

AI Second – Threat Centric Agentic approach on Vulnerabilities

Applying AI SECOND –
- Because terrible data in → terrible data out – just prettier

Warning revelation ahead



I've seen things you people
wouldn't believe.

CAUTION: BAD DATA



BAD DATA QUALITY
MAY RESULT IN
FRUSTRATION AND
LEAD TO DROP
KICKING YOUR
COMPUTER

Warning



About the speakers



This presentation was not LLM generated

No LLM were hurt generating this presentation

Francesco Cipollone

CEO & Co-Founder Security Phoenix, Board CSA UK



**Who Does what where is more important than raising vulnerabilities
Human Centric approach to LLM, AI Second approach**

**LLM - AI / We need a Human Centric
approach to this tool**

**This talk is centered on an AI Second
HUMAN FIRST initiative on LLM/AI**

For anything more detailed about Phoenix talk to Phil



Beyond his client work, Phil actively contributes to the industry as President of the Cloud Security Alliance's Charlotte chapter. He is also the author of Friday Tech News, a widely-read weekly newsletter that curates the latest developments in technology and cybersecurity for IT professionals and business leaders.



Agenda

Intro & Context

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

P1 - Challenges in prioritization with old metrics a practitioner story

P2 – Attributing the right problem to the right team in the right context – Reachability Analysis

P3 – AI based - Threat Centric approach on vulnerability for prevention

Conclusion & Q&A

Hands up if ..

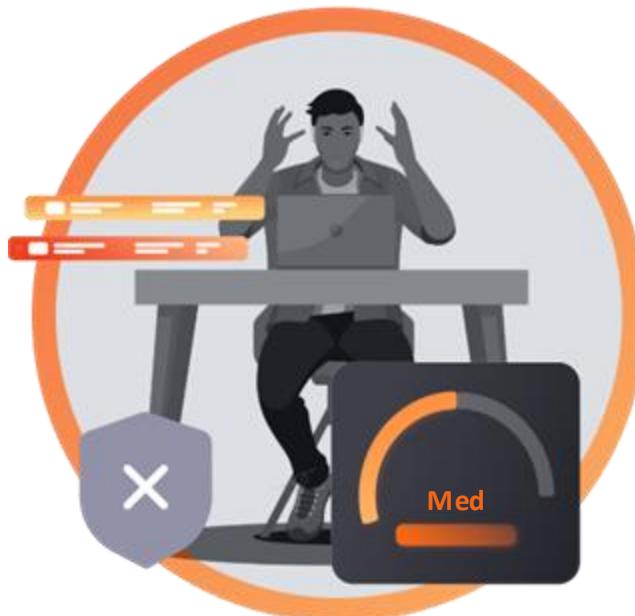
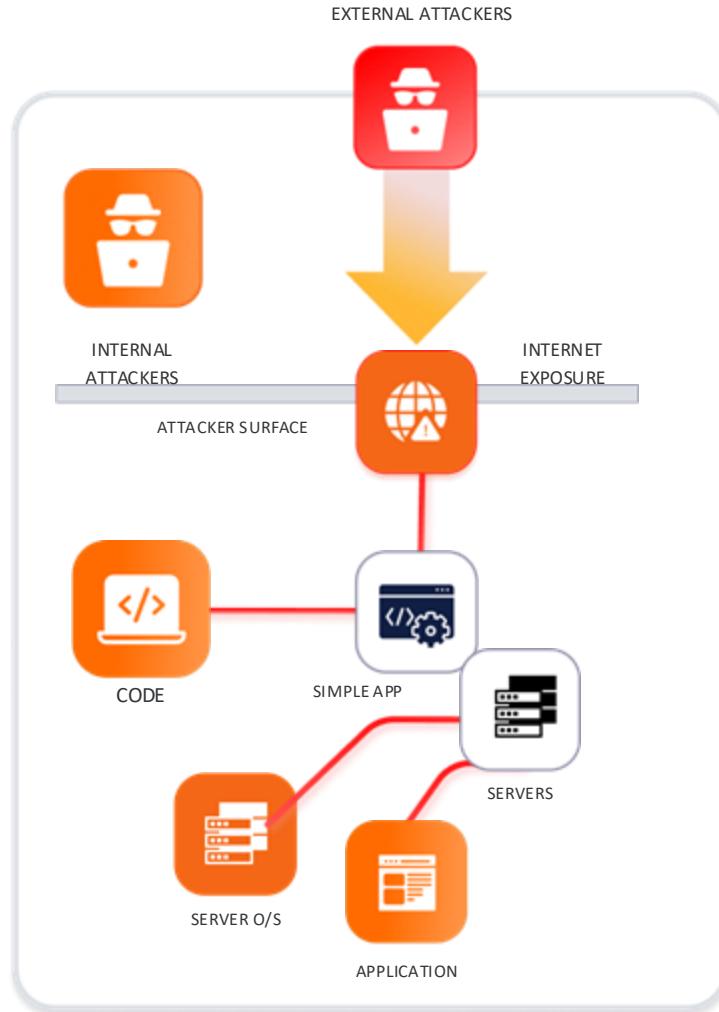
You believe your company has a functional vulnerability management program

