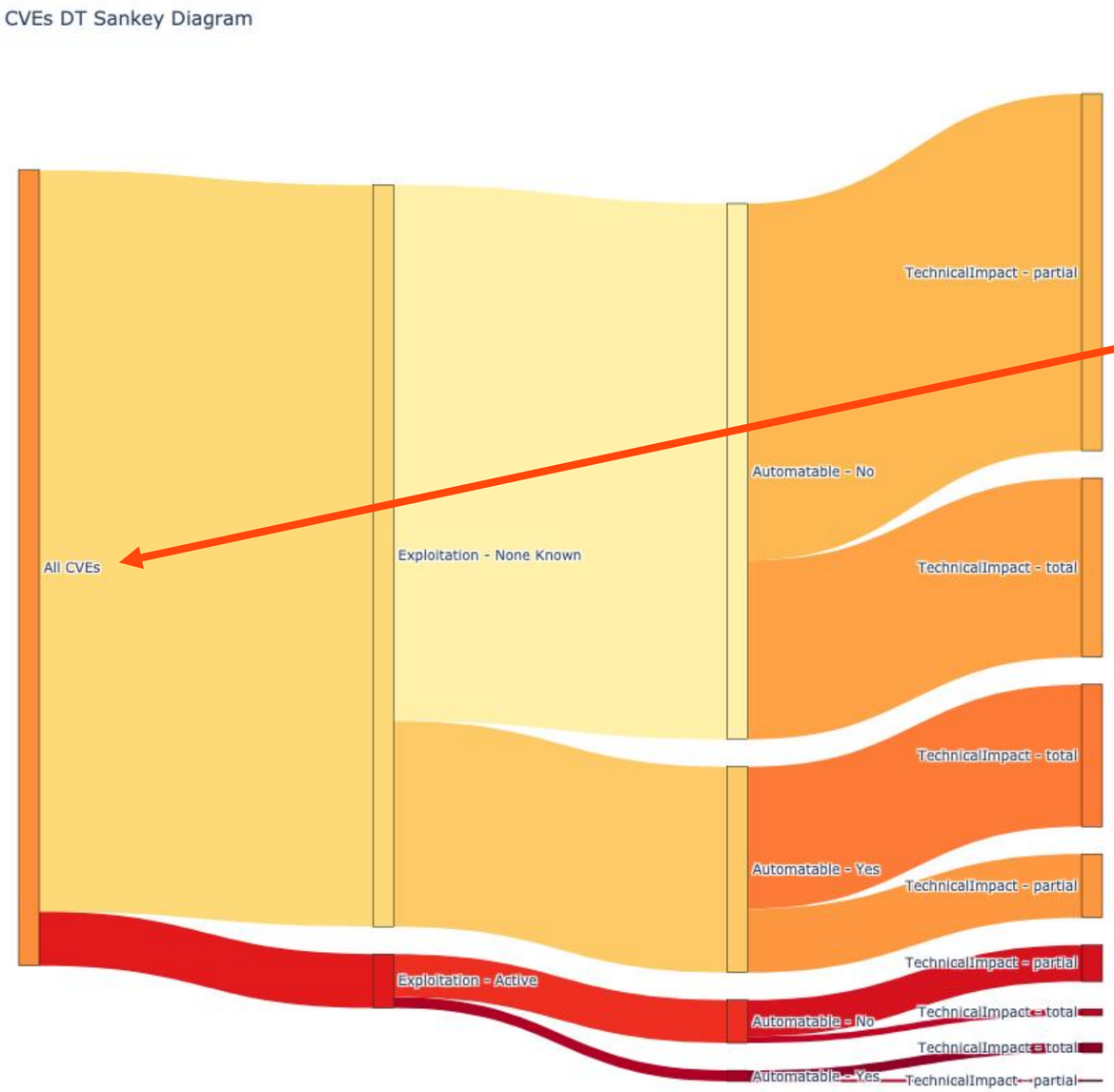
Actual Remedy

Objectives



Gartner

Current Flow of vulnerabilities only 1% are exploitable



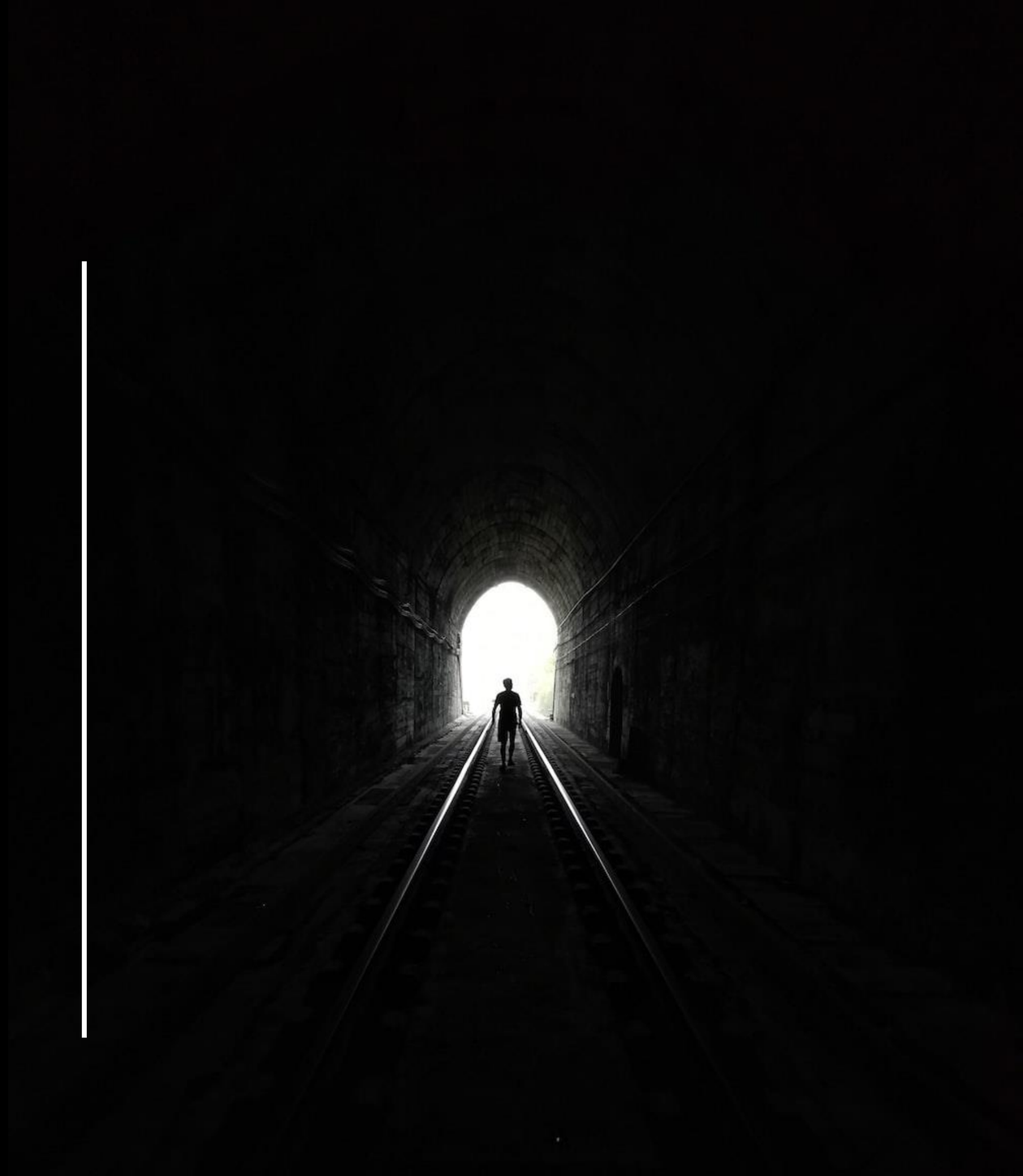
Current Focus

Really important to focus on

All Doom and gloom?

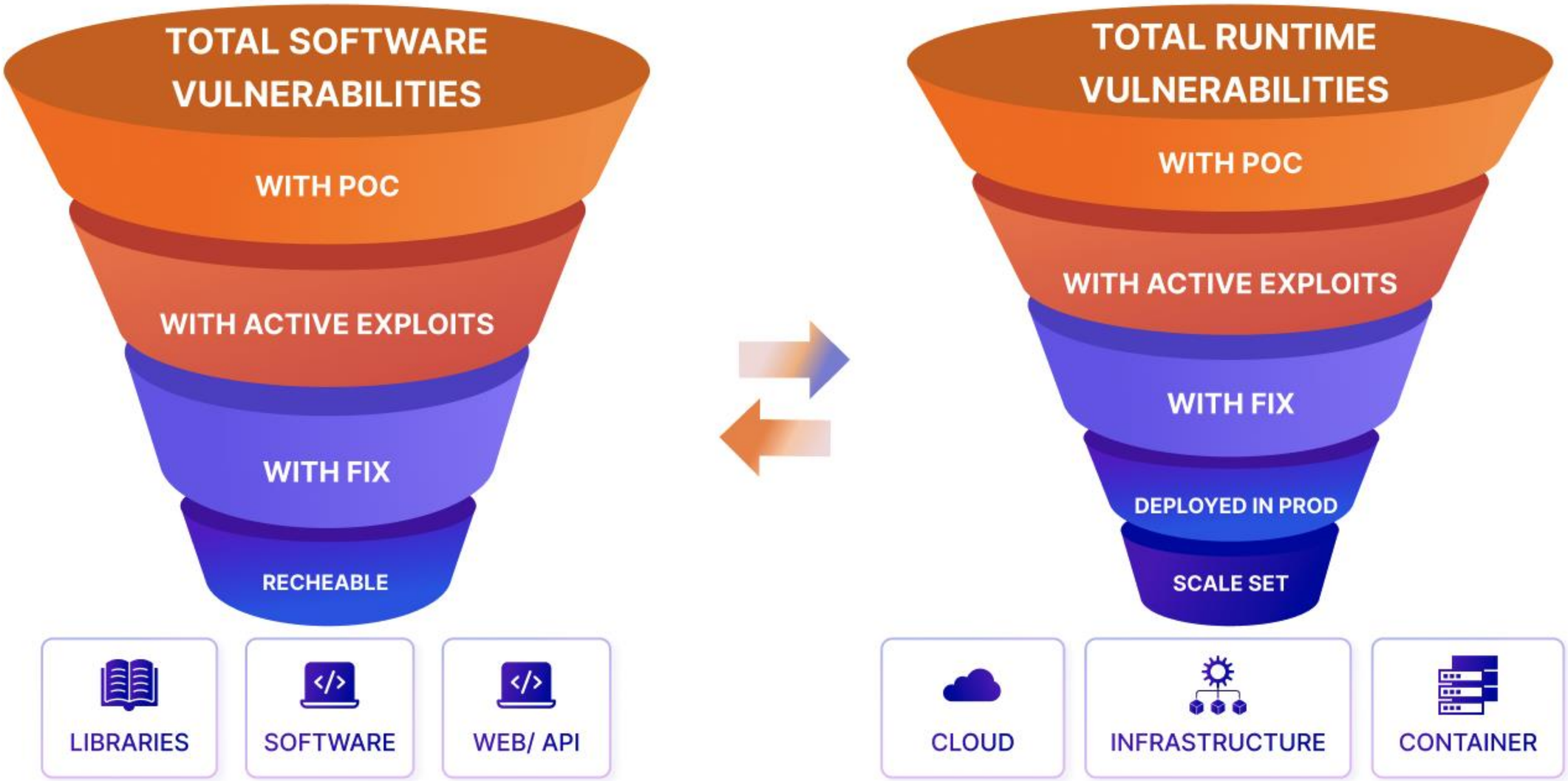
There is a light at the end of the tunnel

- Vulnerability ARE NOT fixed on risk objectives
- Vulnerabilities ARE NOT Prioritized or contextualized
- Vulnerabilities ARE NOT Attributed to the right team
- Asset inventory still a myth, are you aware what software runs in your pipeline

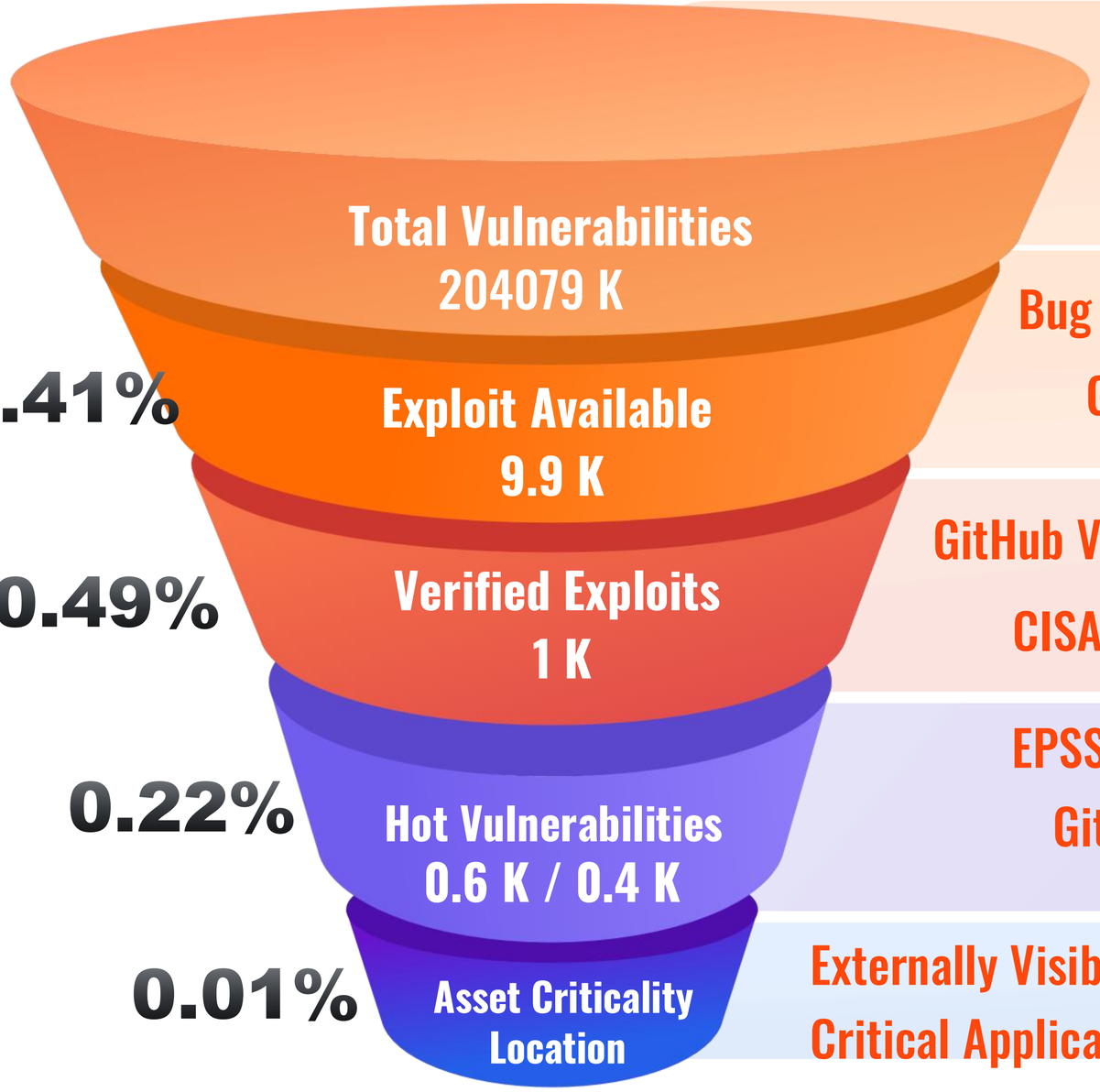


Part 1 - Attributing the right vulnerability with right context

Common Root Cause



Not all the vulnerabilities require equal attention



Bug Bounty Popularity (active 17K) **8.33%**

GitHub Exploit (9.9K) **4.41%**

GitHub Verified Exploits (0.93K) **0.47%**

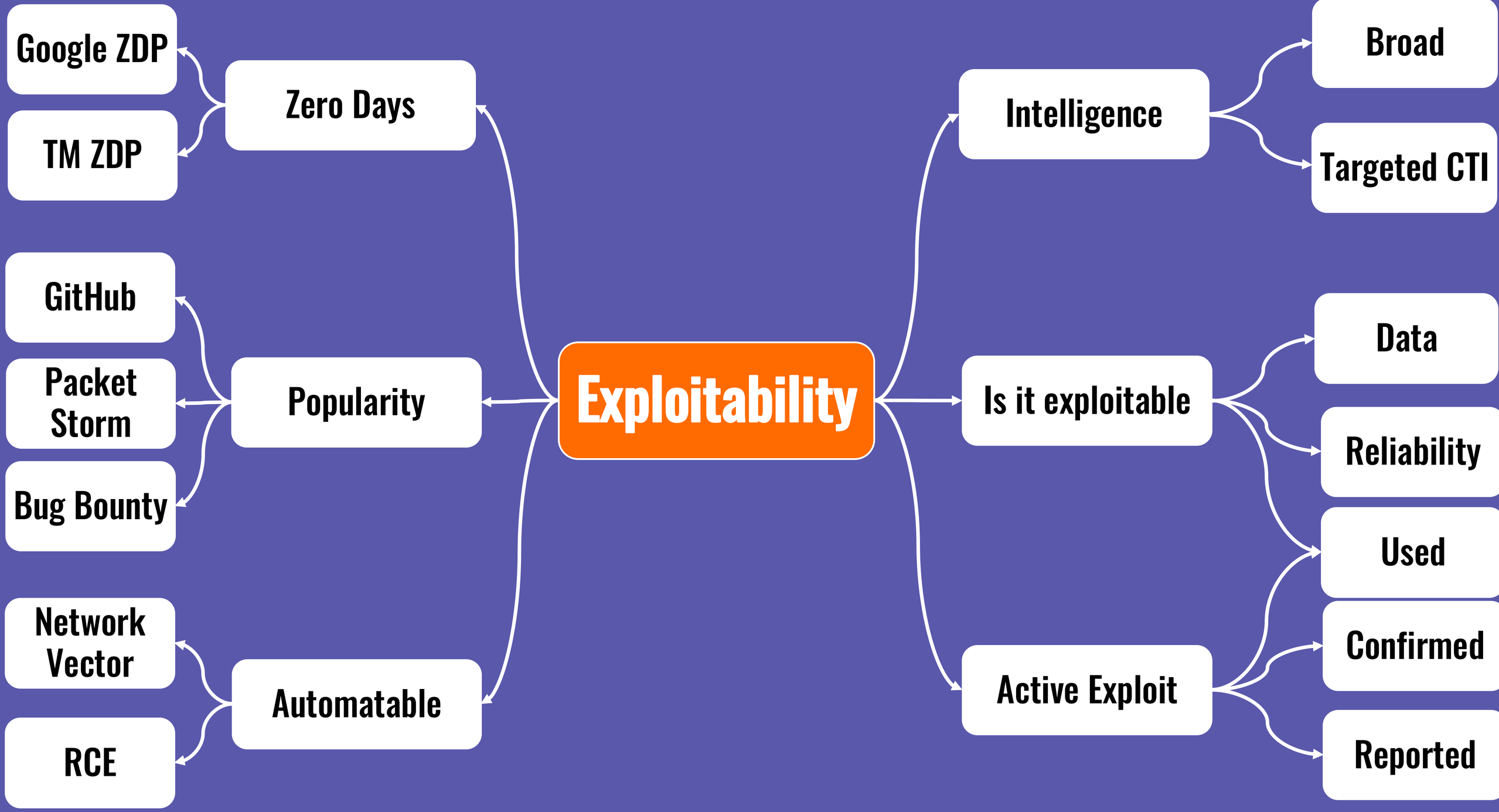
CISA KEV (1K) **0.49%**

EPSS > 0.7 (688) **0.34%**

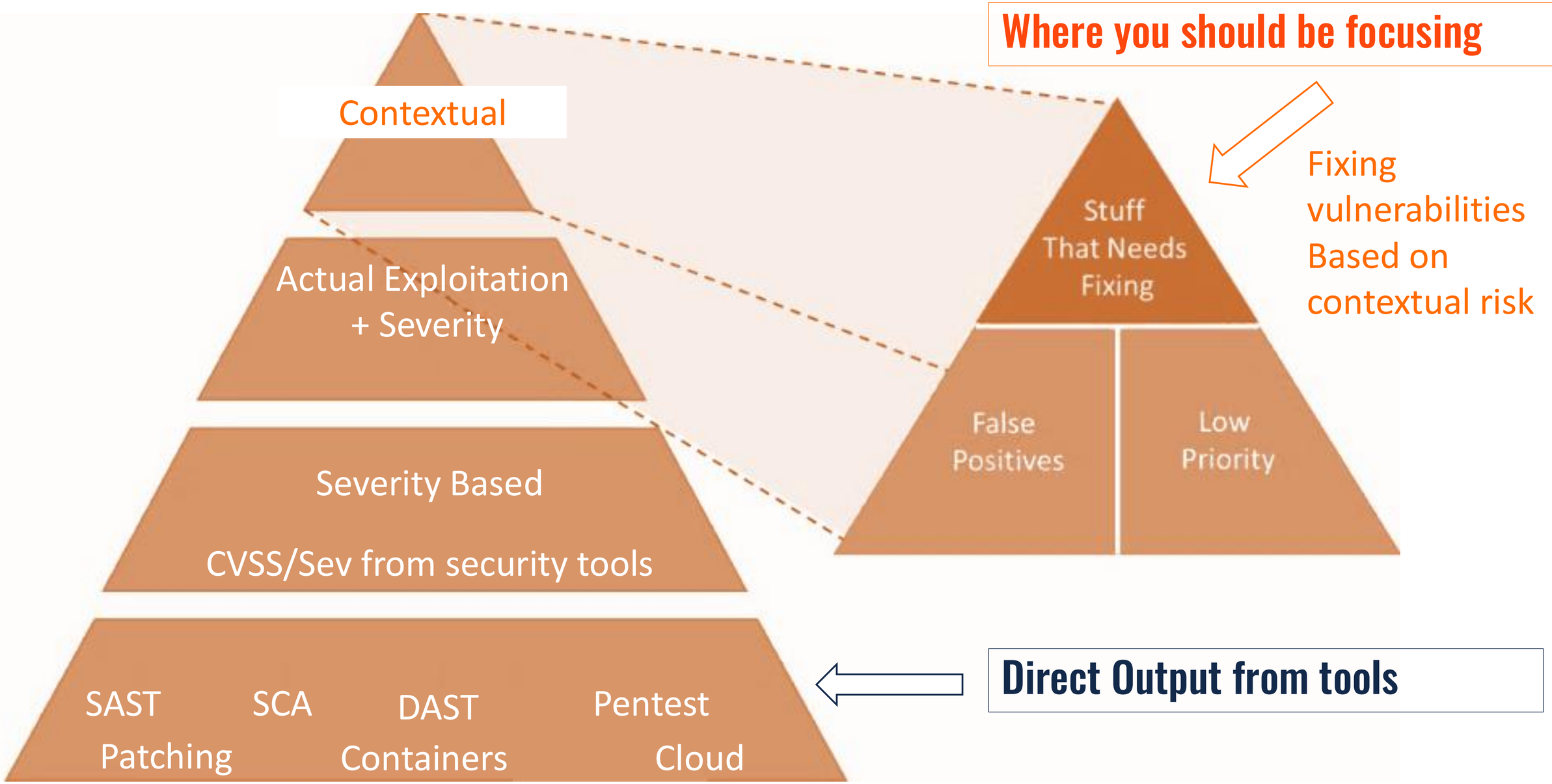
GitHub Active Exploit (0.40K) **0.22%**

Externally Visible (0.14) ? **0.?%**

Critical Application


Google ZDP
TM ZDP
Zero Days
GitHub
Packet Storm
Bug Bounty
Popularity
Network Vector
RCE
Automatable
Exploitability
Intelligence
Broad
Targeted CTI
Is it exploitable
Data
Reliability
Active Exploit
Used
Confirmed
Reported