You believe it covers all areas and scope of vulnerabilities

You are attributing all the assets to the right team

We don't believe you. Let's see if you still think so by the end.

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

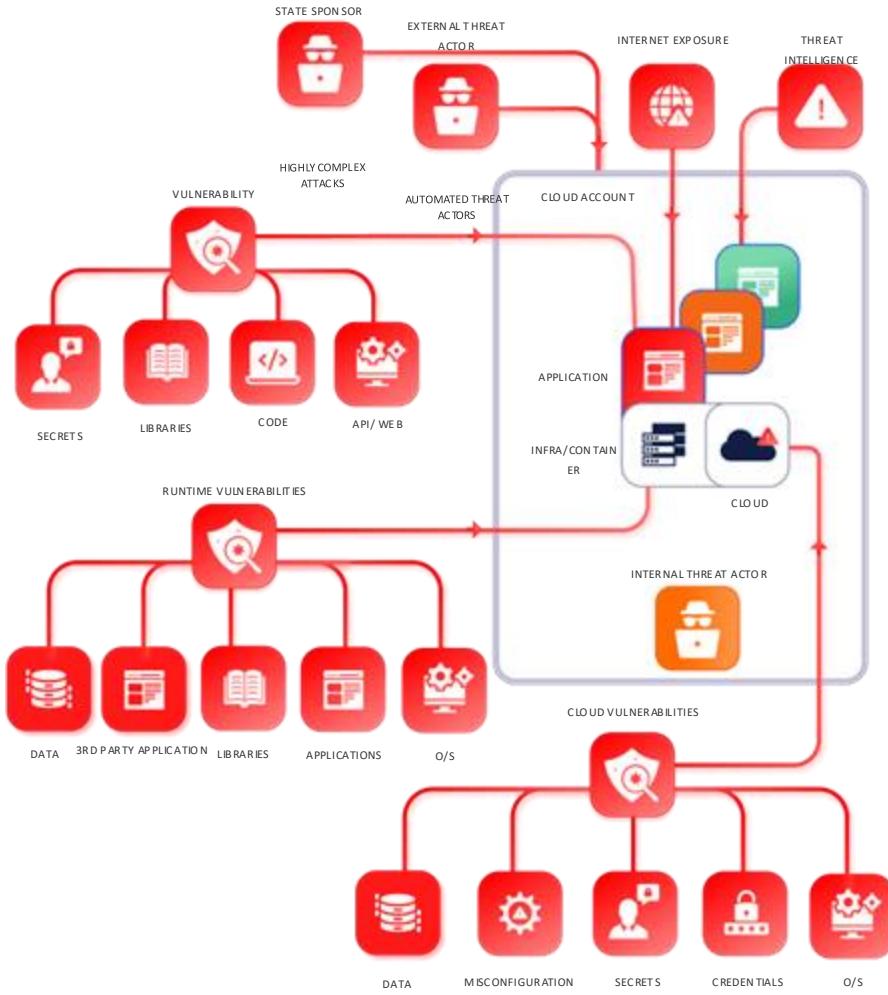


! Total Number of CVEs:
15 K (now 222 K+)

! 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**

Larger software attack surface built by multiple teams releasing frequently

I feel your pain



Alert FATIGUE No COMMON LANGUAGE = BURNOUT

THREAT INTELLIGENCE & CONTEXT



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%

Budget increase for
security team down from
17% ***

262 950

2024

~ 40 X



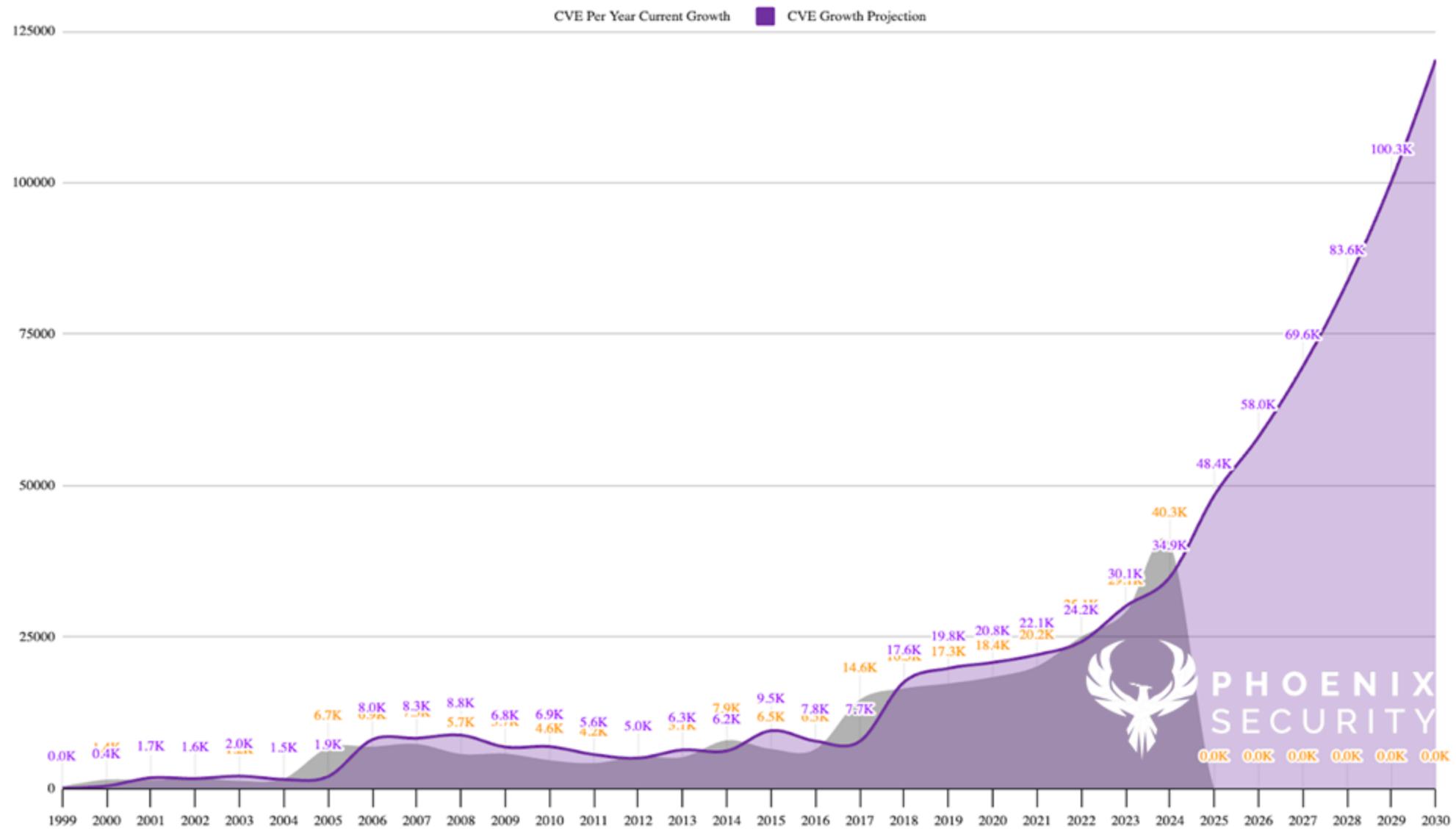
*FIST / EPSS

** NVD/CVE STATISTICS

*** UK GOVERNMENT

Road to 1M Vulnerabilities

CVE Projection Per Year



CONTEXTUALIZE | PRIORITIZE | ACT ON RISK THAT MATTERS MOST