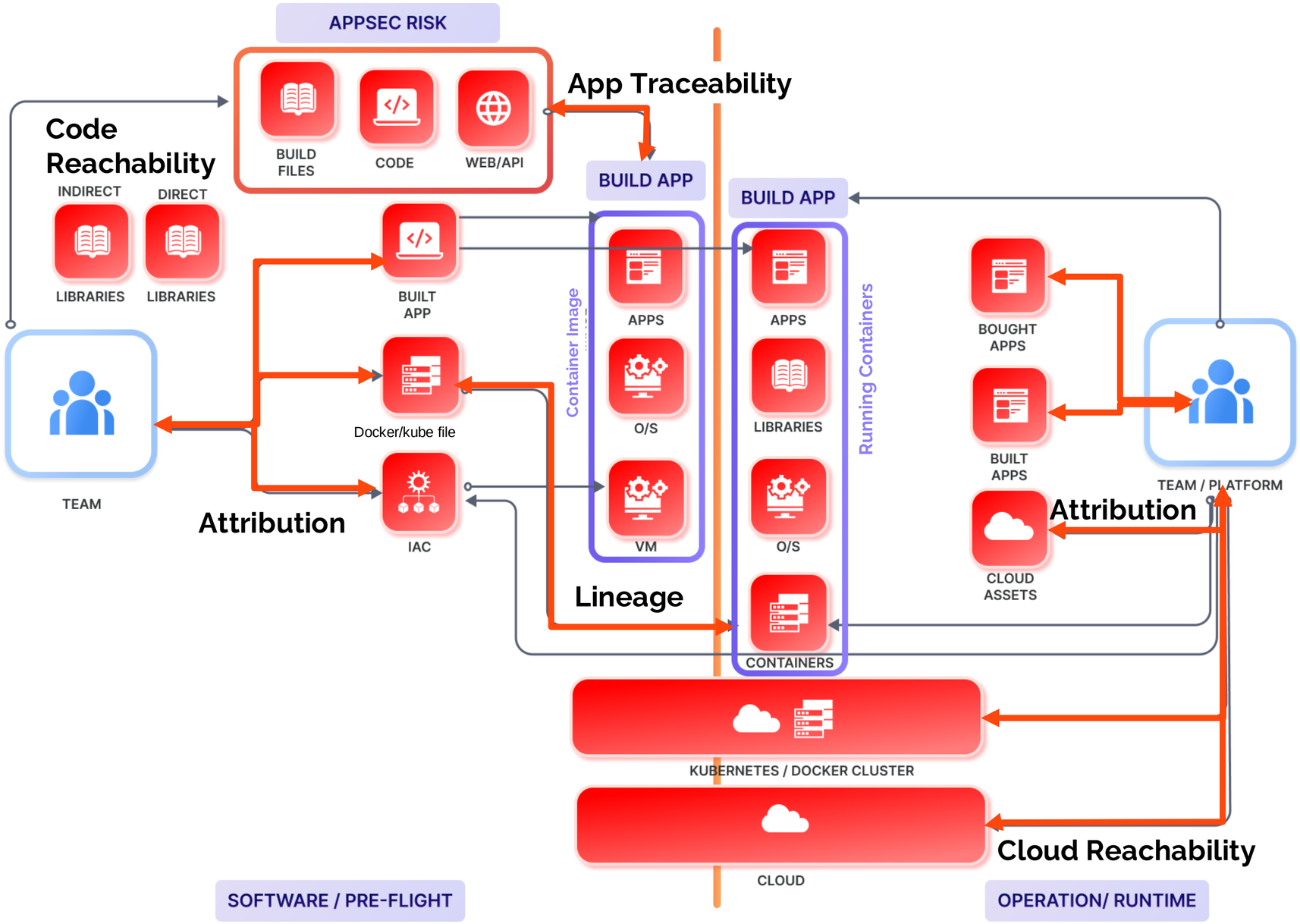
Prioritization is so 90....



Part 2 - Reachability and what problem solve

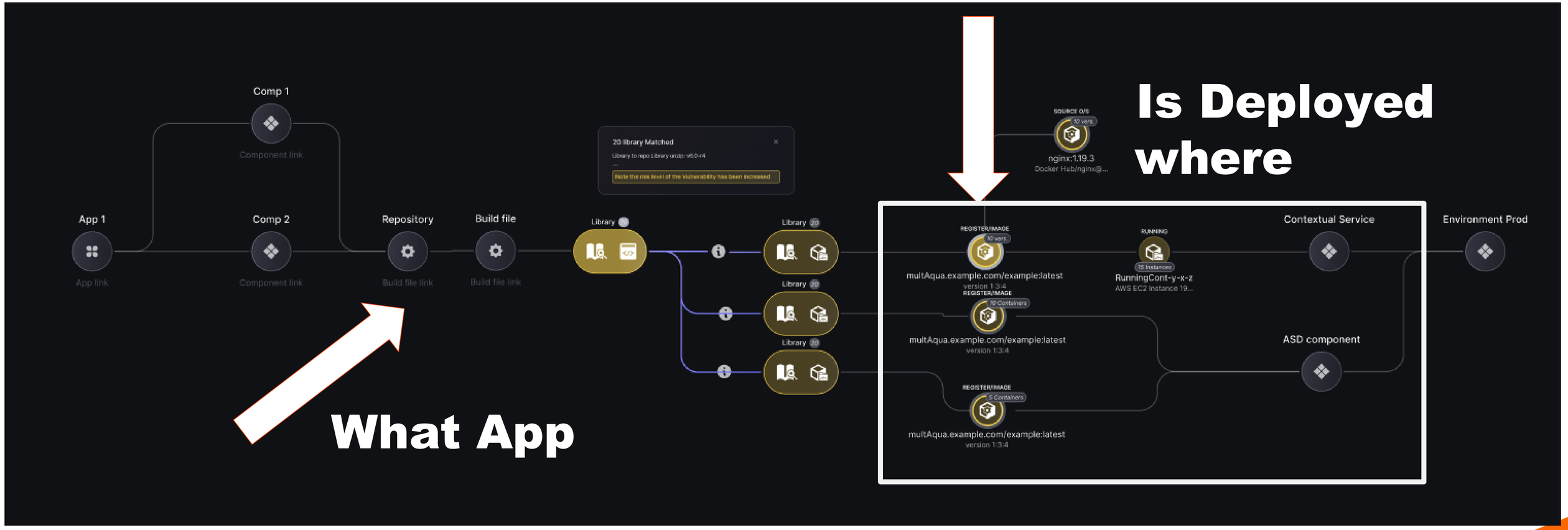
Phoenix correlates, contextualizes and deduplicates by linking together assets using 4 dimensions

- Attribution
- Lineage
- Traceability
- Code/Cloud Reachability



Container Lineage to Complete the pictures

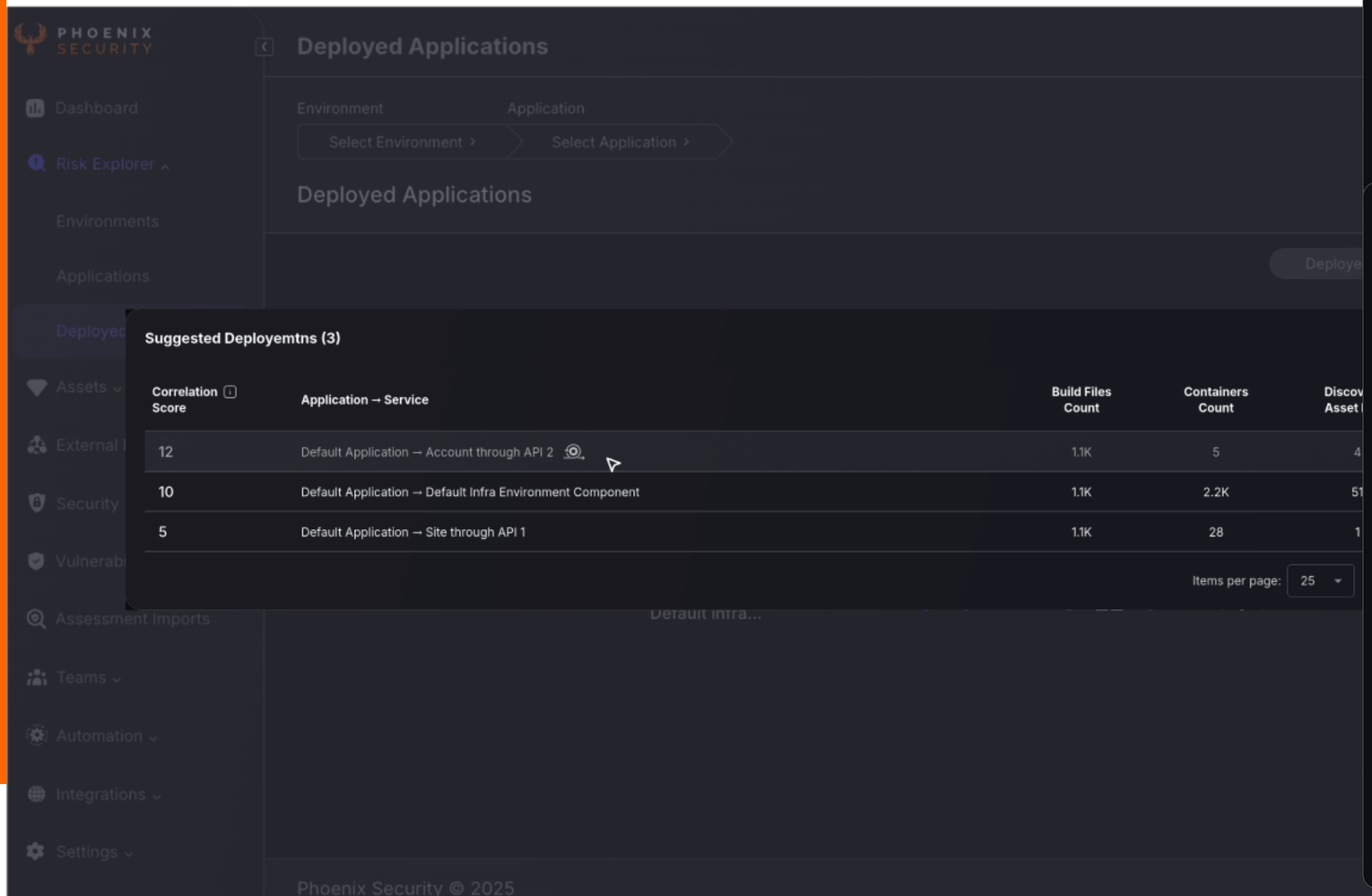
Libraries that are deployed : fix in the library

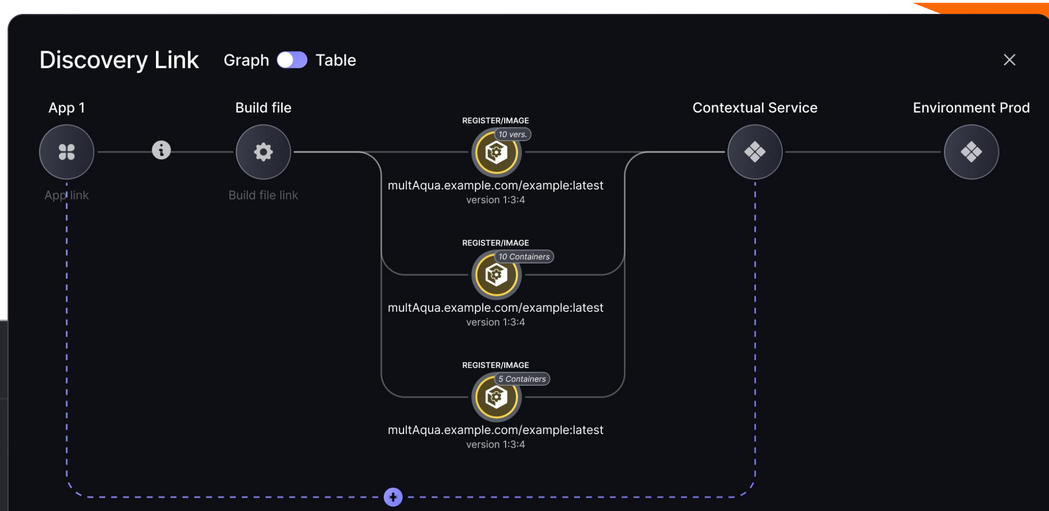


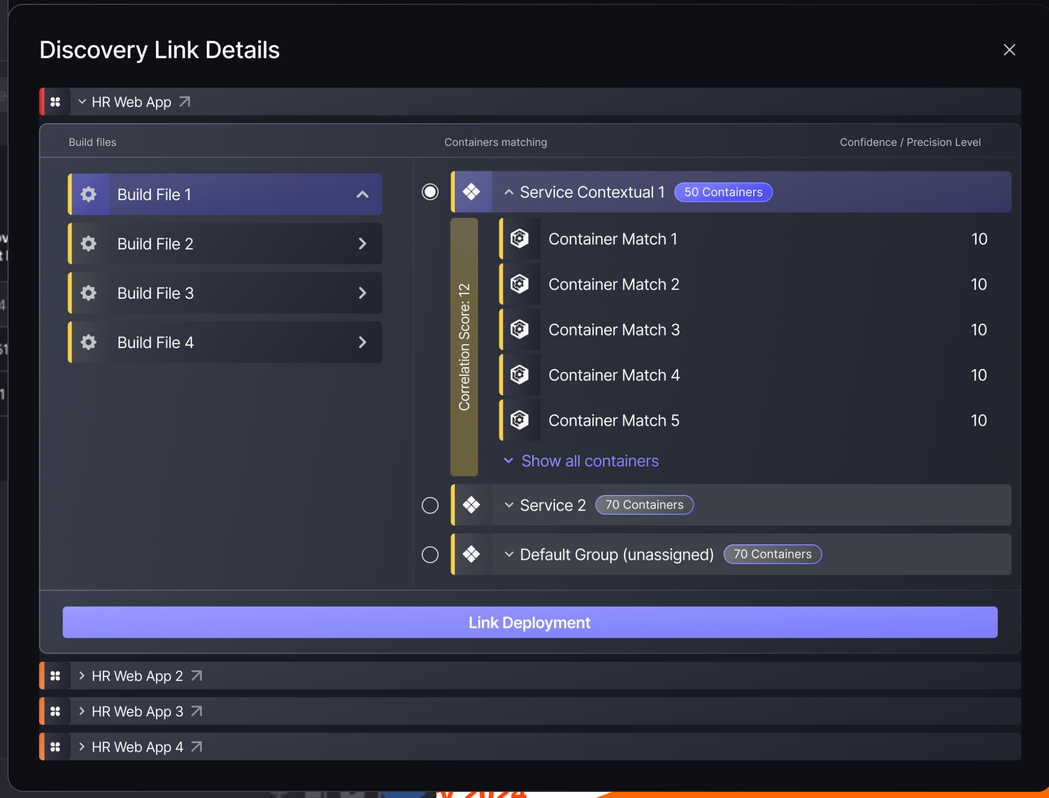
Container Deployment Discovering which app is running where

Which Application is running Where ?

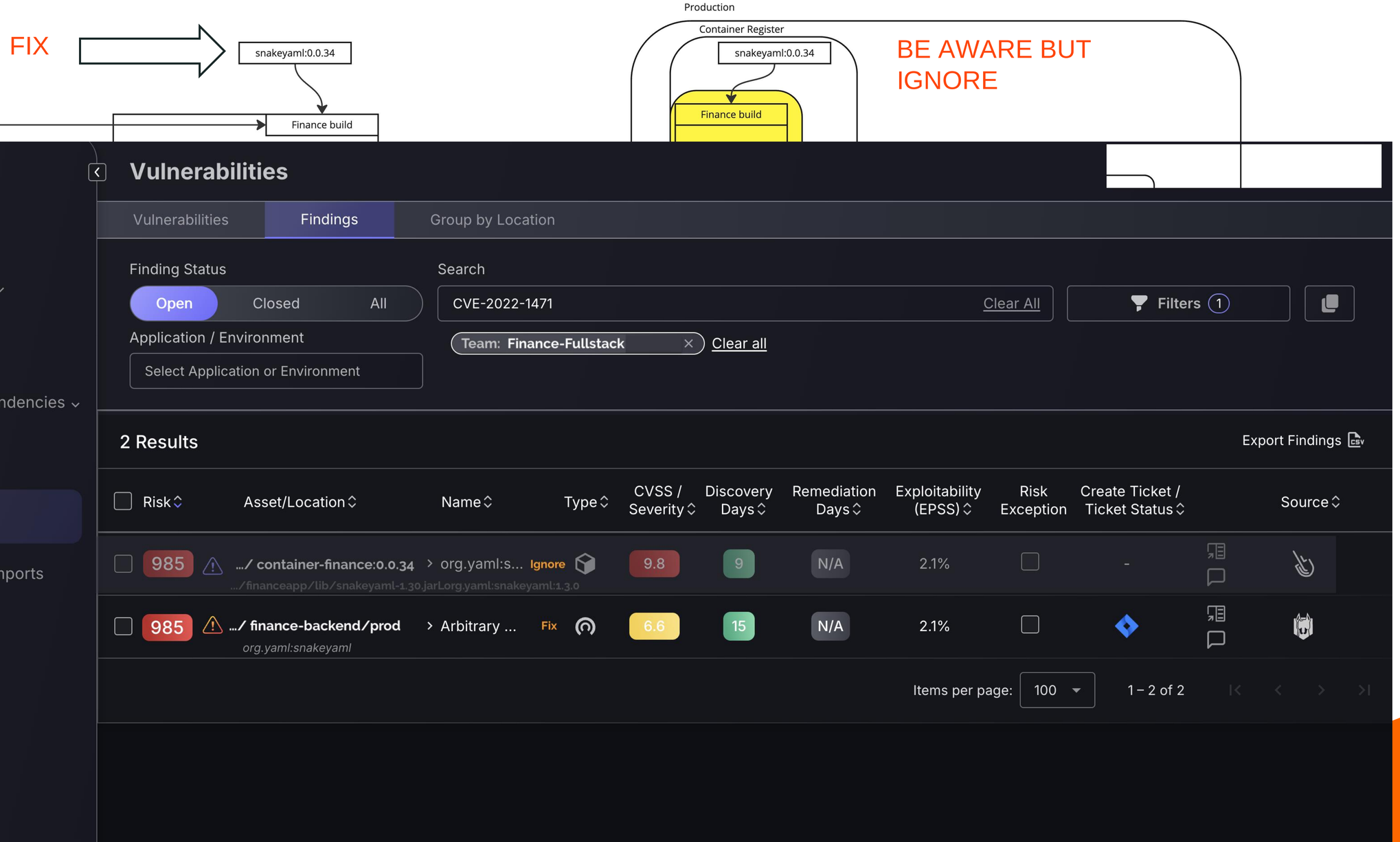
- Create Automatically groups based on deployment patterns
- Use tags or Tracing based on profile deployment