© Phoenix Security 2024

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 bytes*	●	○	●	No	Jailbreak prompts
MakerGPT [49]	10 bytes*	●	○	●	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

* Bytes is the forum token of [hackforums.net](#); ● indicates implicit mention.

<https://arxiv.org/abs/2401.03315>

AI-generated bug reports are becoming a big waste of time for developers

Better sounding cr**** s still hallucinated Cr**** curl project creator states

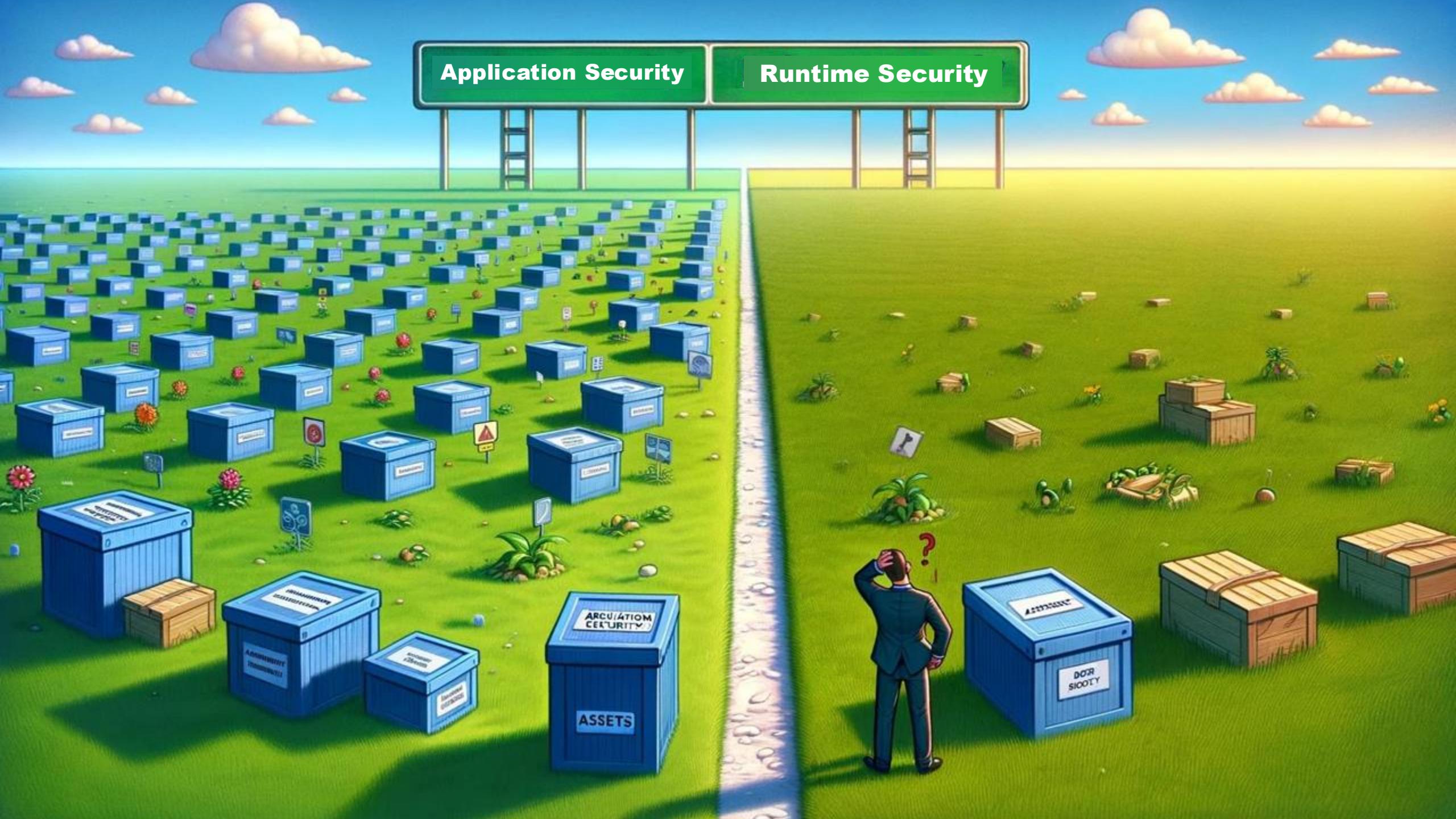
By Alfonso Maruccia January 8, 2024 at 12:36 PM

CVE-2024-27198 Vulnerability Timeline | March 4th



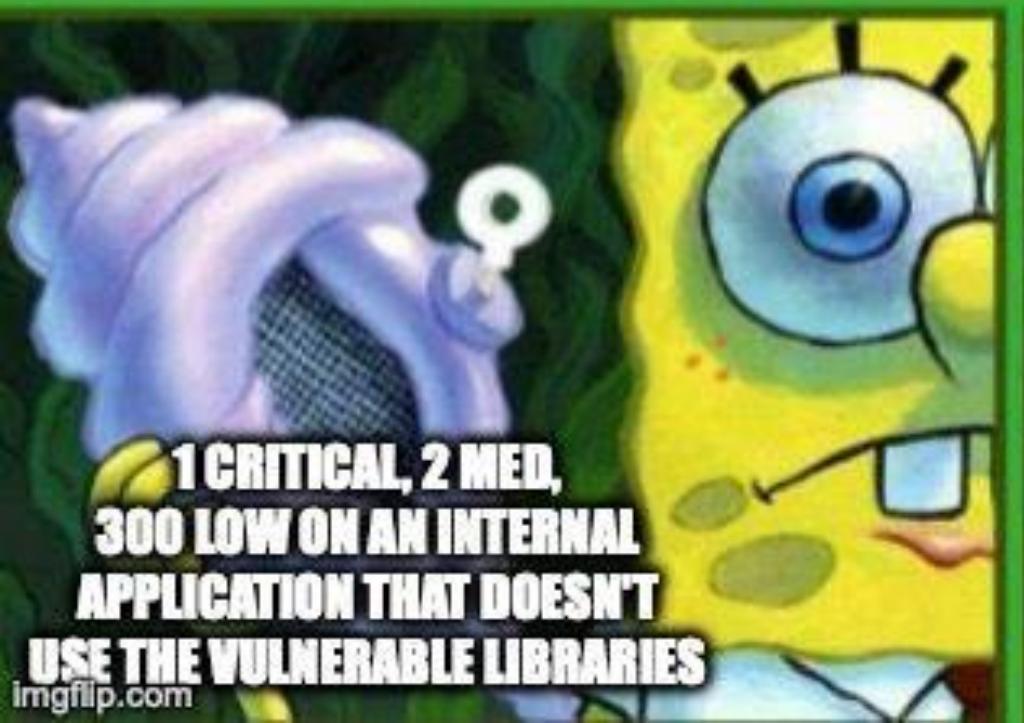
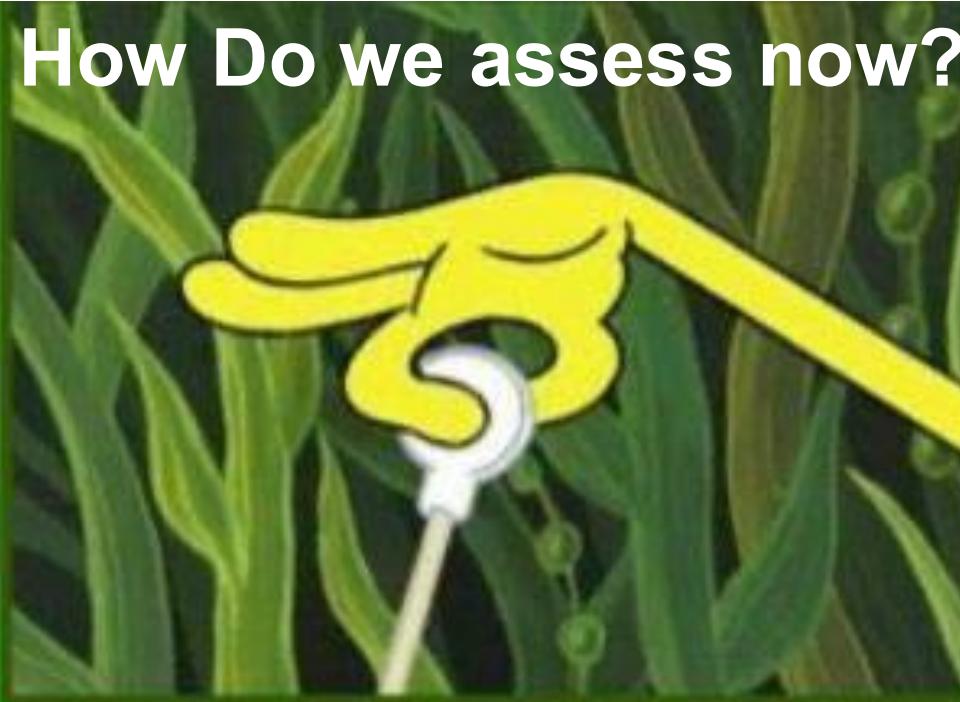
← 3 Minutes** →