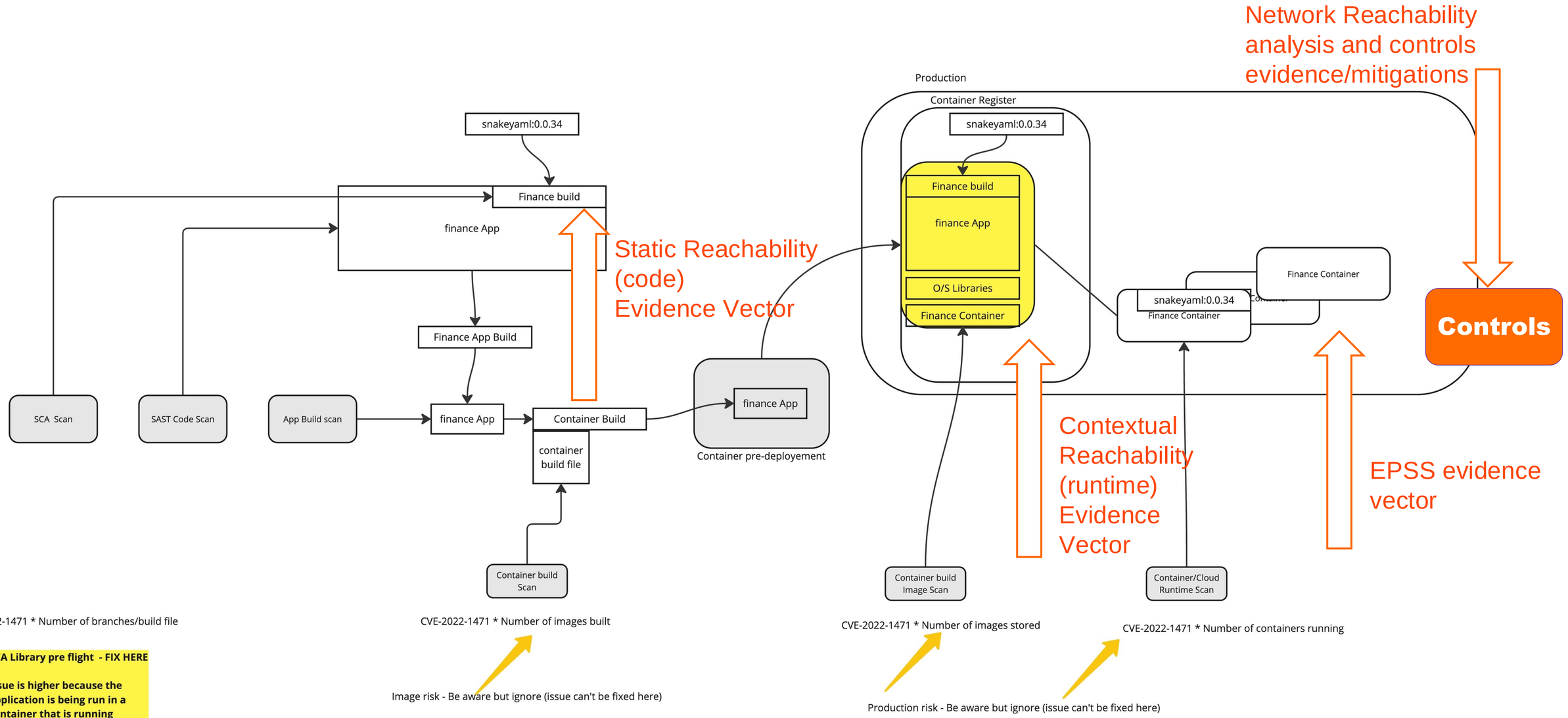




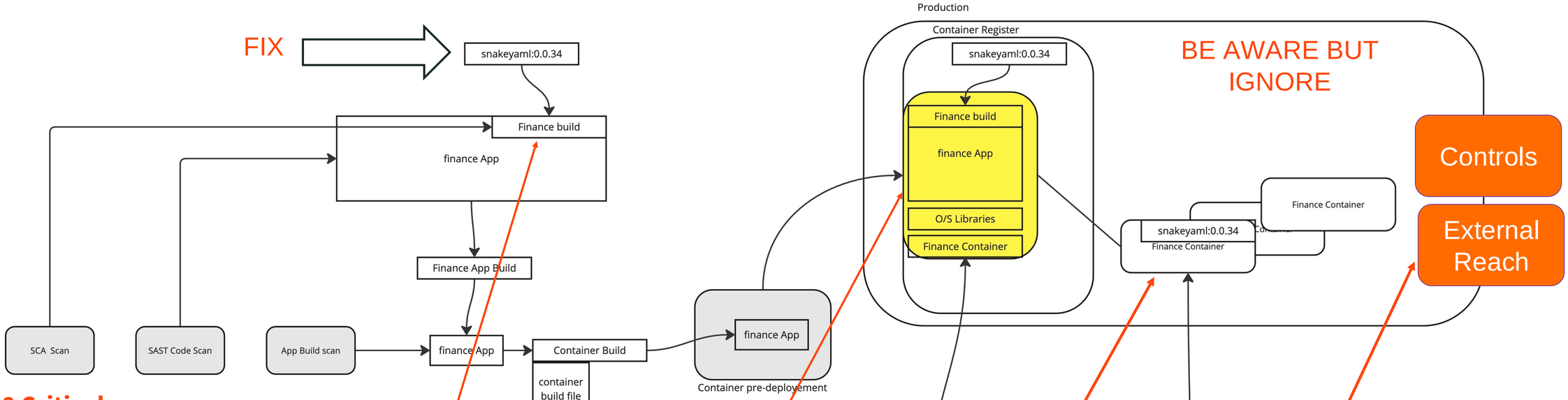
Real Case Scenario : Deduplicating Contextually Code and Libraries



Real Case Scenario : EPSS vs Static Reachability vs Runtime - contextual Reachability



Real Case Scenario : Deduplicating Contextually Code and Libraries



Sysdig: 9.8 Critical

Snyk: 6.6 Medium

830

CVSS - Base Score: 9.8 High

CVSS - Base Score: 8.3 High

CVE-2022-1471 * Number of branches/build file

CVE-2022-1471 * Number of images built

CVE-2022-1471 * Number of images stored

CVE-2022-1471 * Number of containers running

SCA Library pre flight - FIX HERE

Issue is higher because the application is being run in a container that is running multiple times

Image risk - Be aware but ignore (issue can't be fixed here)

Production risk - Be aware but ignore (issue can't be fixed here)

CVSS - Base Score: 8.3 High

700

If not statically reachable -> 700

If network raceable 9 (loaded in a container)

900

660

!

EPSS - 2% 0.021010000

Exploit available

RCE

980

If externally facing

9.8 / 980

Reachability Analysis of a Vulnerability

	1 Code Reach Reach Analysis	2 Runtime Reach Analysis	2 Container Reach Analysis	4 Network Reach Analysis	3 CTI	3 CTI - Exploitability
WHAT	Analyze function or library being created	Test if library being in container	Detect if the container is being being loaded	Verify if a container's library/node reachable	Like EPSS identify if a vulnerability is is being exploited	Exploit evidence of vulnerability
WHEN WERE	Code, Repo, Build	Runtime/ Build	Cluster analysis of container	Cloud/ Operation	Everywhere	Everywhere
BENEFITS	Reduce vulnerabilities in vulnerabilities in lib/function not used	Helps identify if the code is being loaded container, and which container	Image of the container is being being used in runtime	Helps identify if the the vulnerability can be reached from Remote	Prioritization of vulnerabilities based on exploitation in	Prioritization of vulnerabilities on exploit
LIMITATIONS	Complex and per language	More intrusive and intensive in some instances Might require Pipeline integration	Requires connection to container	Cloud/ Network reachability analysis	Only works for network detectable Exploits	Base indicator If there is an exploit in the wild

Part 2 (cont) - Container / Container Images / Running Containers

The question we try to answer NOW

How do we fix / patch containers

Questions we should be answering

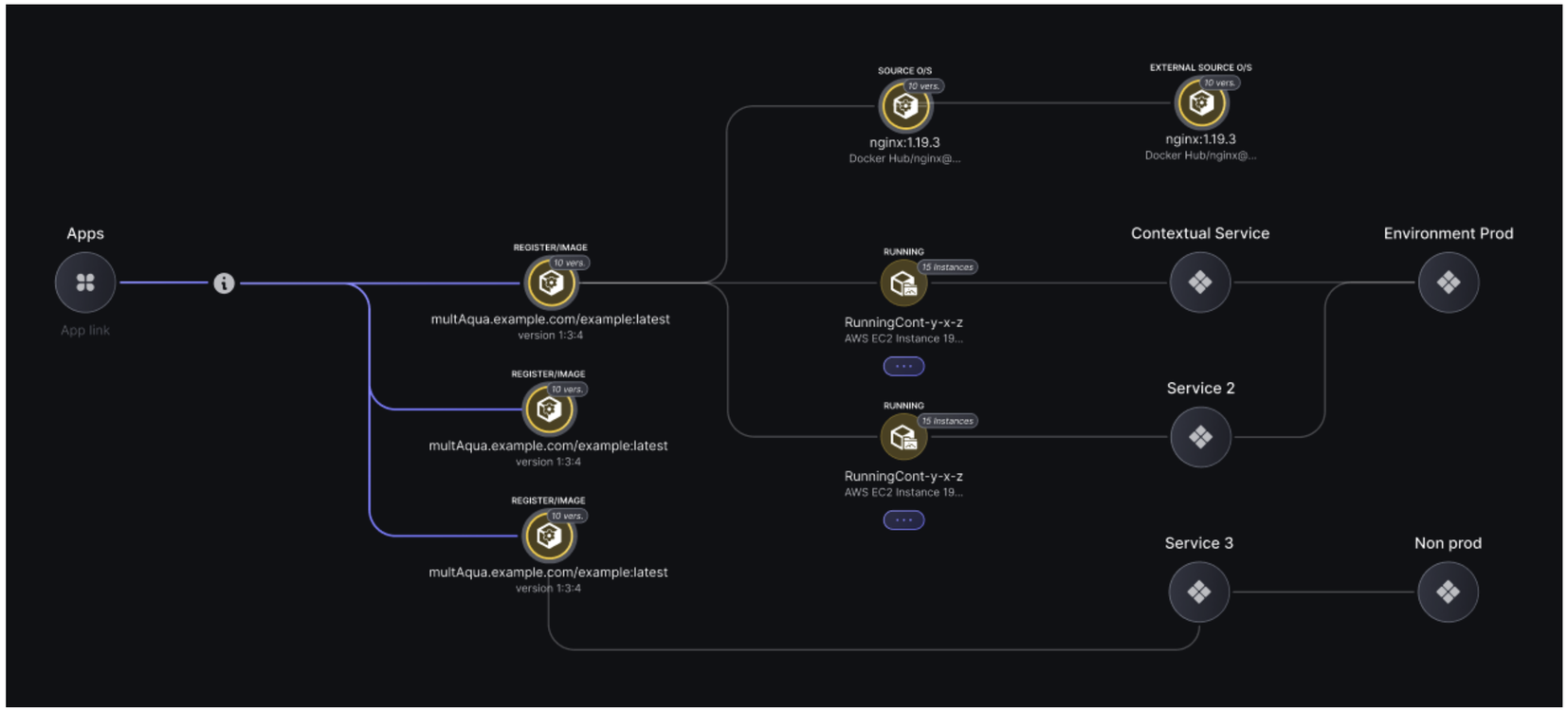
- Which container is active
- What container image is running
- Who owns it
- Do I fix it in code or in container base image?

But really is it raceable

Container Lineage – Where container come from How many running containers

Which Application is running Where ?

- Create Automatically groups based on deployment patterns
- Suggest application grouping



Asset Details

External

Asset Name
prn:foss:github:null

Asset ID
prn:foss:github:null

Asset Type
Container

Teams
external Team, Team Orphan Infra

Environments
INST Prod

Services
container active

Dockerfile
Security-Phoenix-demo1/exploit-CVE-2024-3094:Dockerfile

Resource Type
Container

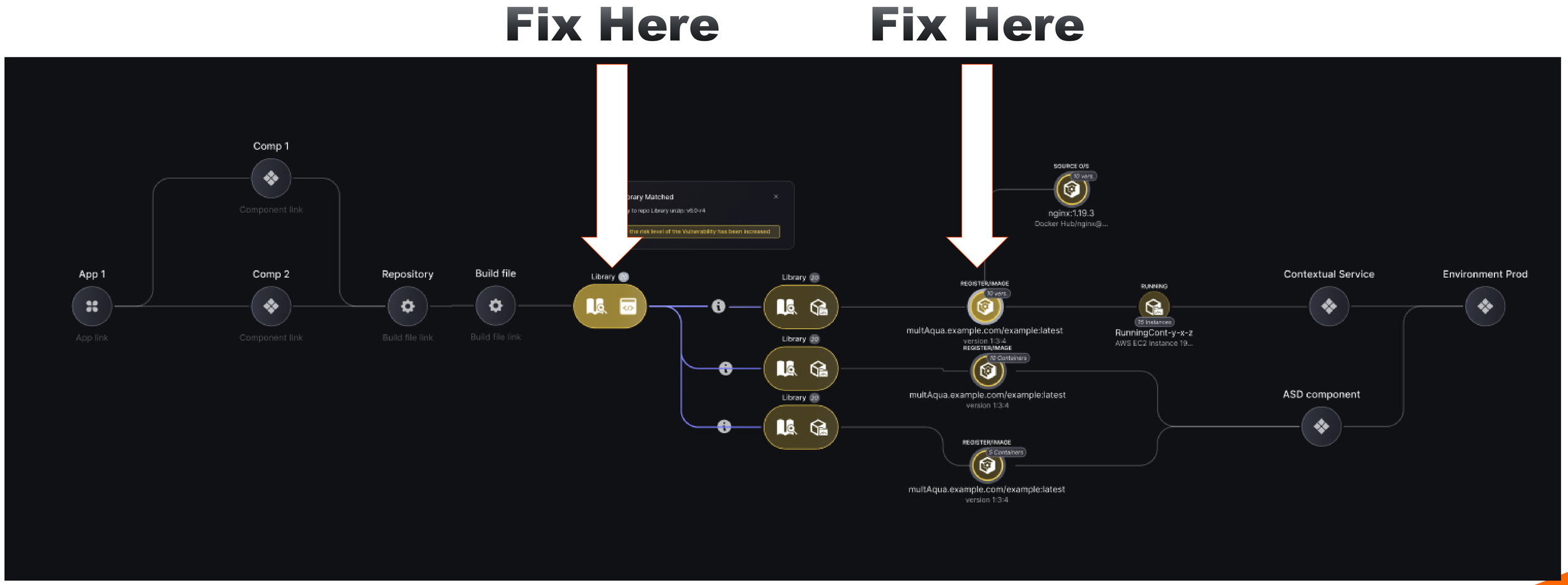
Tags
deployment:123 active v1.3

Sources

Import

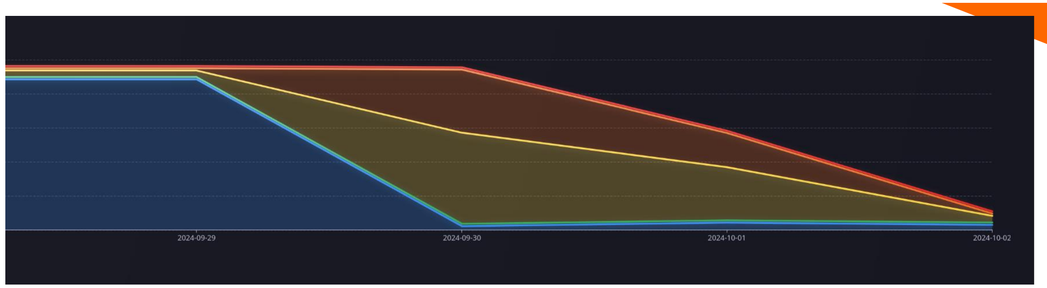
Container Lineage to Complete the pictures

Libraries that are deployed : fix in the library

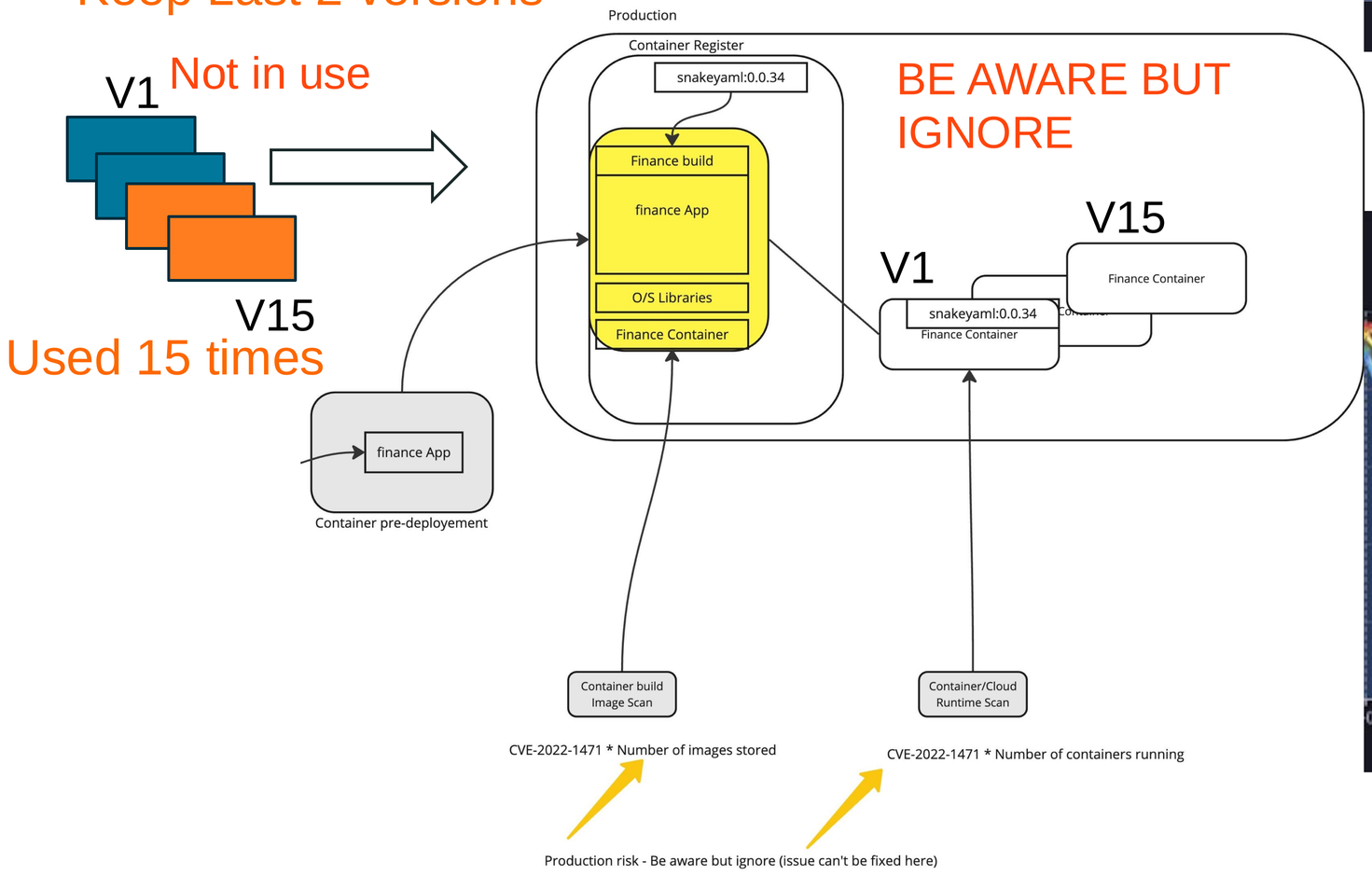


Container Version Throttling

How many active containers do I need to scan?



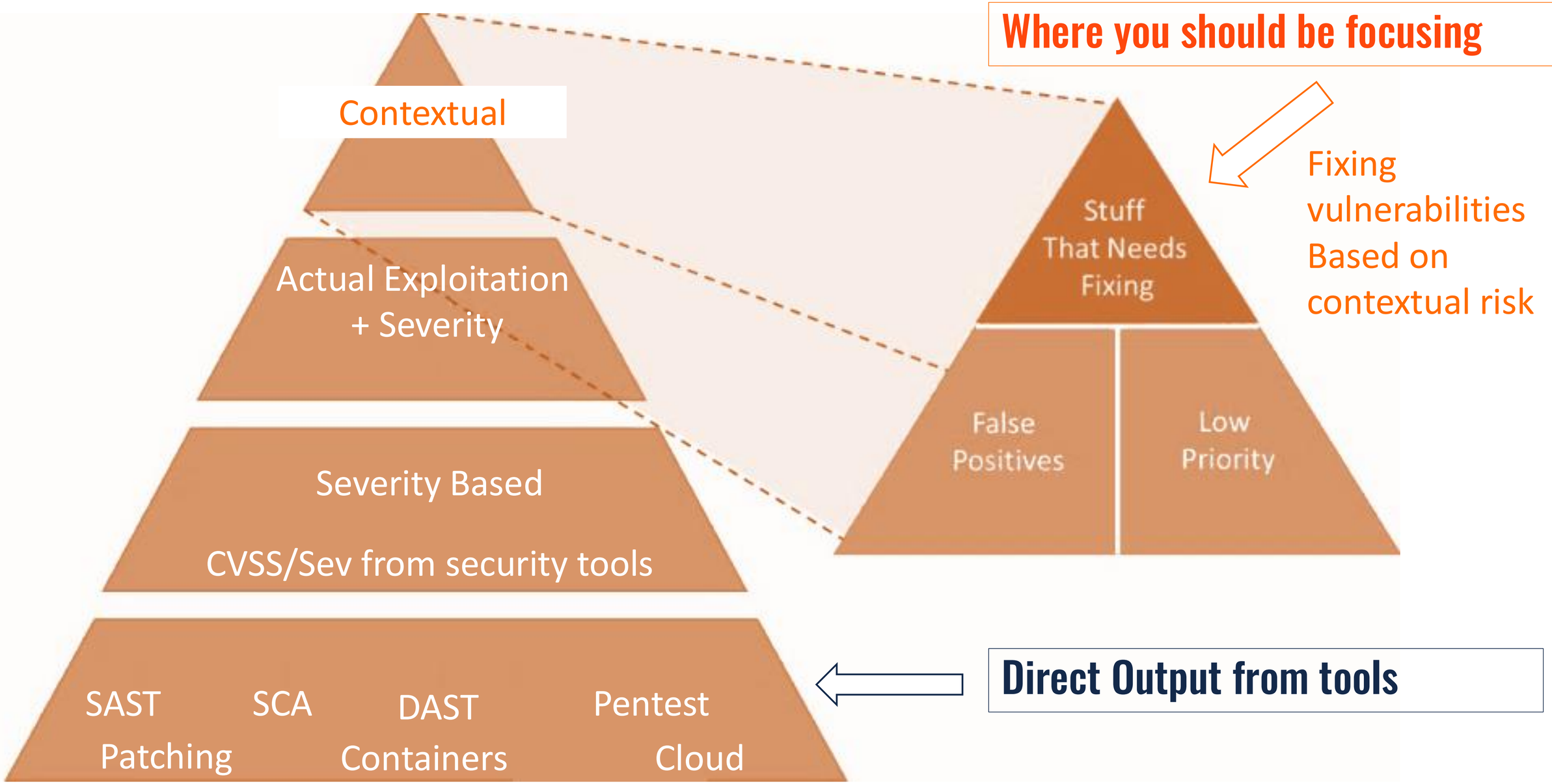
Keep Last 2 versions



91% Reduction of Container Vulnerabilities

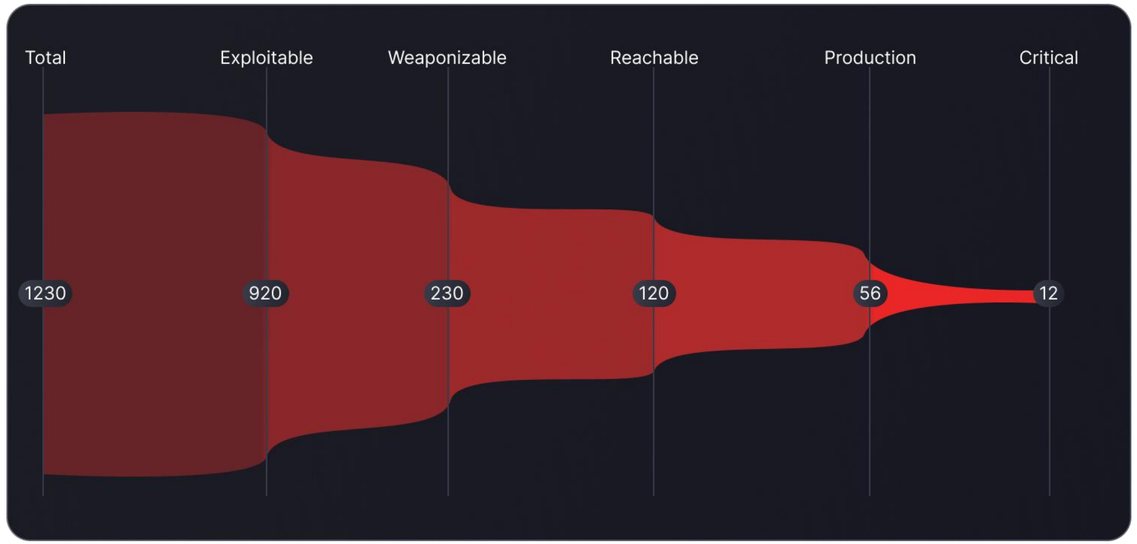
Part 2 (cont) - Goal

Prioritization is so 90....