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





How Do we assess now?



The Vulnerability Cycle



Step 1 – Overload Dev

Step 2 – Pray they catch that 1 vulnerability

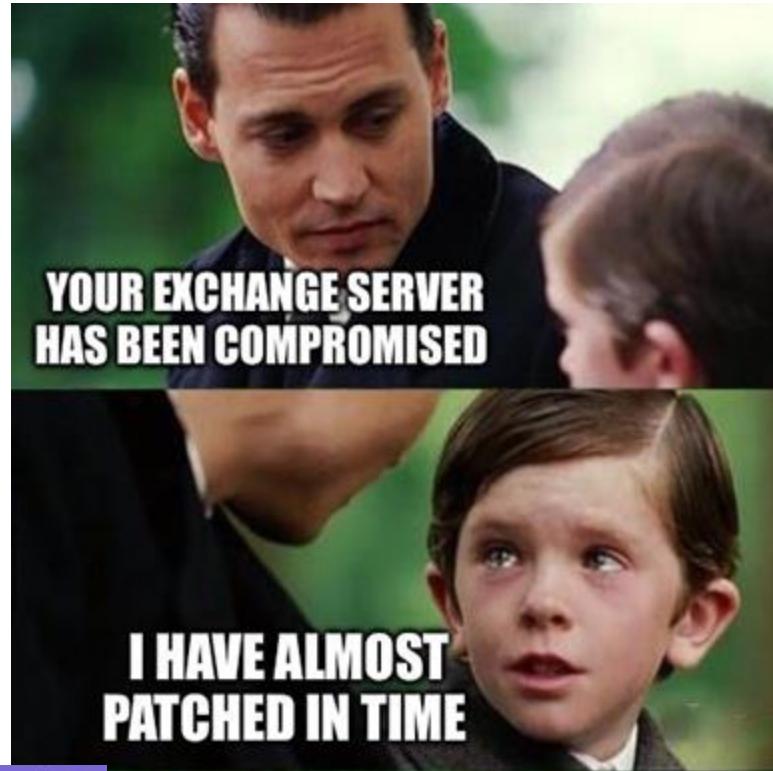
Step 3 – That 1 vulnerability get compromised

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**

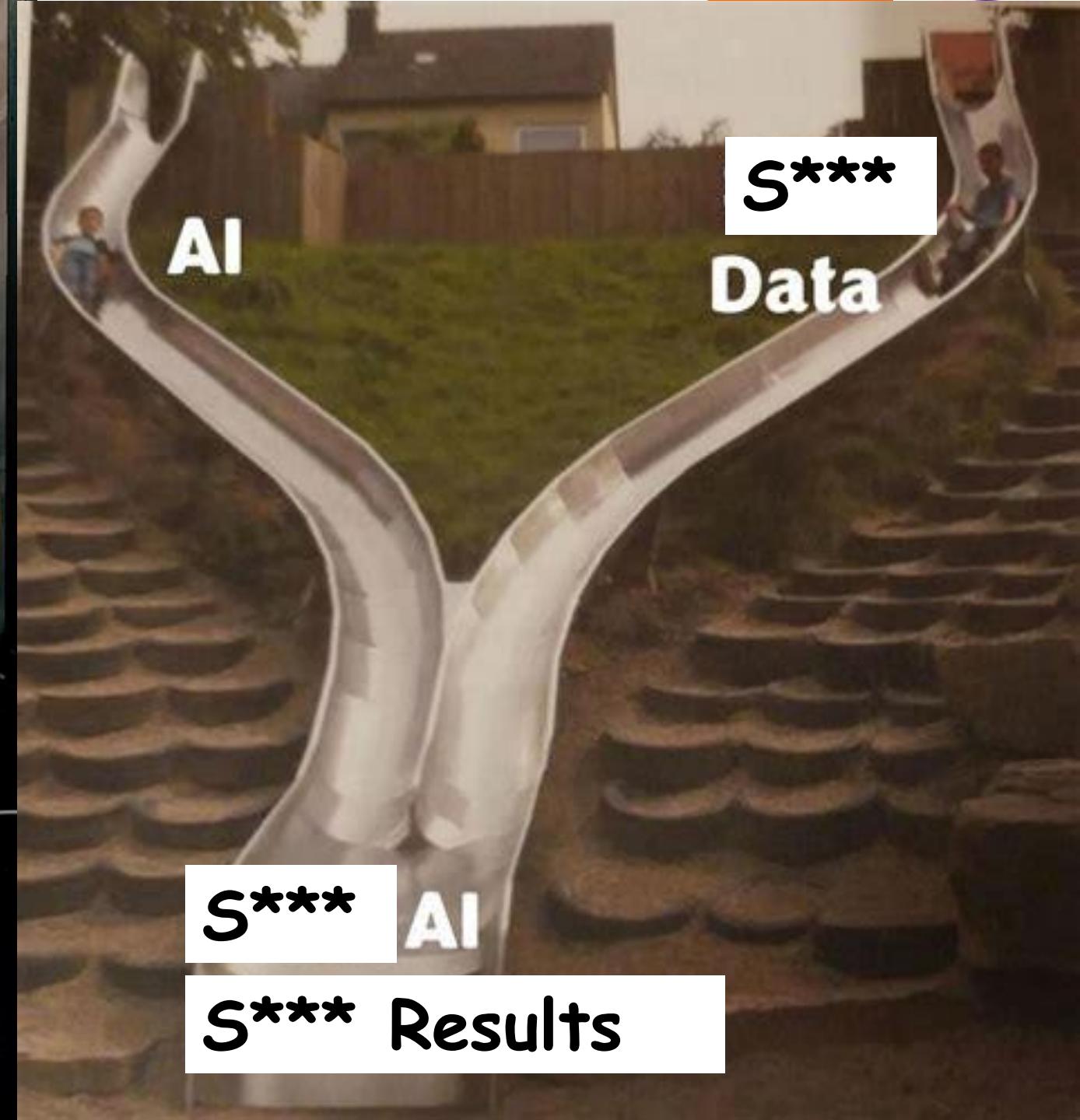
Questions we should be answering

**WHO does WHAT where
and how IMPORTANT is it**

**But really... can we use AI/LLM
now?**



**But daaad...
can we use
AI/LLM now?**



Can we use AI/LLM now?