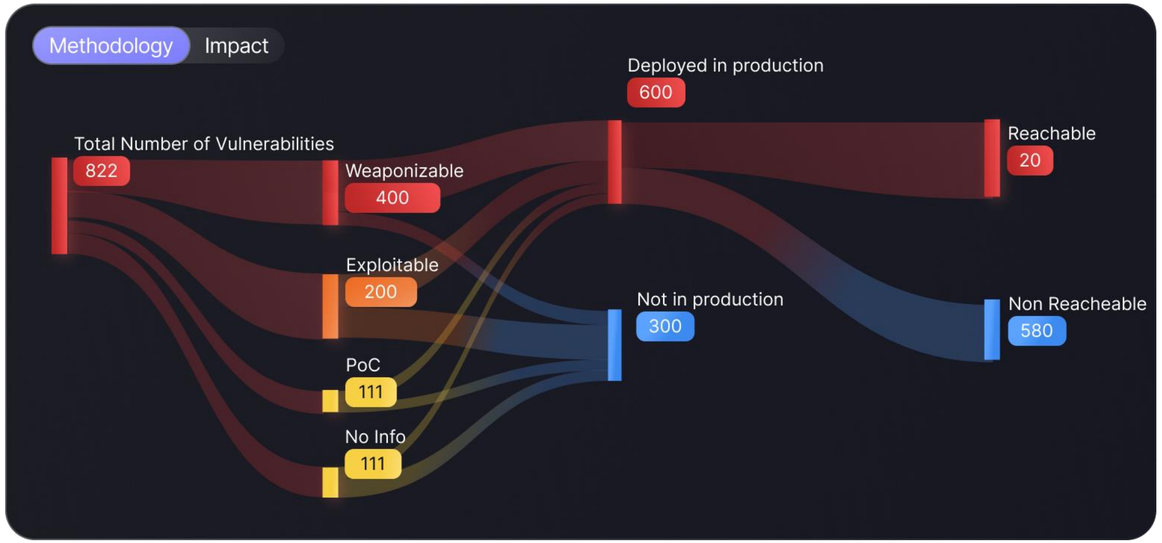


ANSWERING QUESTION

How many Vulnerabilities are actually important

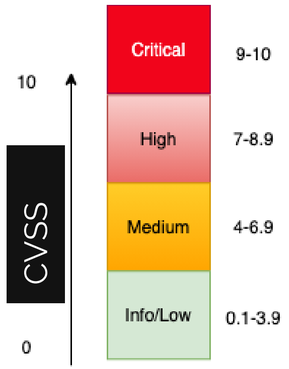


How many exploitable/Weaponizable vulnerability you have

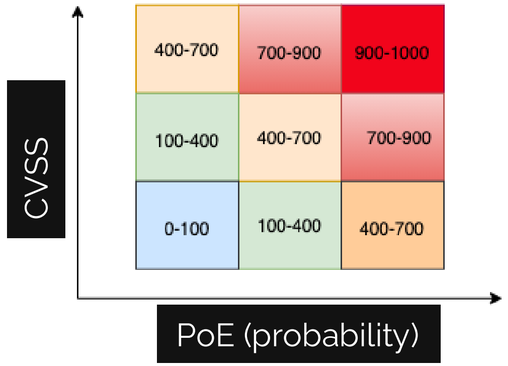


PHOENIX BRINGS OUT THE 4TH DIMENSION OF REACHABILITY

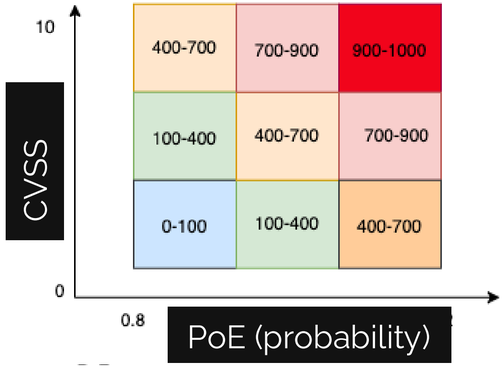
Current Method



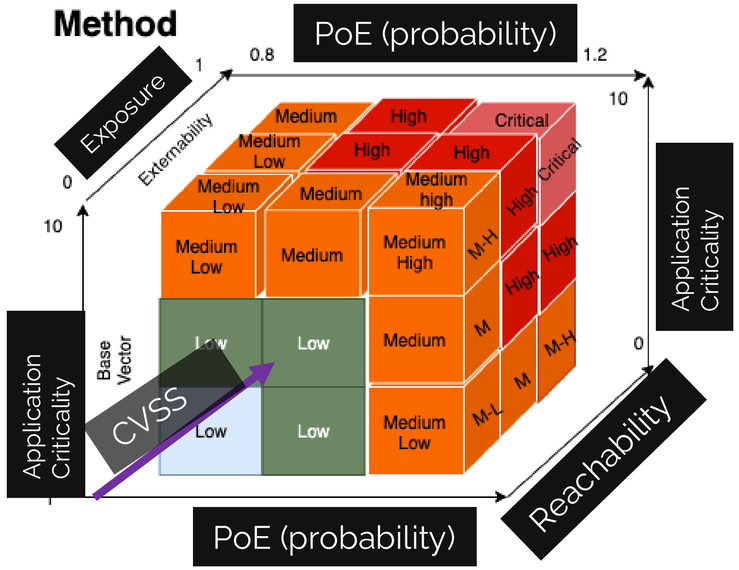
Risk Based Method



Context Based Method



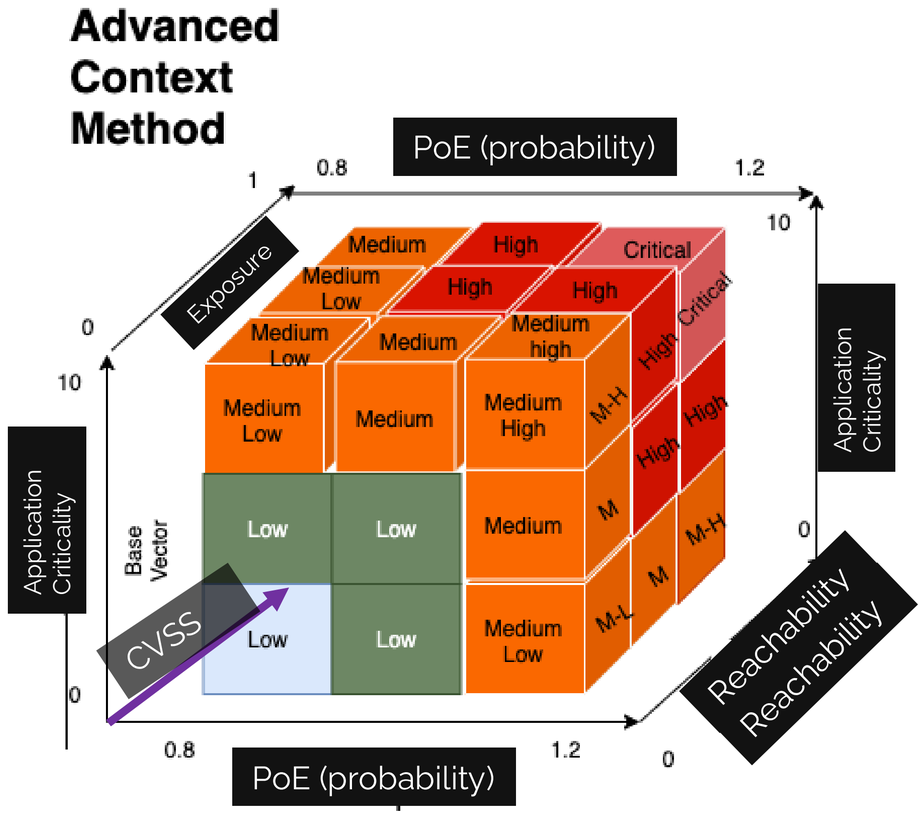
Advanced Context Method



Vulnerability method from past

Phoenix 4D Contextual Reachability Risk

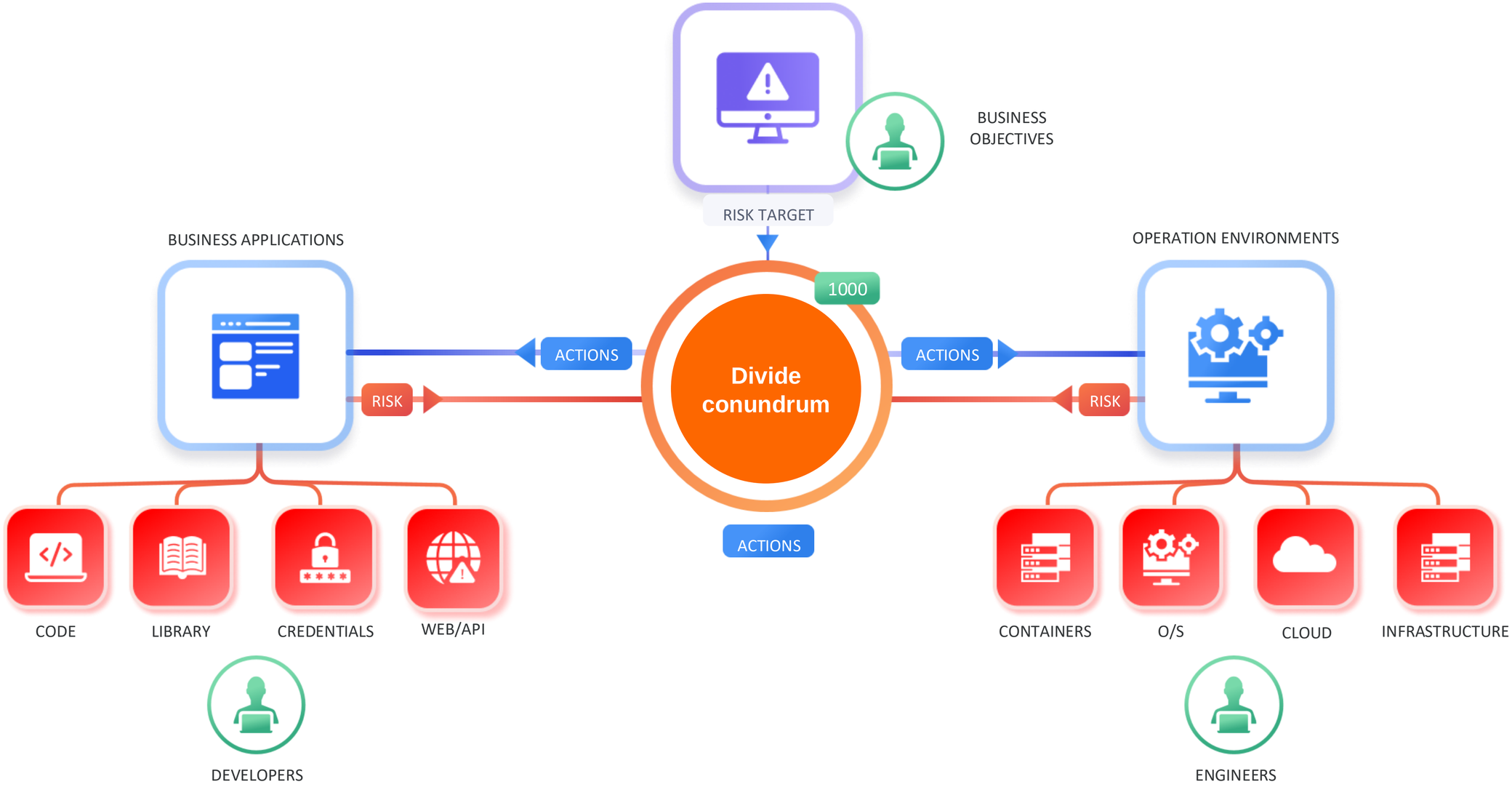
PHOENIX BRINGS OUT THE 4TH DIMENSION OF REACHABILITY



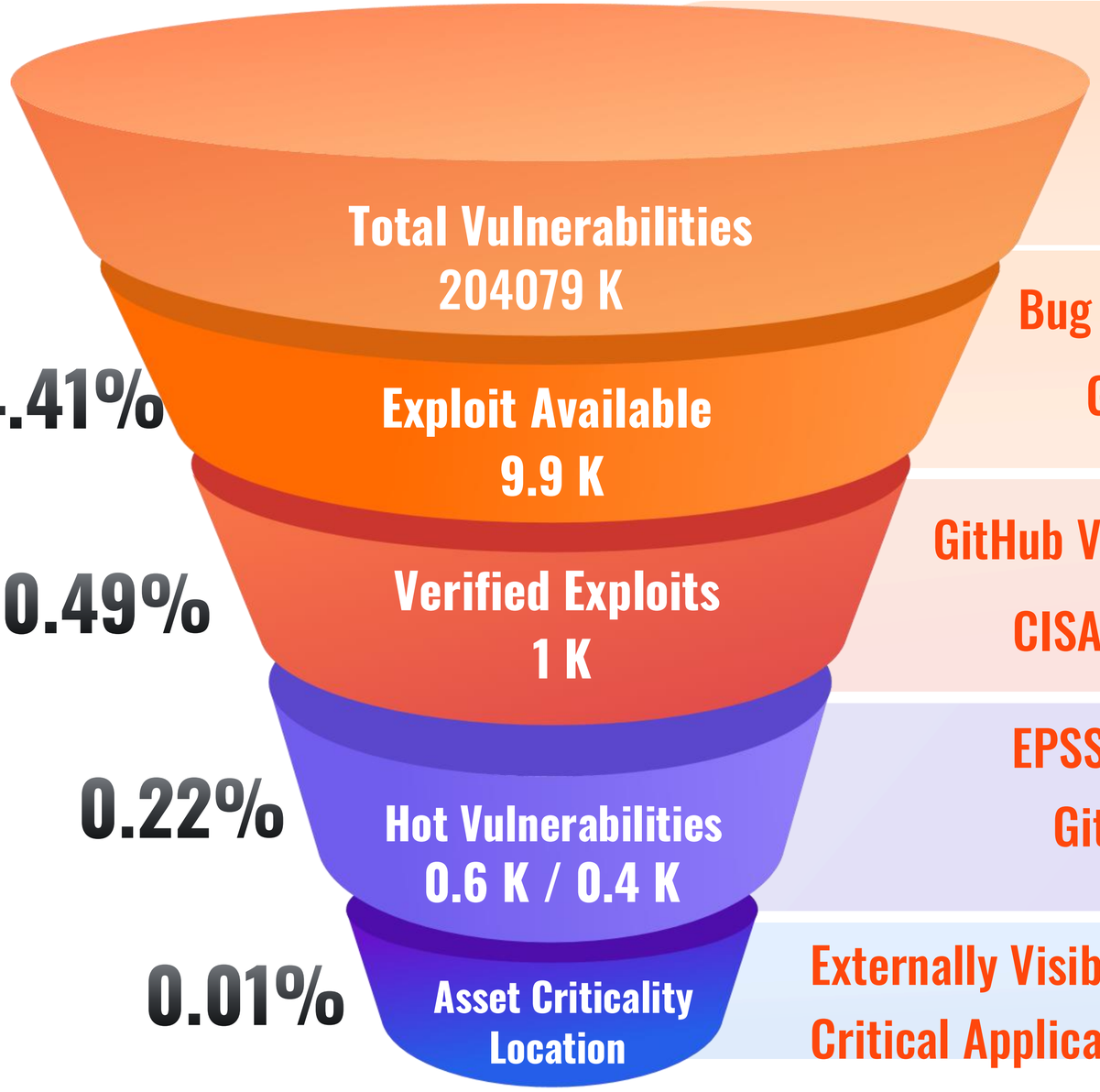
Part 2 (cont) - Communicating with the right context

From Number of Vulnerabilities to risk objectives

Drive Risk down, Connect left to right



Not all the vulnerabilities require equal attention



4.41%

0.49%

0.22%

0.01%

Bug Bounty Popularity (active 17K) 8.33%

GitHub Exploit (9.9K) 4.41%

GitHub Verified Exploits (0.93K) 0.47%

CISA KEV (1K) 0.49%

EPSS > 0.7 (688) 0.34%

GitHub Active Exploit (0.40K) 0.22%

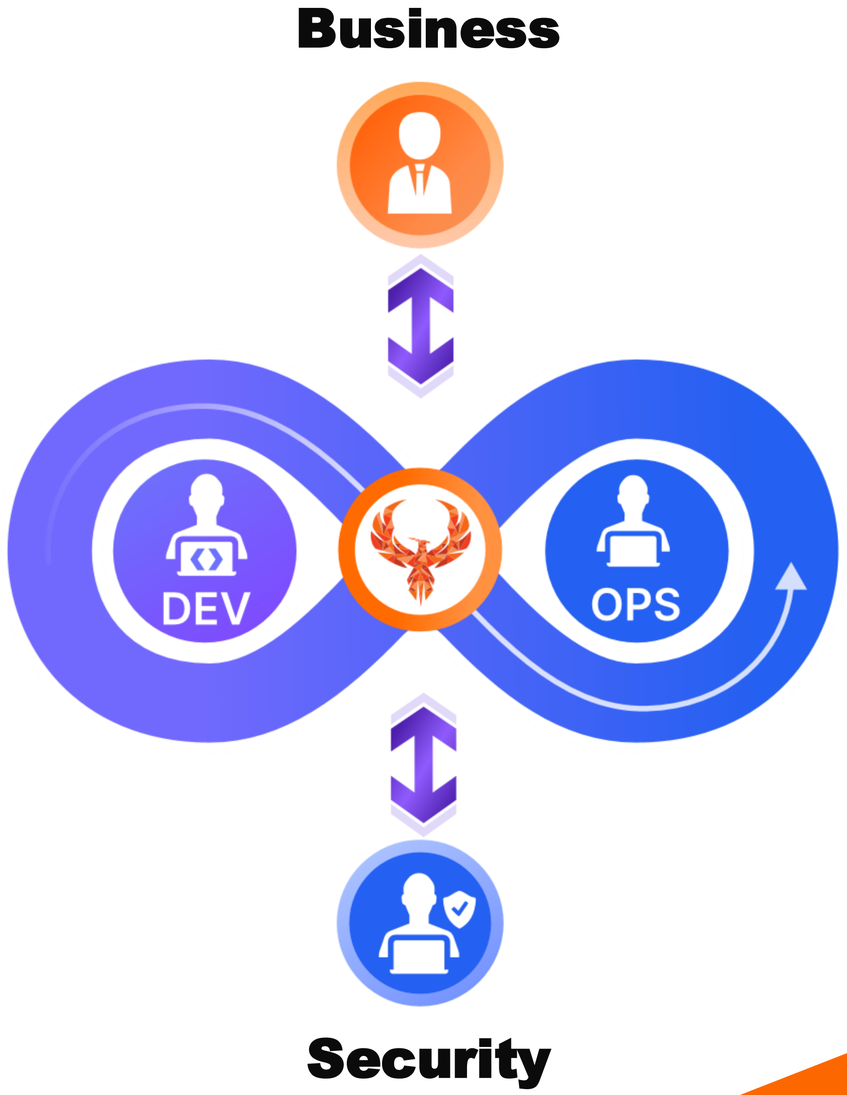
Externally Visible (0.14) ?

Critical Application 0.?%

Part 3 - Scaling without an army Data Driven Approach

Phoenix Security translates Business Risk objectives into precise actions for engineers

Identifying with Contextual AI the best fix to resolution



RISK COMMON LANGUAGE

Code Runtime Attack Path

Reachability analysis

FIX AVAILABLE

THREAT INTEL

EXPLOITABILITY

HOW MANY USERS

HOW IMPORTANT

MORE ACCURATE SCORING

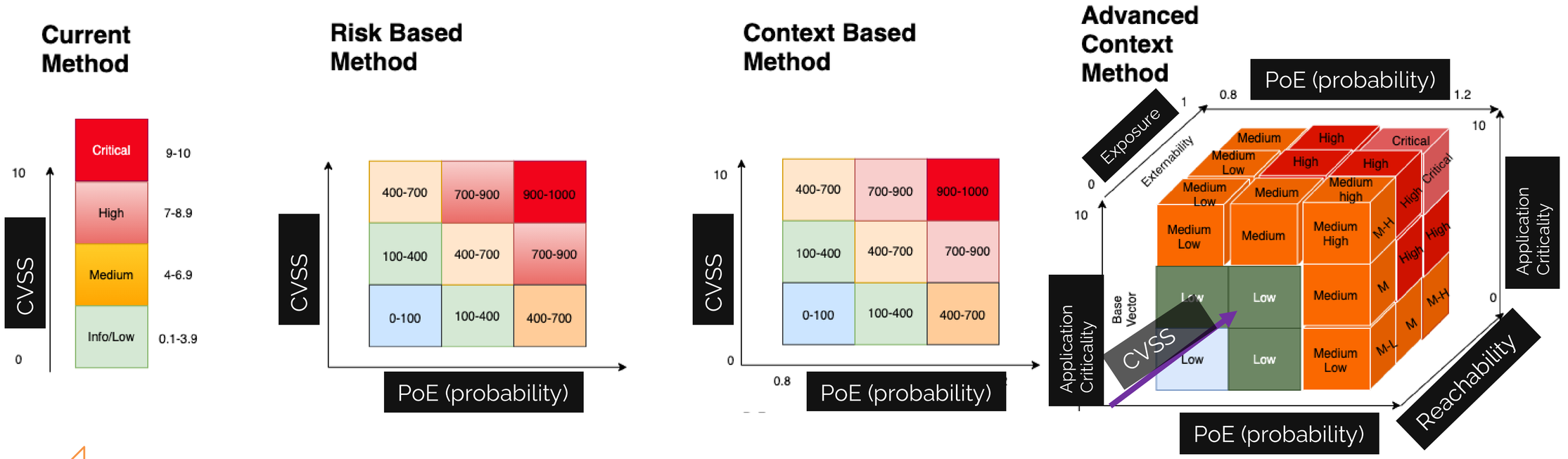
SEVERITY

PROBABILITY

IMPACT

CONTEXT

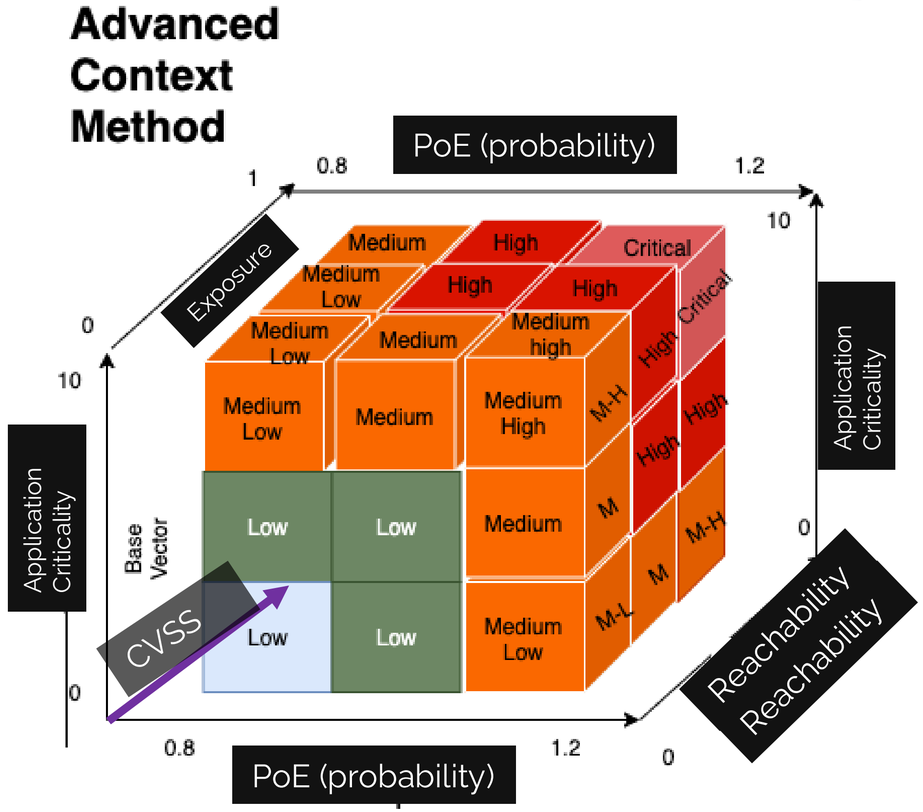
Phoenix brings out the 4th dimension of reachability



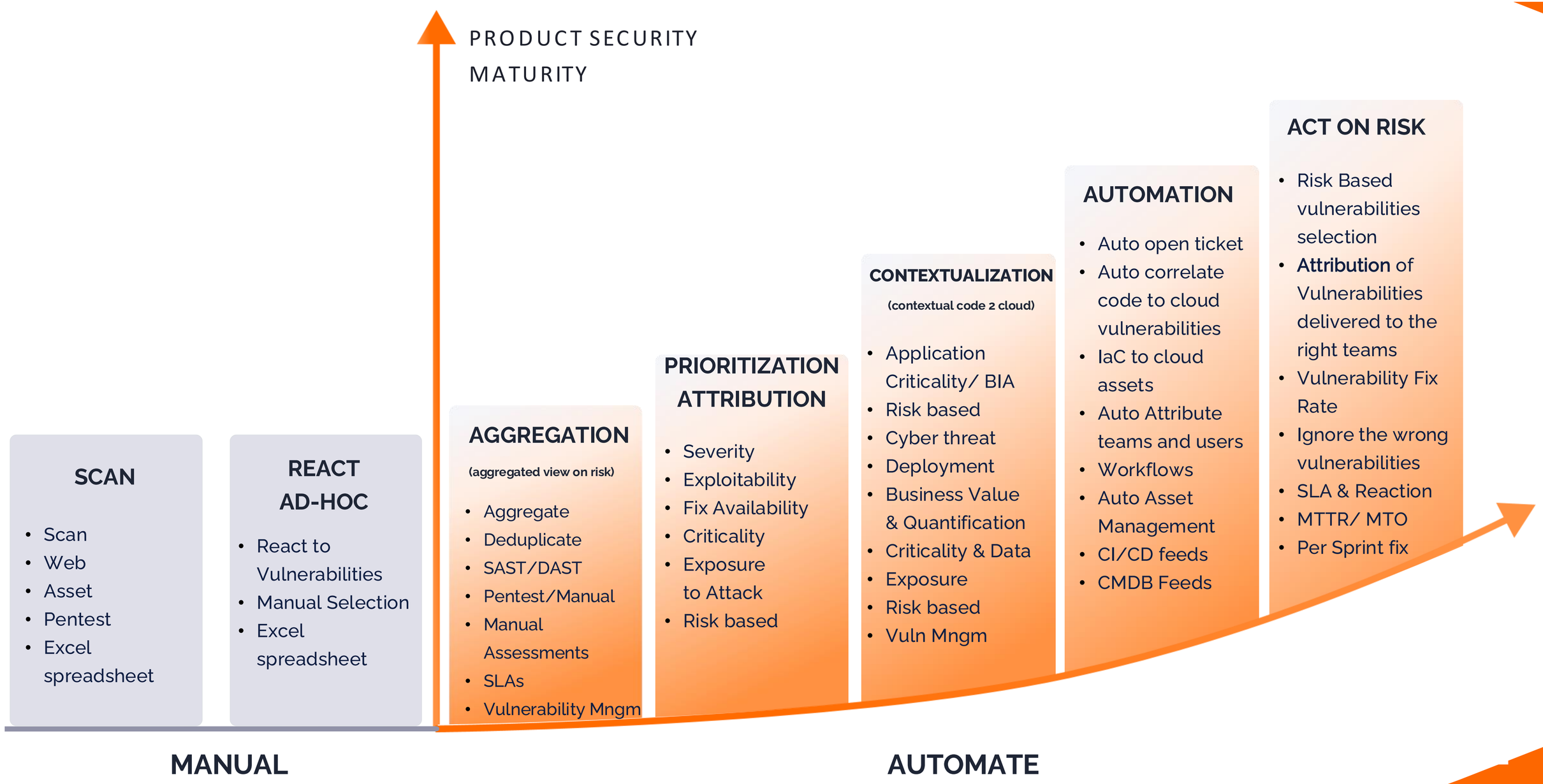
Vulnerability method from past

Phoenix 4D Contextual Reachability Risk

Phoenix brings out the 4th dimension of reachability



WHERE ARE YOU IN YOUR SOFTWARE SECURITY MATURITY JOURNEY?



Conclusions

So we solved security right?

There is a light at the end of the tunnel

- Prioritize what matters: exploitable weaponizable
- Application Security + Environment > products and owners
- Remove noise with reachability and throttling
- Help dev to focus on remediation: container, libraries

PHOENIX SECURITY

ACT ON CONTAINER VULN

ACT ON ENDPOINT VULN

ACT ON CLOUD VULN

CONTEXTUALIZE, PRIORITIZE & ACT ON RISK

ACT ON APPSEC VULN

ACT ON INFRA VULN

ACT ON CODE VULN

ACT ON SBOM VULN

PHOENIX SECURITY

CYBER RISK DEFENDERS CLUB

WEBINAR

OWASP

REACHABILITY ANALYSIS AND THE FUTURE OF ASPM

Fireside conversation with James Berthoty

James Berthoty

APPLICATION SECURITY EXPERT
TURNED ANALYSIS @ LATIO TECH

ASPM, CNAPP, Reachability analysis, why they are all connected and why is the future of application security actionable considering what you build and where you run it so important

www.phoenix.security.com 30 October

10 AM PST / 12 AM CT / 1 PM ET/ 5 PM GMT

Phoenix Security Unify ASPM & CSPM for a contextual approach

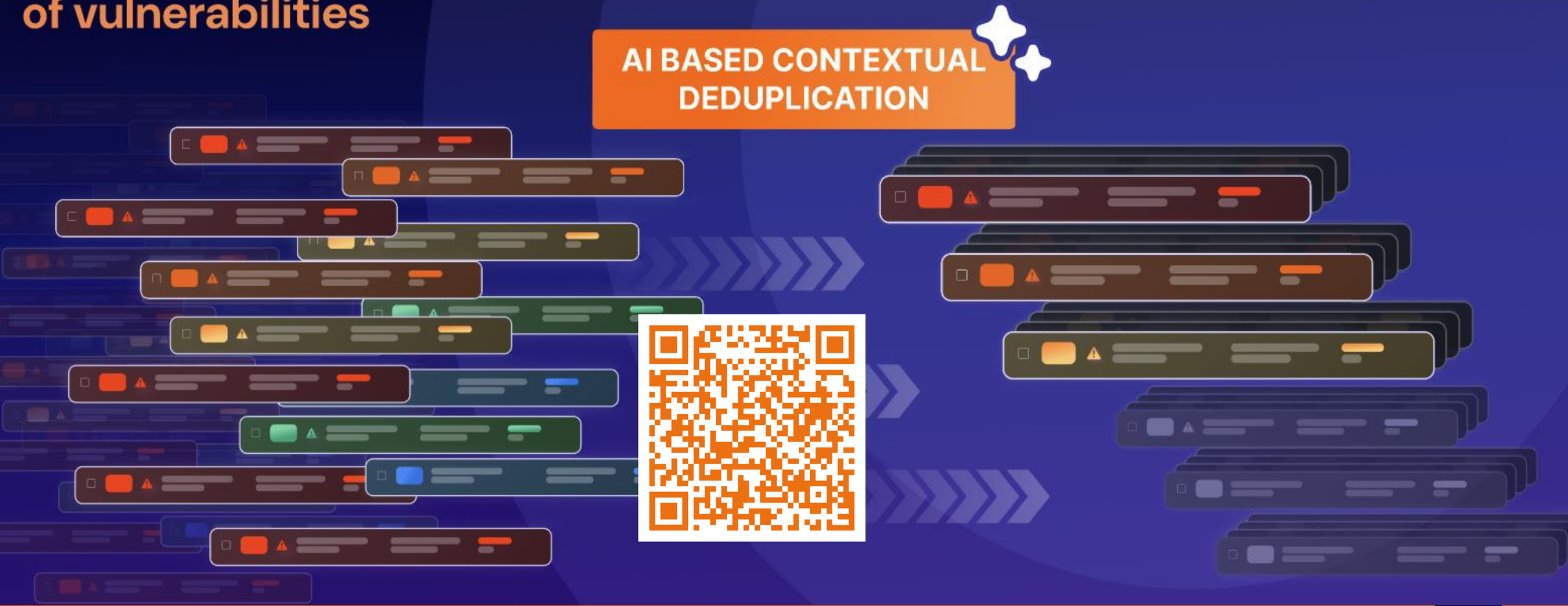
OWASP®

IDENTIFY PROBLEMS | **ORGANIZE, PRIORITIZE, CONTEXTUALIZE** | **ACTIONS ON RISK**

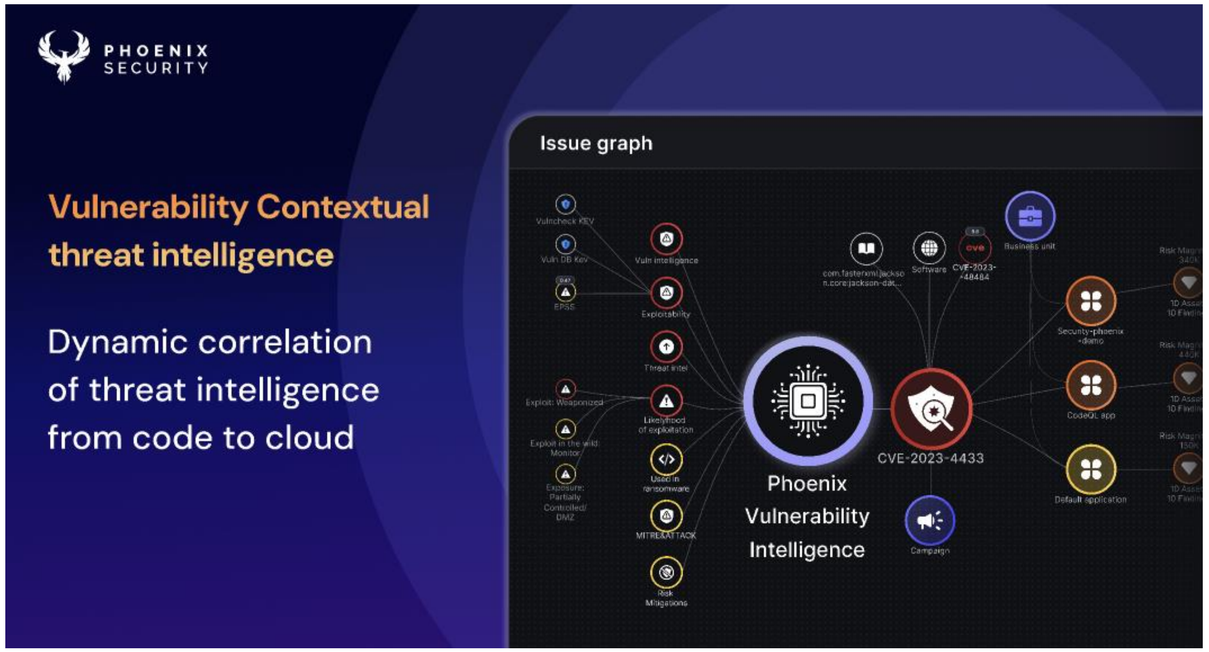
PHOENIX SECURITY

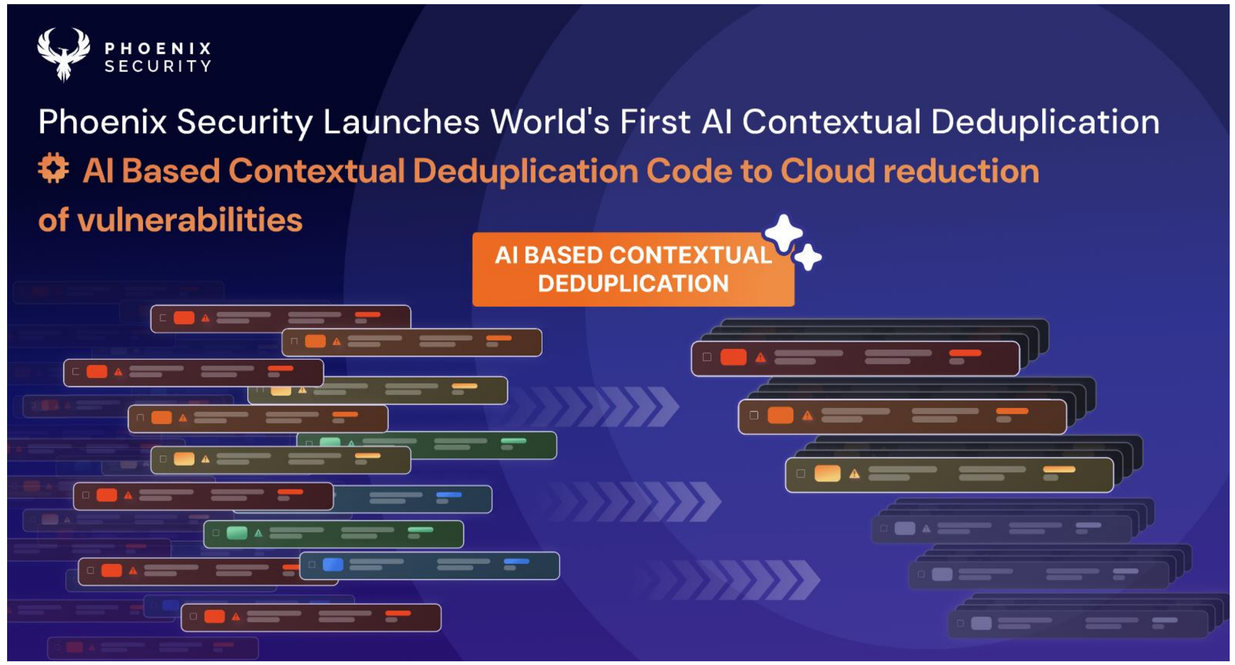
Phoenix Security Launches World's First AI Contextual Deduplication

AI Based Contextual Deduplication Code to Cloud reduction of vulnerabilities

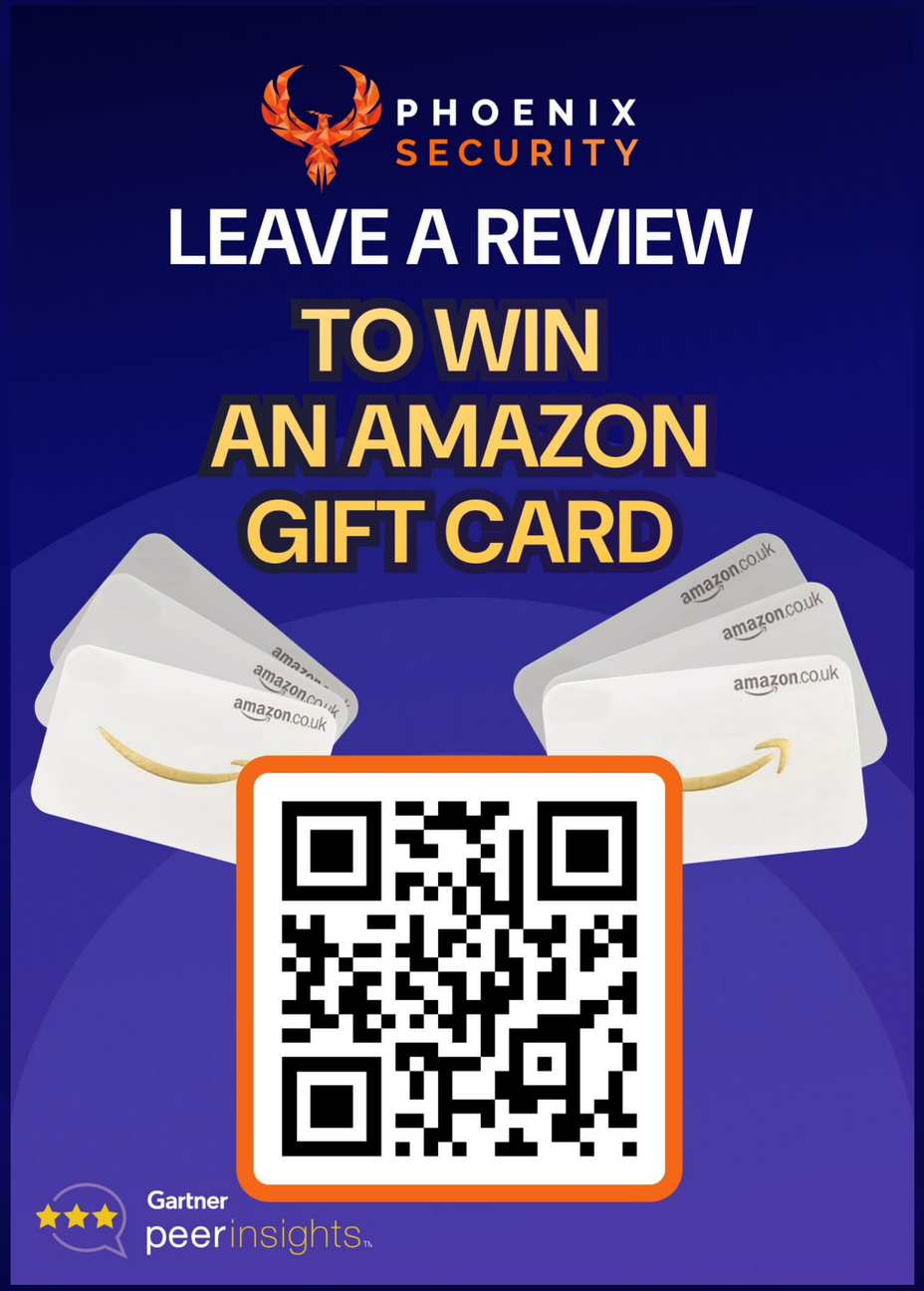


Upcoming New Features





Penny for your time (and thoughts)



Get a demo today and provide your feedback

Win an amazon Gift Card

Building resilient application and cloud security programs

NEW EDITION

PHOENIX SECURITY

BUILDING RESILIENT APPLICATION AND CLOUD SECURITY PROGRAMS

Author
Francesco Cipollone
CEO & Founder
Phoenix Security

Timo Pagel
DevSecOps (DSOMM)

Kane Narrraway
Security @ CANVA

OMO OSAGIEDE
Security Architect

Chris Hughes
CEO & Founder
ACQUIA

Sam Moore
Vulnerability Management @ TMOBILE

Anuprita Patankar
Product Security @ Ecommerce Company

Chintan Gurjar
Vulnerability Management @ M&S

Cyber Risk Defender Club

PHOENIX SECURITY

CYBER RISK

DEFENDERS CLUB

Author
Francesco Cipollone
CEO & Founder
Phoenix Security

Timo Pagel
DevSecOps
(DSOMM)

Kane
Narrraway
Security @
CANVA

OMO
OSAGIEDE
Security
Architect

Chris Hughes
CEO & Founder
ACQUIA

Sam Moore
Vulnerability
Management @
TMOBILE

Anuprita
Patankar
Product
Security @
Ecommerce
Company

Chintan
Gurjar
Vulnerability
Management
@ M&S

PHOENIX SECURITY

ACT ON CONTAINER VULN

ACT ON ENDPOINT VULN

ACT ON CLOUD VULN

CONTEXTUALIZE, PRIORITIZE & ACT ON RISK

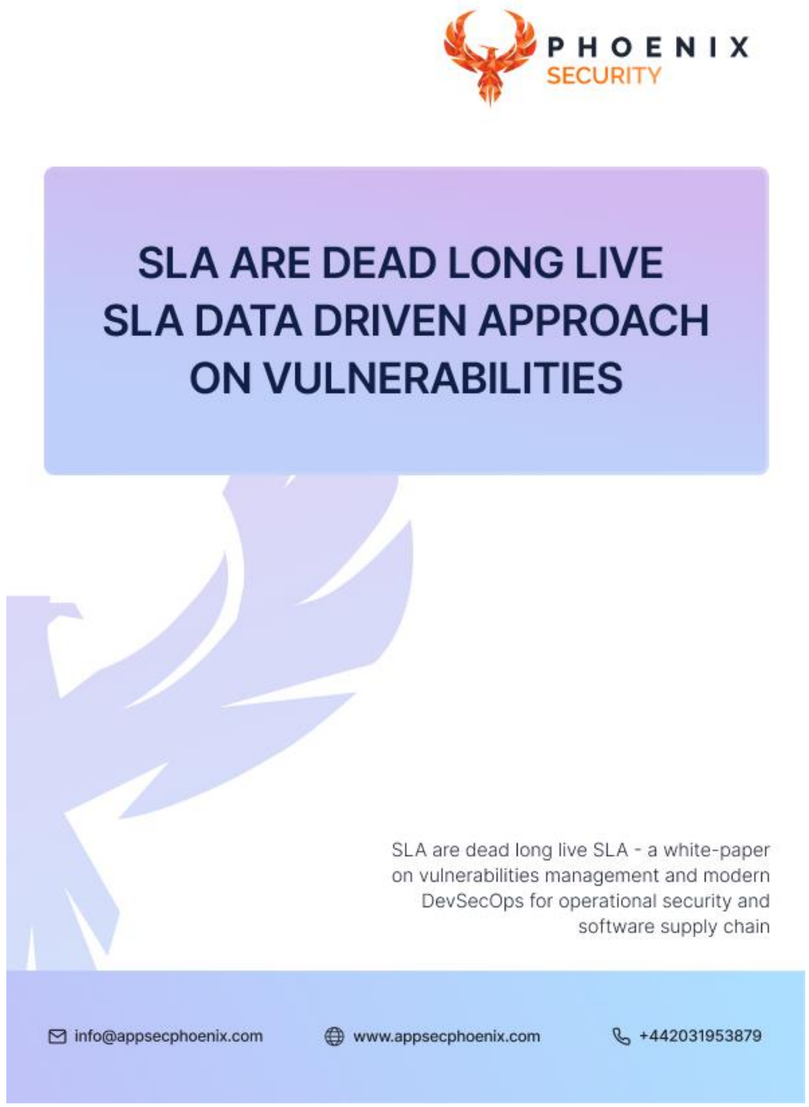
ACT ON APPSEC VULN

ACT ON INFRA VULN

ACT ON CODE VULN

ACT ON SBOM VULN

New Book on metrics that matters



Where can you find more

We have whitepapers on vulnerability management prioritization

PHOENIX SECURITY

APPLICATION & CLOUD SECURITY PROGRAM

PHOENIX SECURITY

VULNERABILITY MANAGEMENT AT SCALE AND THE POWER OF CONTEXT BASED PRIORITIZATION

Cyber Security & Cloud Podcast

By Francesco Cipollone

#CSCP

www.cybercloudpodcast.com

@podcast_cyber

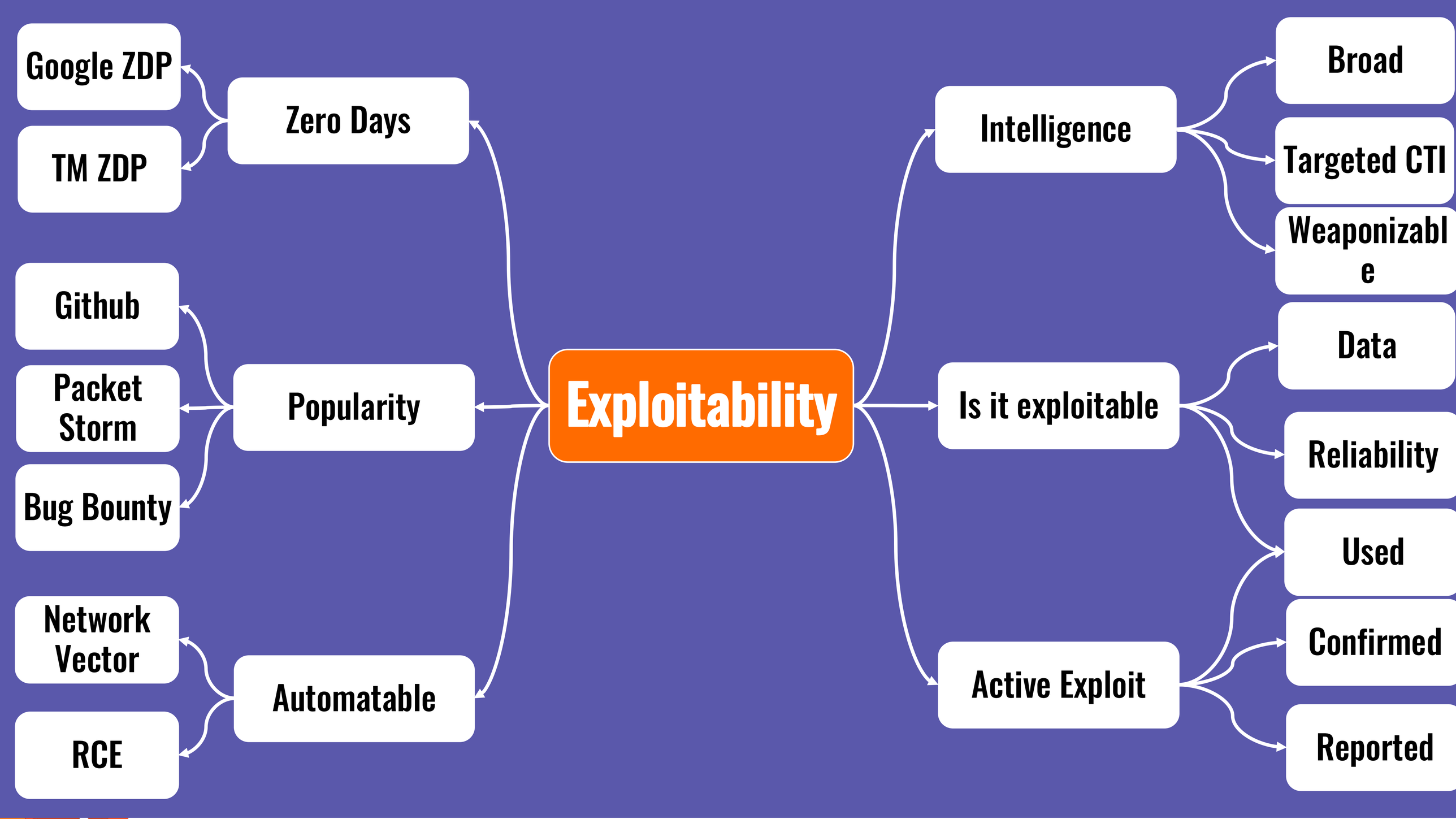
@FrankSEC42

www.cybercloudpodcast.com

Sponsored By

PHOENIX SECURITY

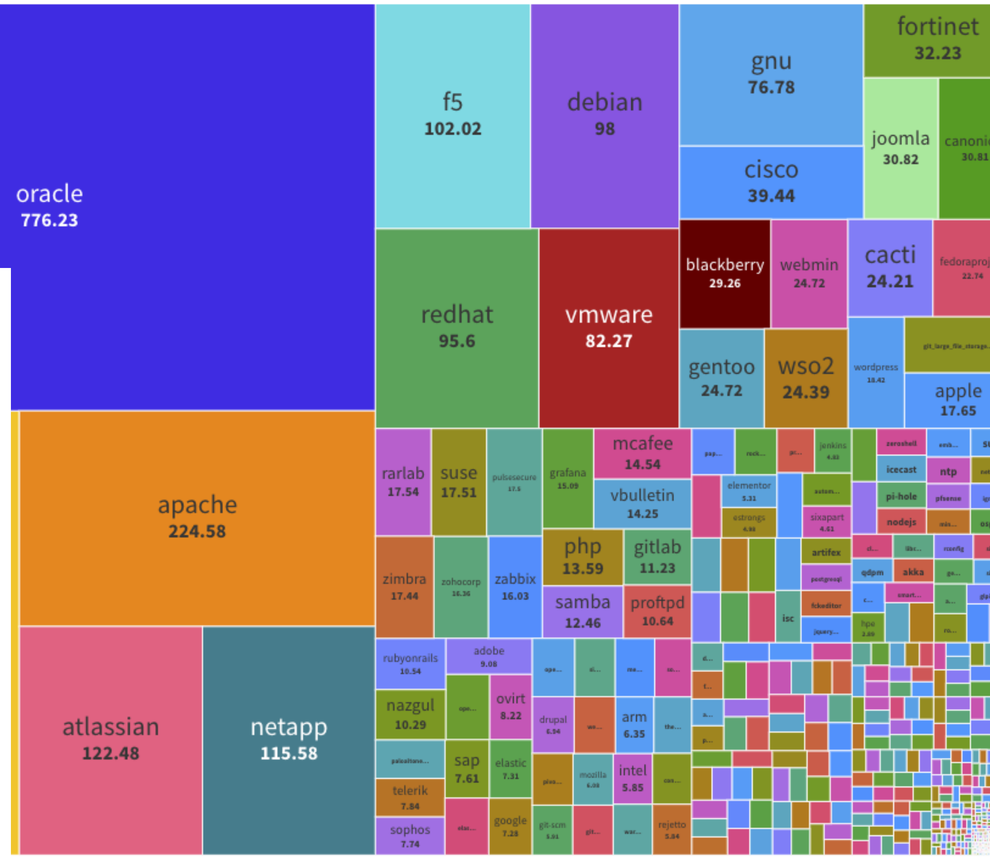
Appendix – CTI Fixing what's more important


Google ZDP
TM ZDP
Zero Days
Github
Packet Storm
Bug Bounty
Popularity
Network Vector
RCE
Automatable
Exploitability
Intelligence
Broad
Targeted CTI
Weaponizable
Is it exploitable
Data
Reliability
Active Exploit
Used
Confirmed
Reported

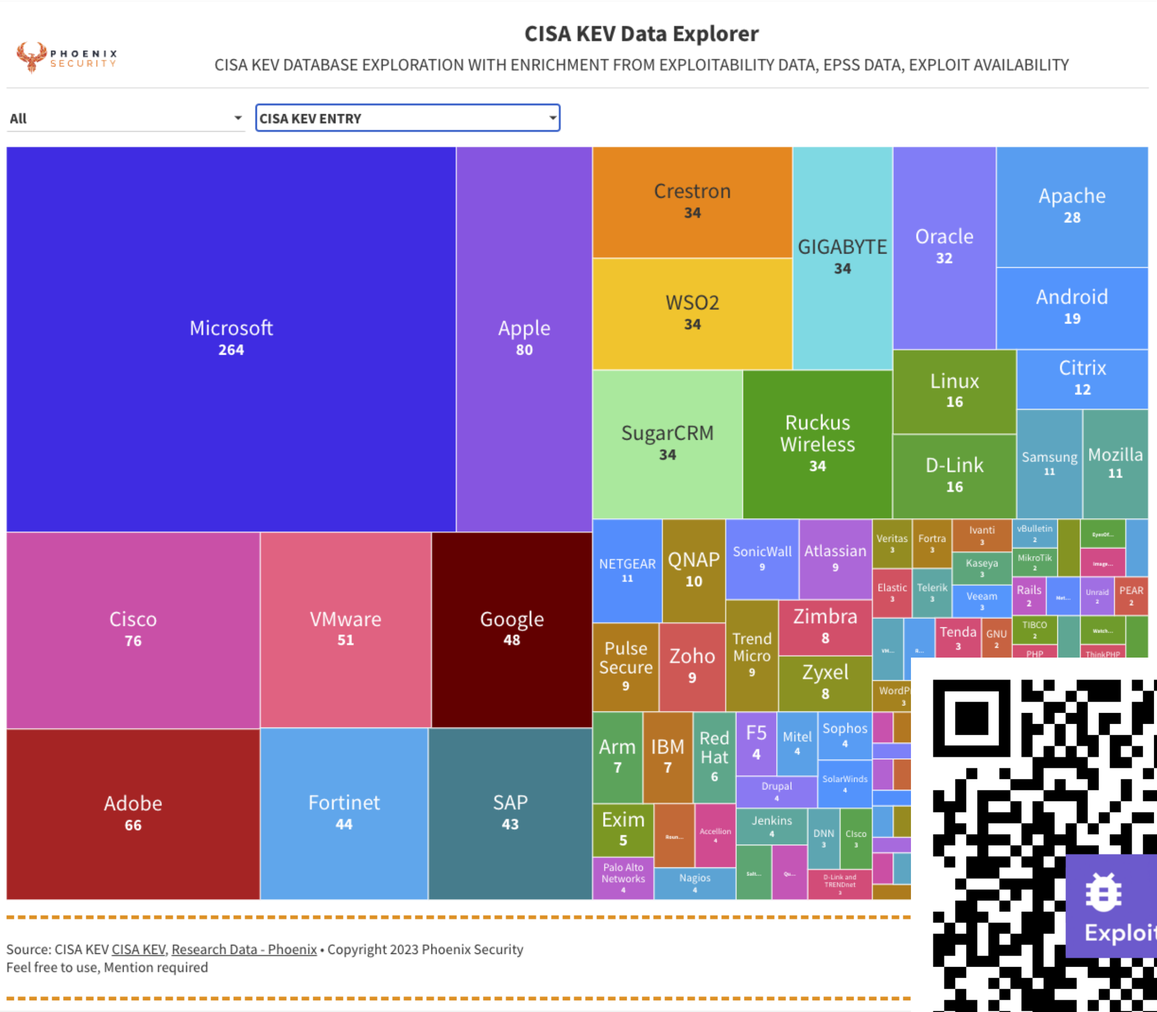
Phoenix CTI - TOP EXPLOITED VENDOR

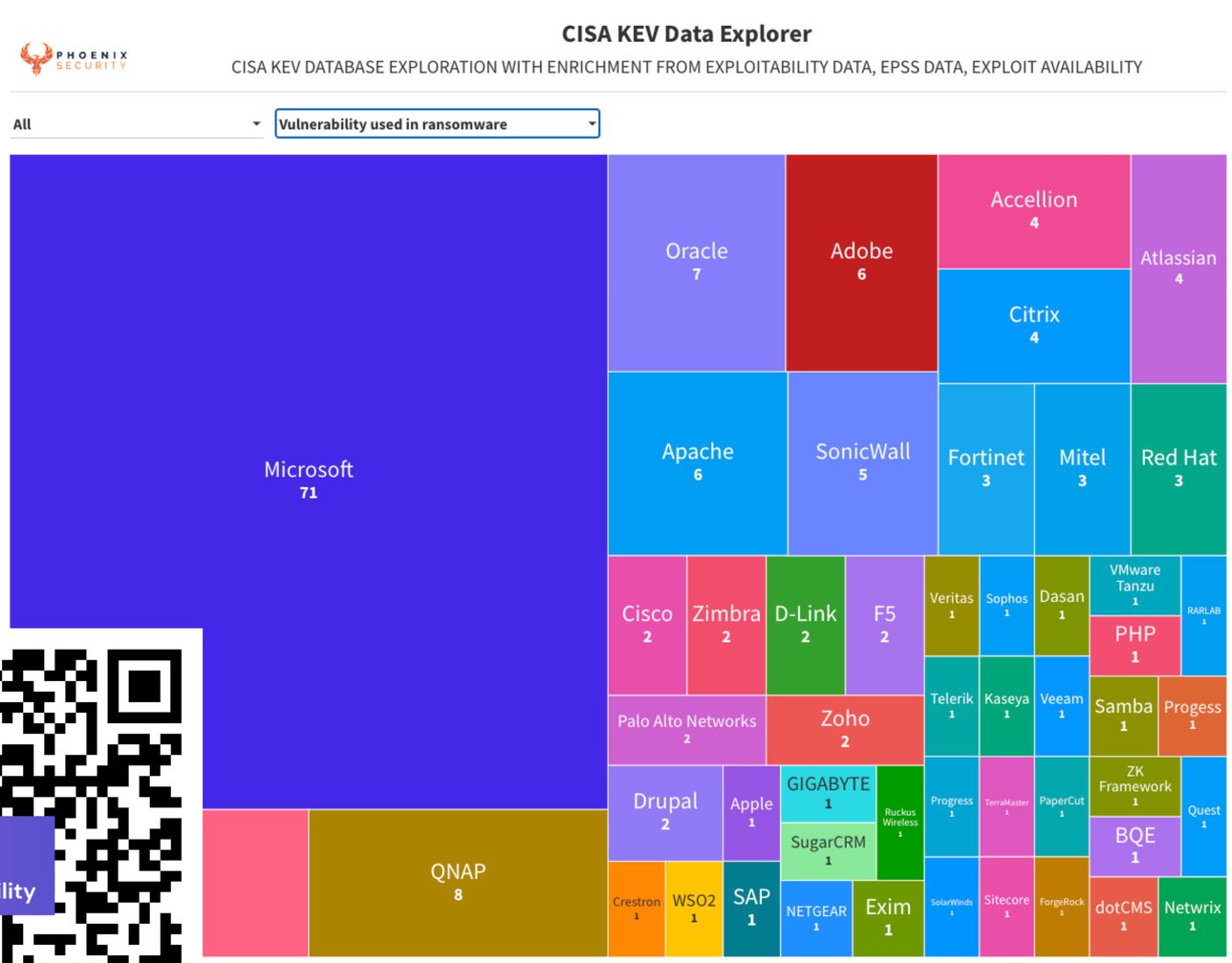
GitHub Active Exploits

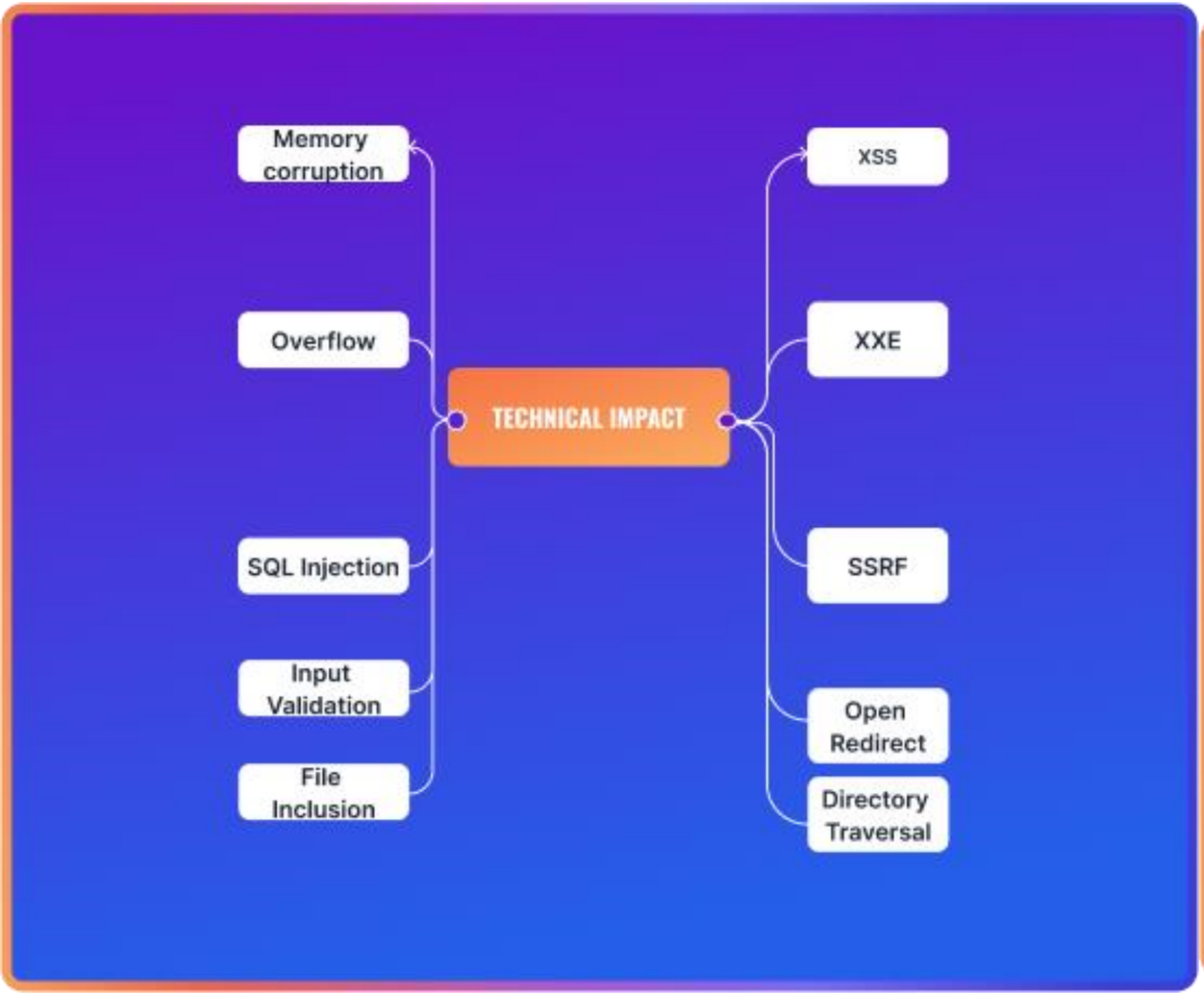
Number of Verified Exploits



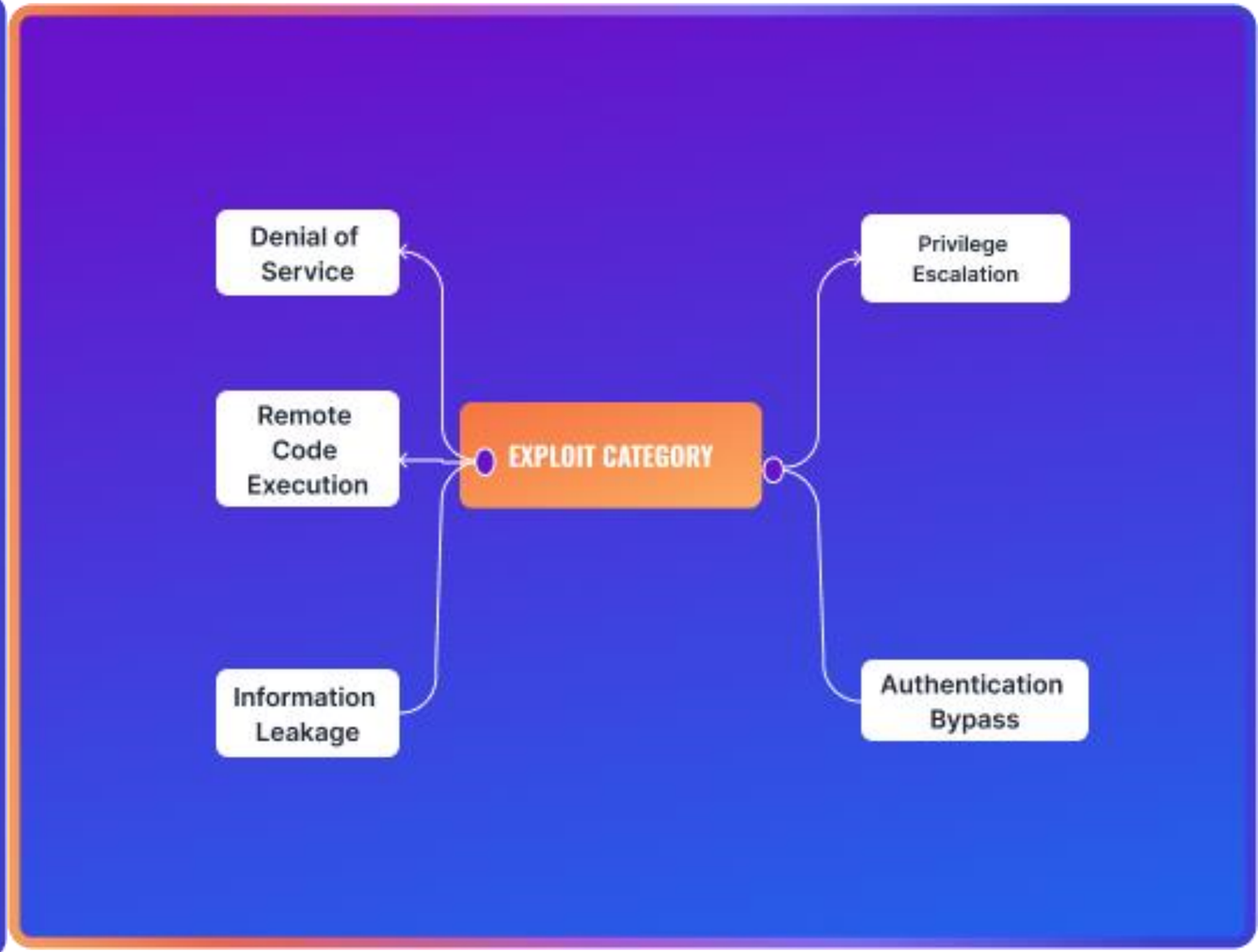
Vulnerabilities used in ransomware







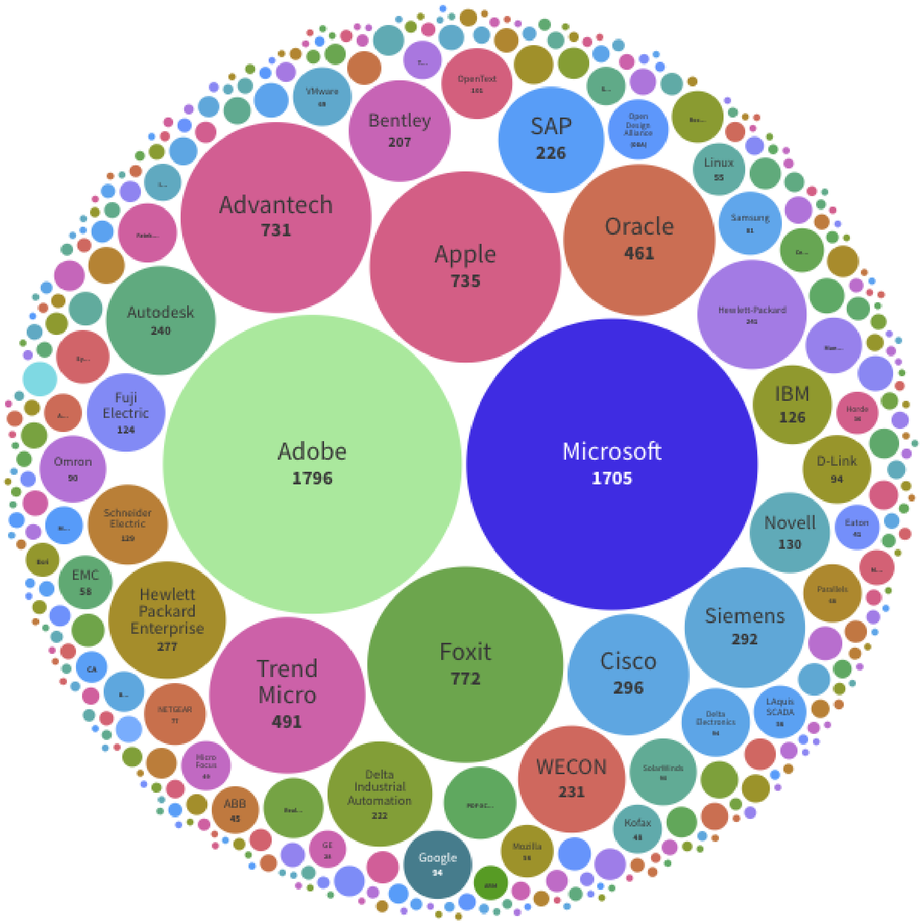
IMPACT OF VULNERABILITIES



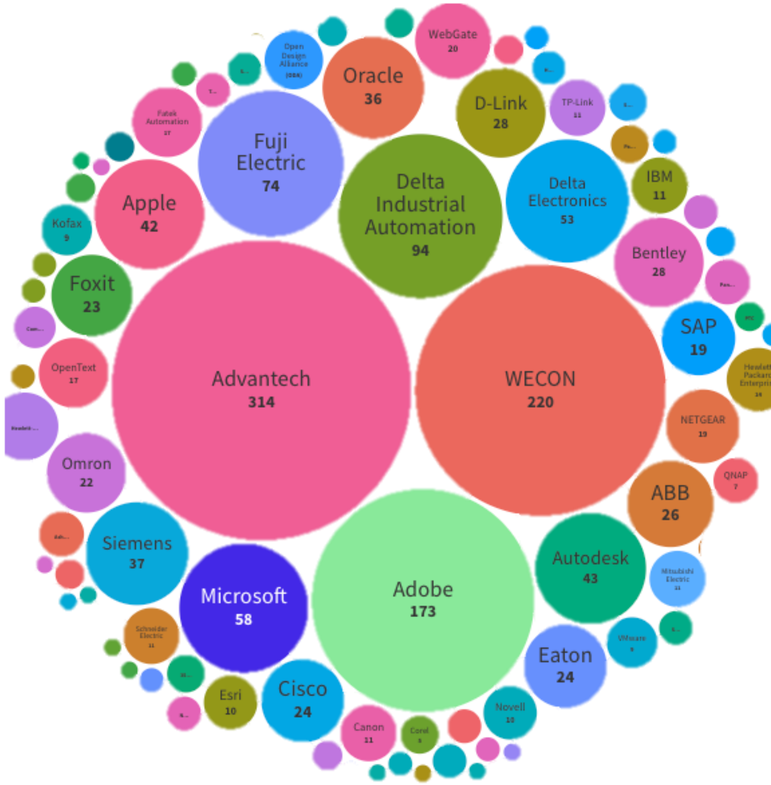
ROOT CAUSE

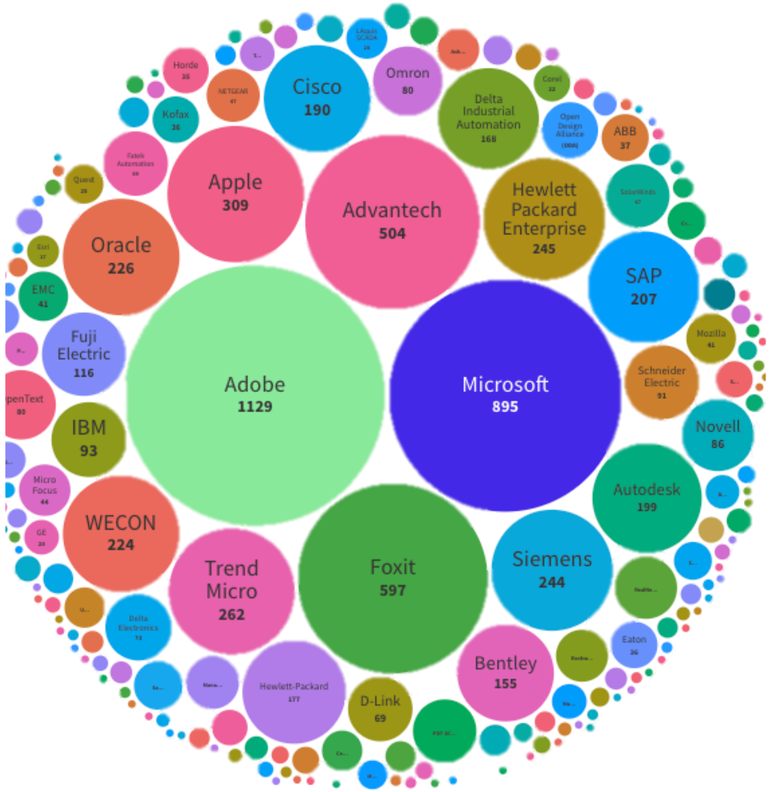
VENDOR WITH MOST ZERO DAY

Exploitability



Buffer Overflow

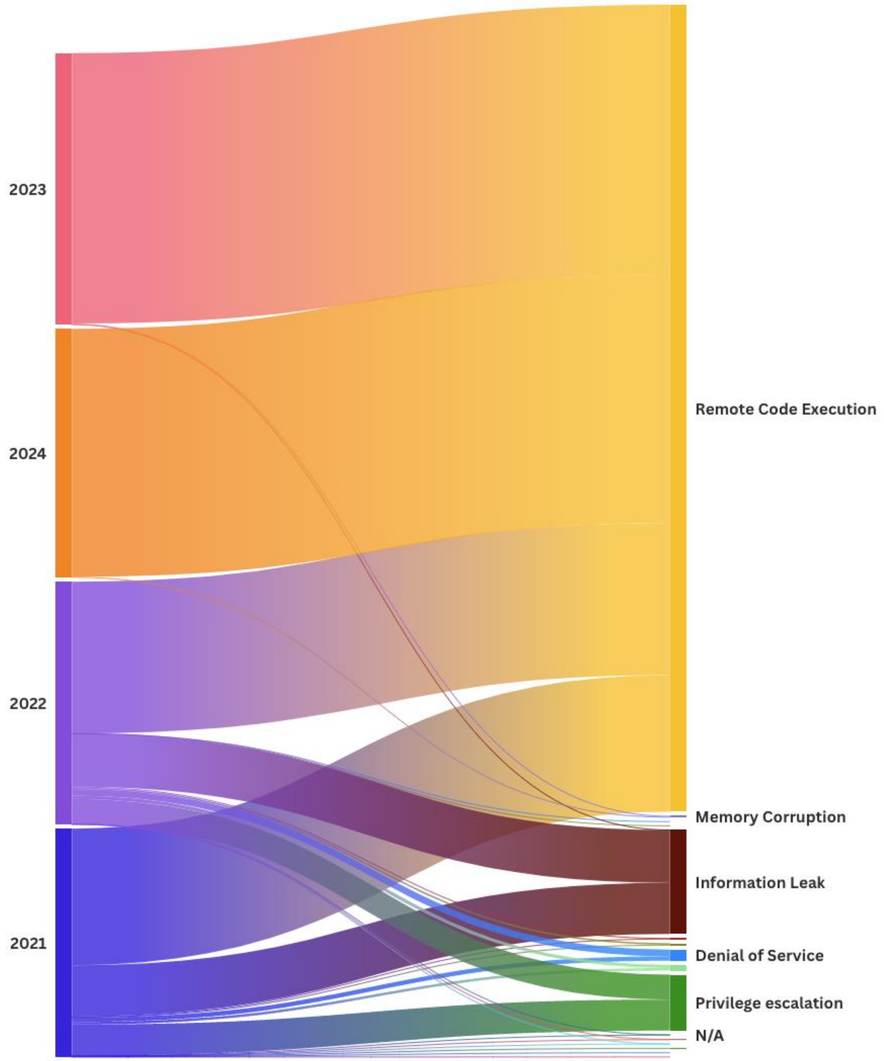




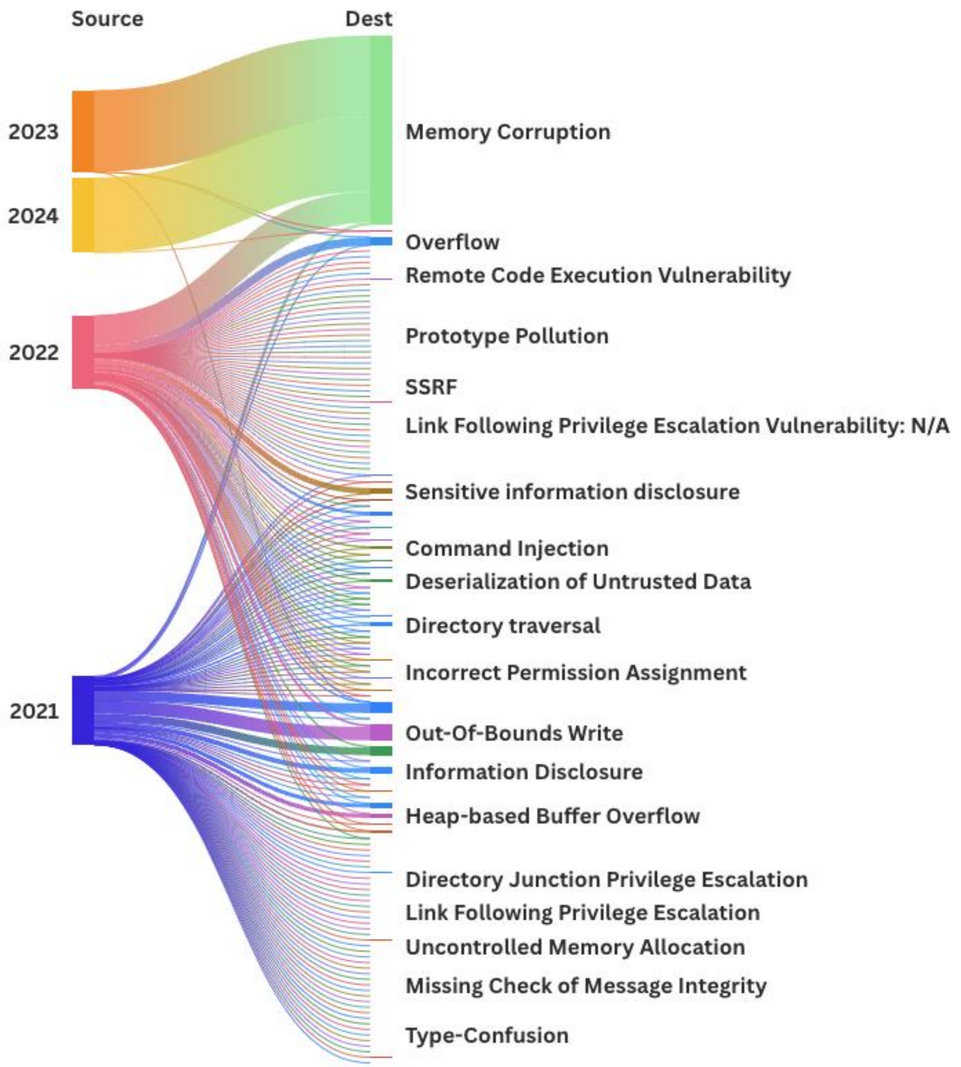
RCE

METHODOLOGIES OF ATTACKS IN ZERO DAYS - WEAKNESS

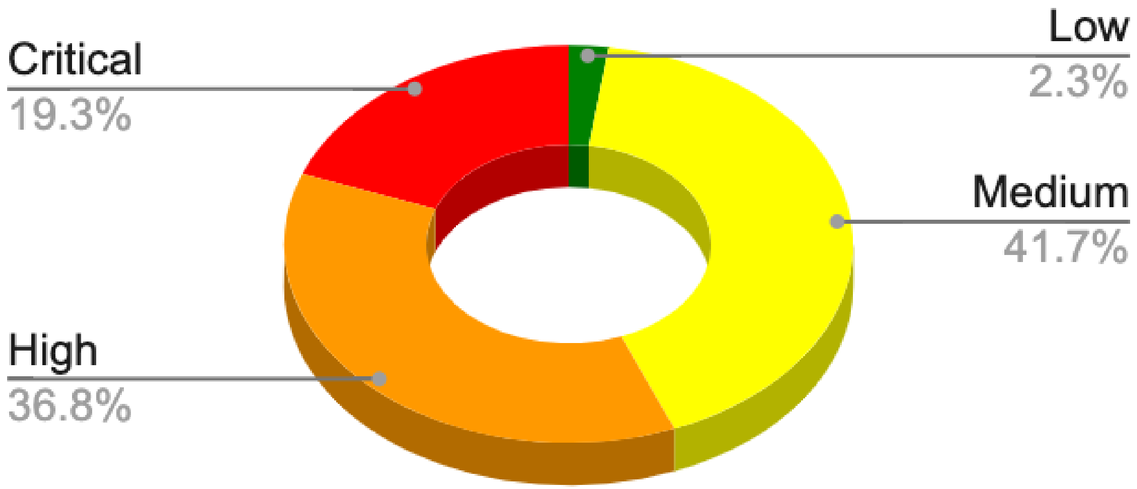
Root Cause : RCE



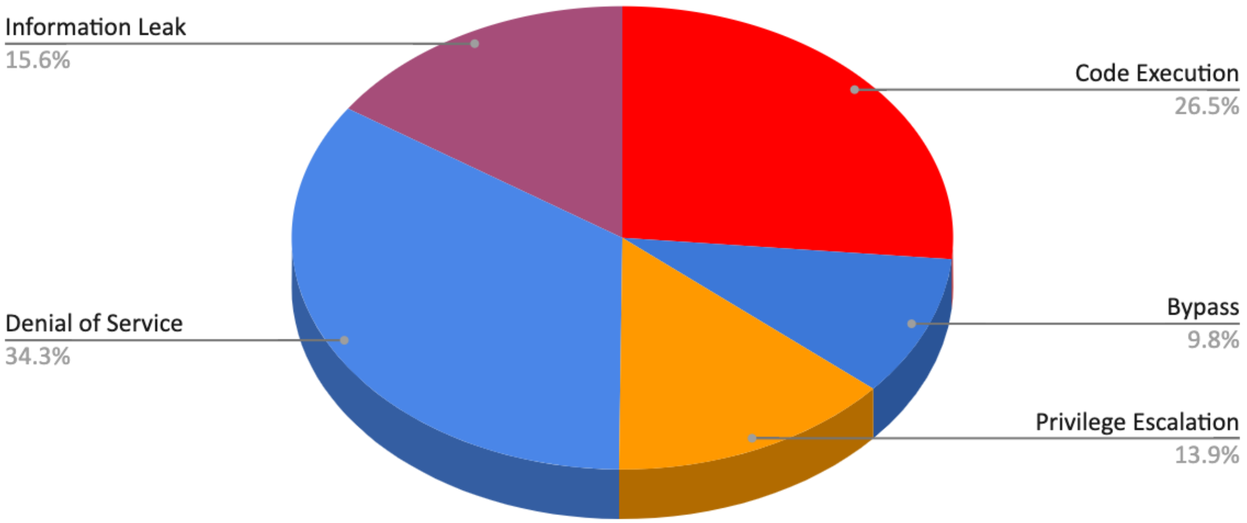
Technical Impact



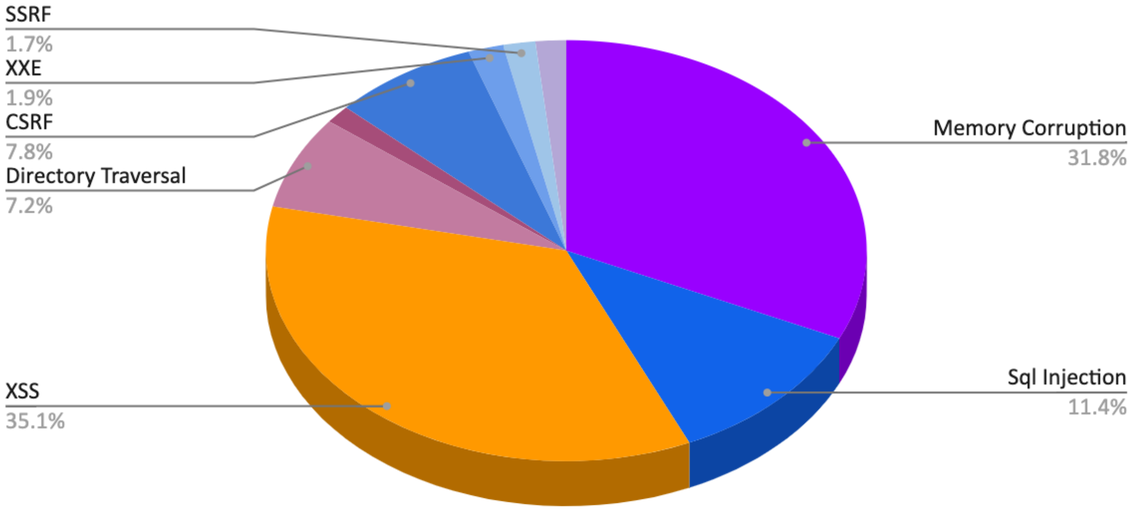
CVE Distribution



CVE Distribution by Effect

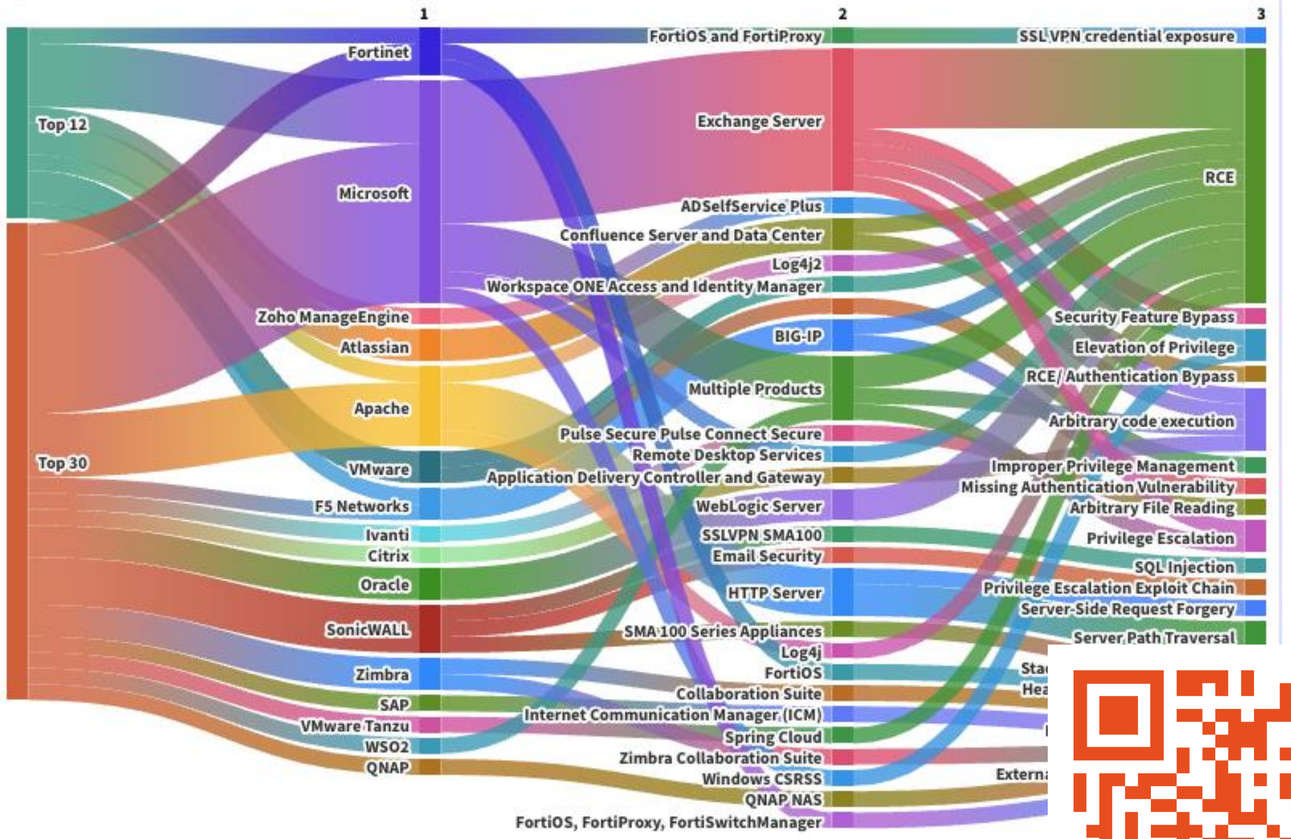


CVE Distribution by Type

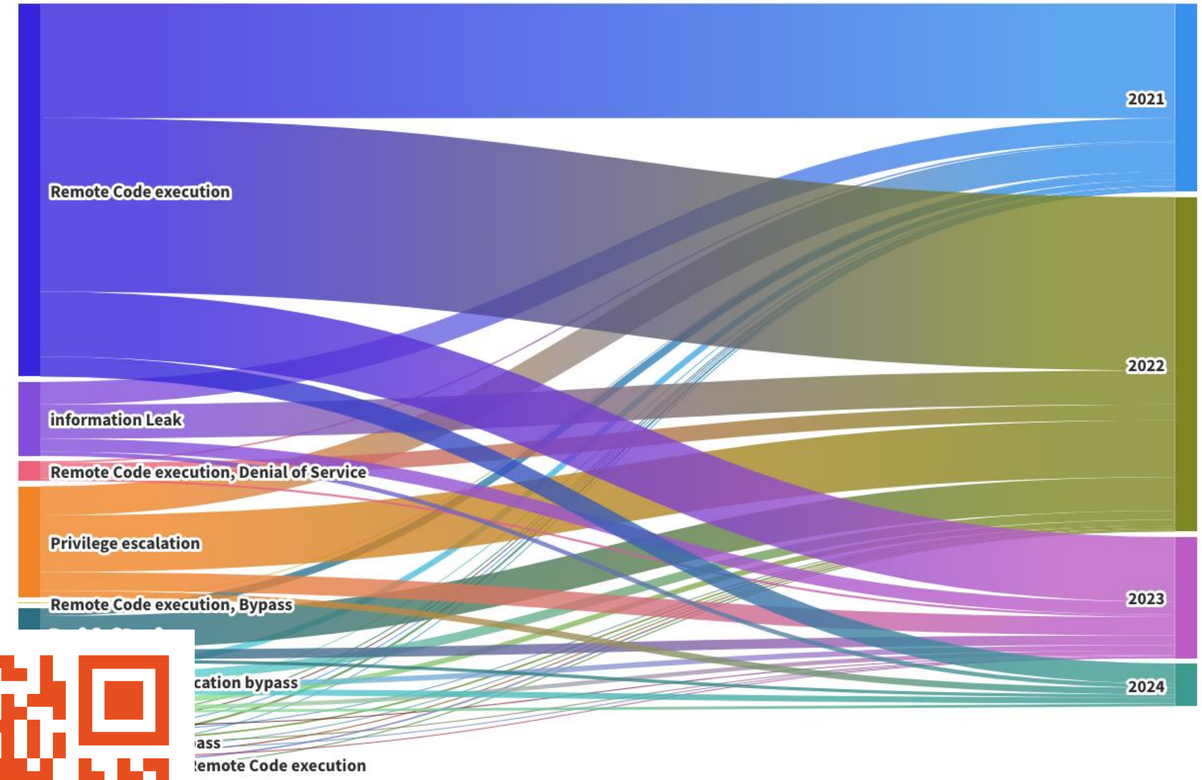


PHOENIX CTI - MOST USED ATTACK METHODS

TOP EXPLOITS METHODS

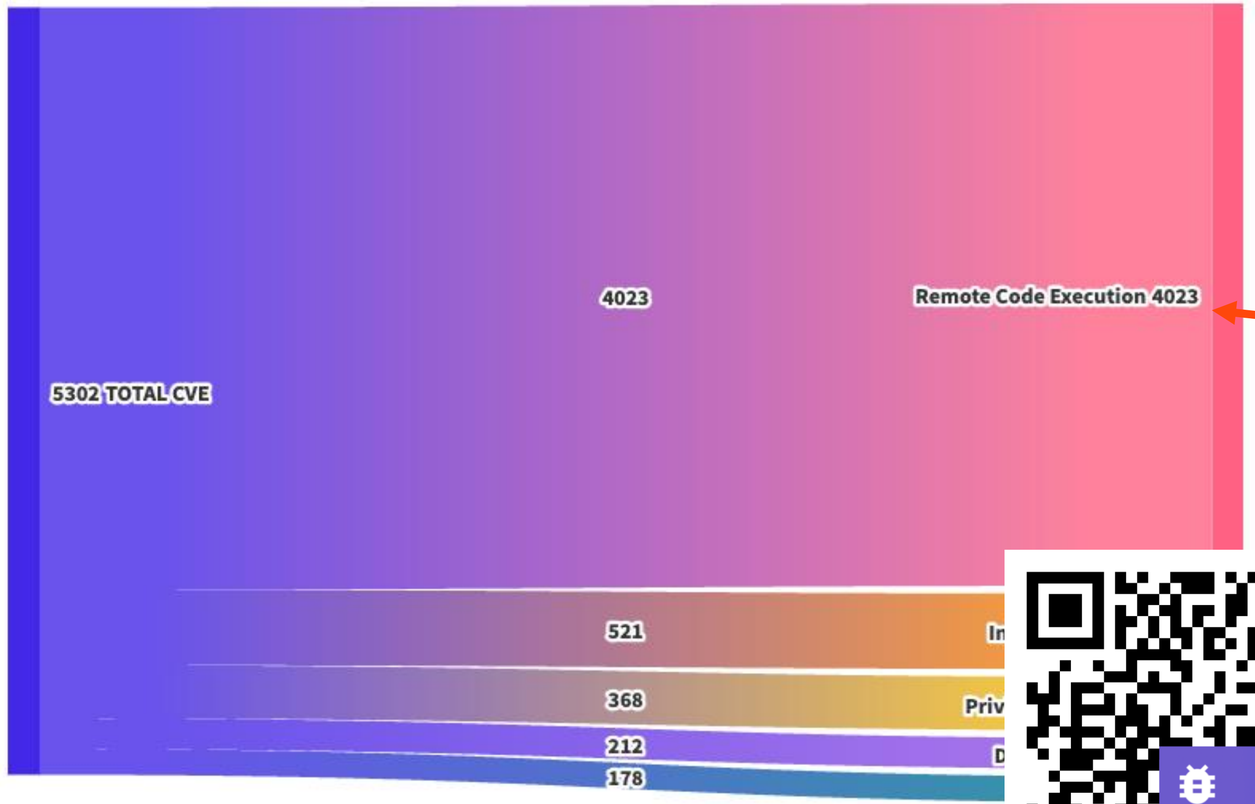


CISA KEV

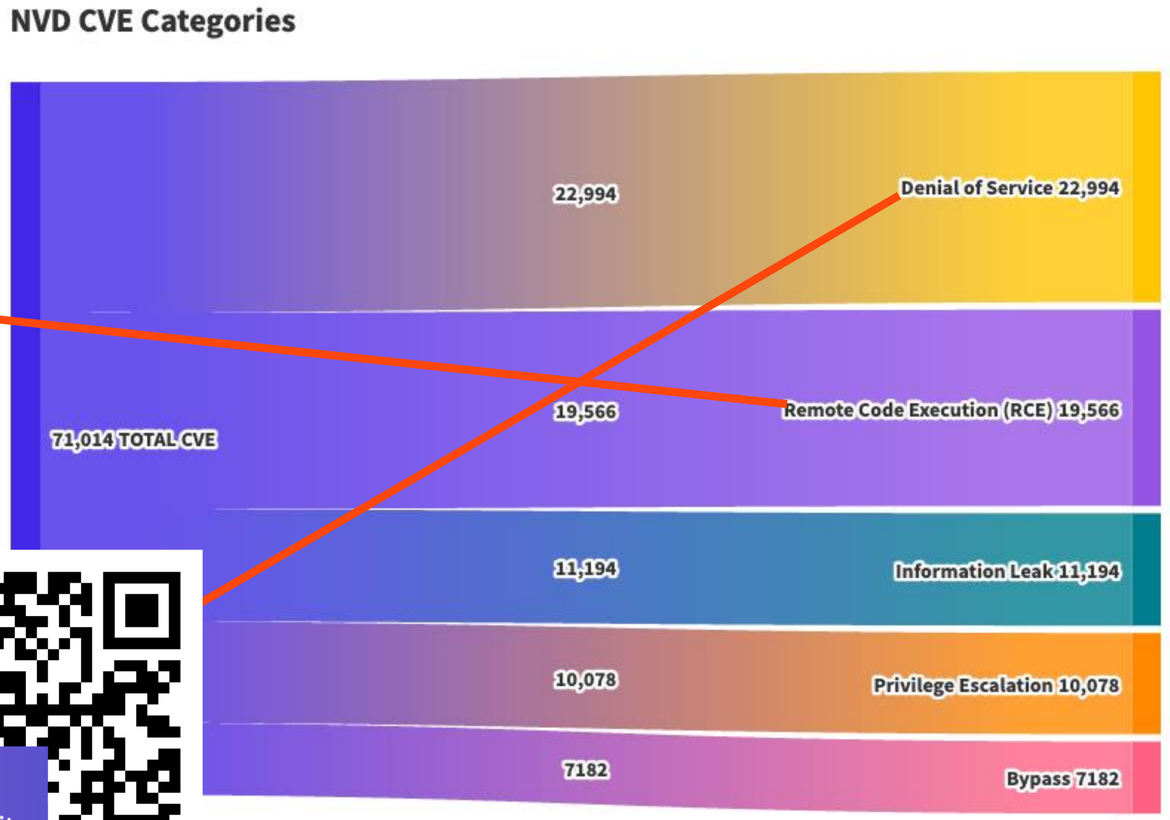


PHOENIX CTI – GITHUB POC – MOST USED METHOD

TOP EXPLOITS METHODS



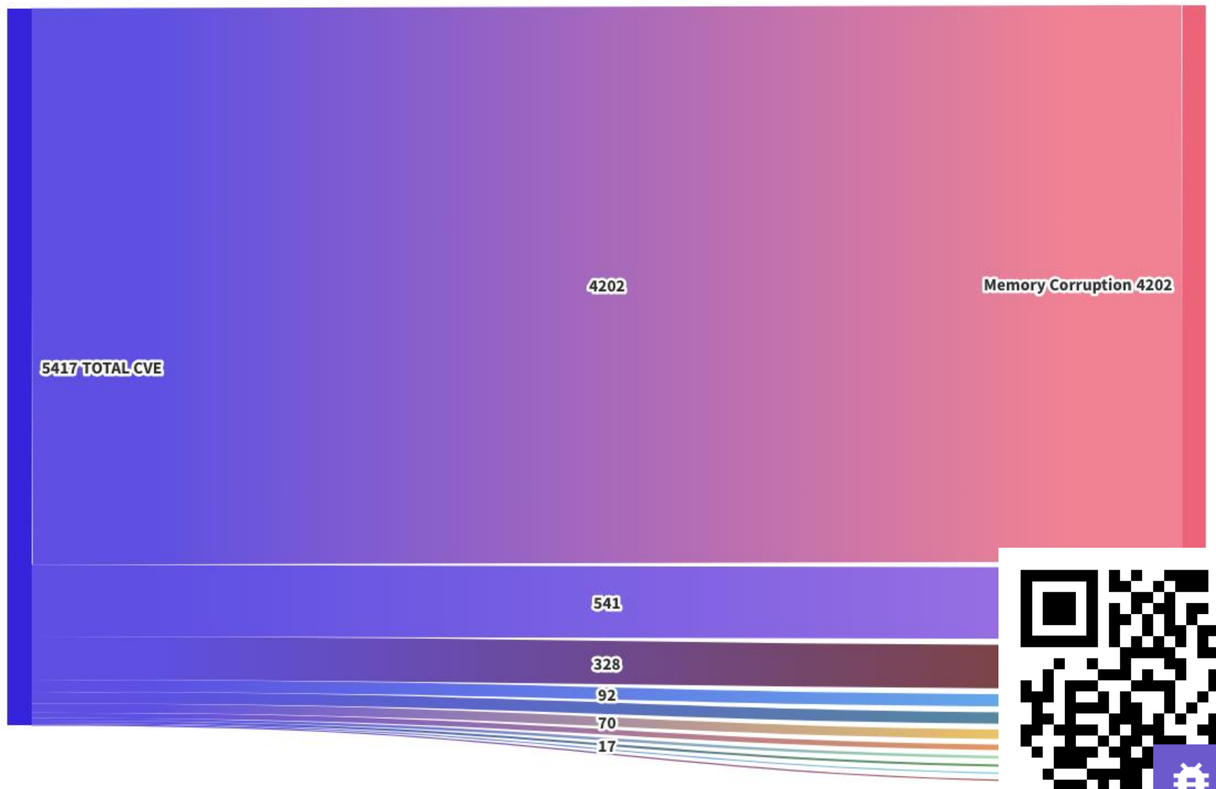
Overall NVD



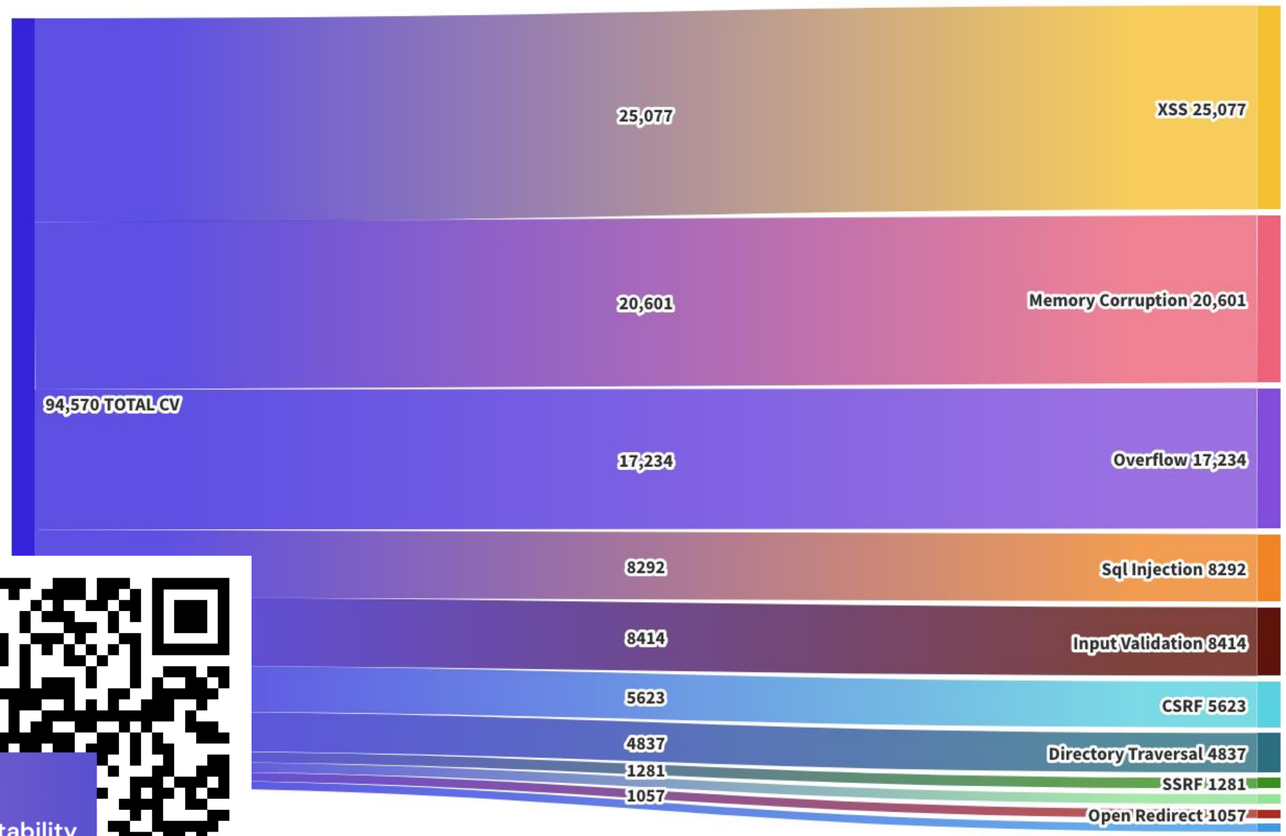
Exploitability

PHOENIX CTI – GITHUB POC – PREVALENT TECHNICAL IMPACT

TOP EXPLOITS IMPACT

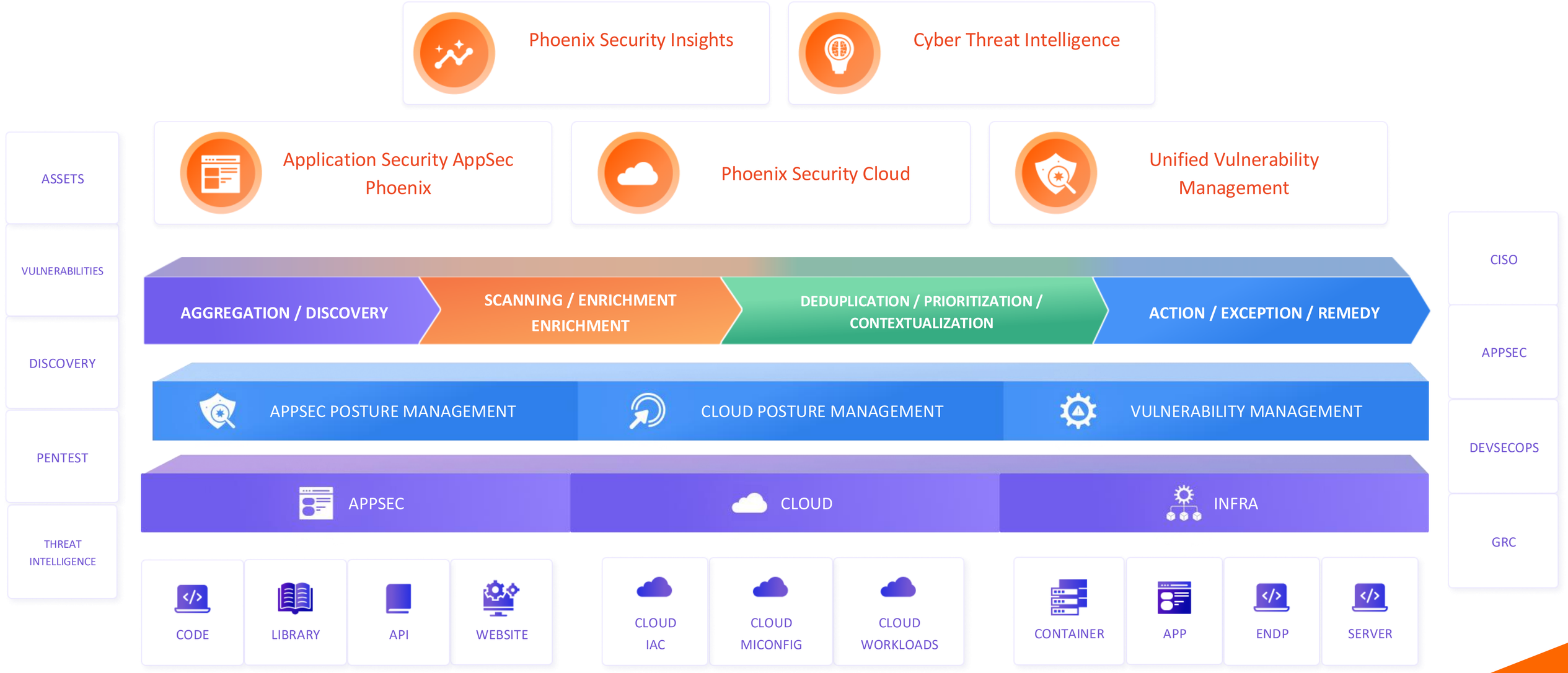


Overall NVD



Exploitability

Phoenix Security platform unifies risk across your entire attack surface



CONTEXTUALIZE PRIORITIZE | ACT ON RISK THAT MATTERS MOST

POSTURE MANAGEMENT ACROSS X SURFACES E(X)SPM

eXtended Surface Posture Management (XSPM)

- ASPM
- RBVM
- Cyber Asset Management
- Team performance tracking/ Attribution/ Traceability
- Correlation of asset across domains (library from SCA and container)
- Deployment traceability with canaries
- Cloud Based Vulnerability Management

EXTENDED SURFACE POSTURE MANAGEMENT XSPM

Secure Runtime, Application

in one view empowering business to make risk based decision actionable from engineers / developers

ASPM Application Posture management

Prioritize fixable cloud native application/ scanning and recheability

- Aggregation of multiple assets classes
- Deduplicate/ Correlate/Prioritize assets and vulnerabilities
- Attribution of team to code
- Traceability of application to cloud
- Prioritization based on deployment

Identify what's fixable based on the deployment of deployment of the application

EASM External Attack Surface Management

Scan your external attack surface and correlate with internal surface

- Correlation and contextualisation of internal and external
- Threat intelligence and prioritization of the vulnerabilities
- Correlation with application/deployemnt
- Correlation with application

Identify what's important to work on from outside in

Risk Based Vulnerability Management

Manage internal vulnerability with risk based prioritization

- Prioritize vulnerability using threat intelligence
- Aggregate asset classes and extract insight across multiple sources
- Dedupliacte, Correlate, cross domains
- Attribution and Application treceability

Trace application on prem-cloud and correlate threat inel

CSPM Cloud Security Posture Management

Prioritize internal vulnerability in the cloud and create internal/external attack surface

- Traceability of applicaiton to cloud deployment
- Conceptual segmentation of production
- Correlate Container and cloud pre-post flight
- Transfer insight cross domain (e.g. recheability of an application

Trace application on cloud-cloud and correlate threat inel

Removing Manual work to automate, scale effectively security teams

$ 1.780 K PROGRAM COST	1800 DAYS	»	226.6 K PROGRAM COST	150 DAYS

	Without AppSec Phoenix		APPSEC PHOENIX 7X CHEAPER 12X FASTER	
DESCRIPTION	COST	TIME	COST	TIME
TOTAL	**$2,983.00**	**24h**	**$376.00**	**2h**
Export of report/ Vulnerabilities	$56.00	30 min	$0.00	0 min
Notification to Security professional	$3800	20 min	$0.00	0 min
Analysis of reports by DevSecOps	$600.00	320 min	$59.38	15 min
Perform Vulnerability Assessment	$375.00	200 min	$59.38	15 min
Contact the business owner and assess the importance of the application	$375.00	200 min	$0.00	0 min
Research exploitability from different databases & Calculate Vulnerability Matrix	$375.00	200 min	$118.75	30 min
Select subset vulnerabilities to execute across platforms	$338.00	180min	$0.00	0 min
DevSecOps Follow-up with developers on schedule and resolution of vulnerabilities. Assume 1 DevSecOps and 2 devs for 2 meetings	$713.00	180 min	$119.00	30 min
Monitoring resolution of vulnerabilities & follow up on targets with DevOps Teams	$113.00	60 min	$19.00	10 min

*DevSecOps average daily rate 500$,
Dev average daily rate 300$