But I'm here to give you the silver bullet

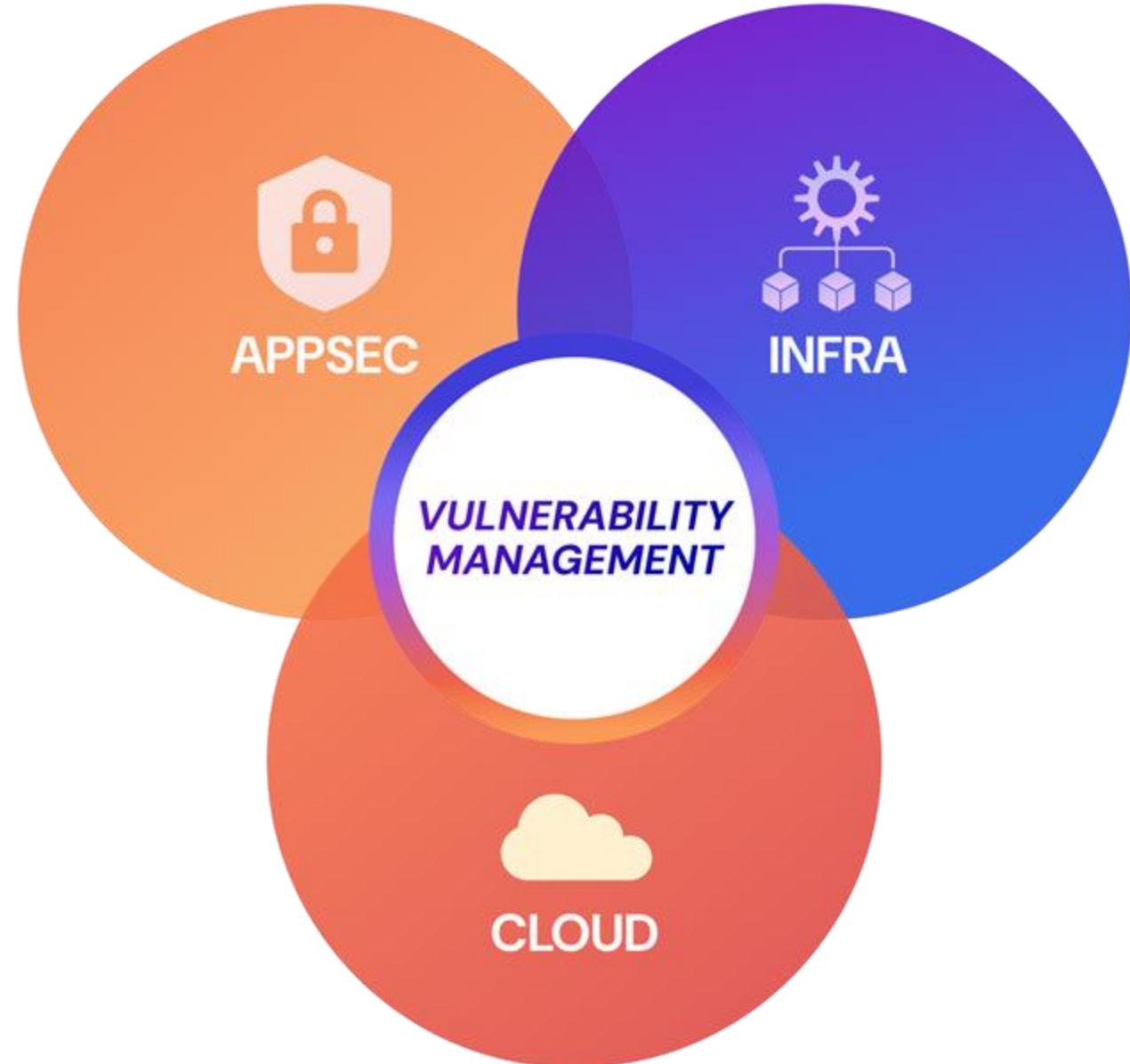


THE QUICK FIX!

What quick fix?



AppSec, Infra,
Cloud, etc., are all
part of the same
Vuln Management
Exercise



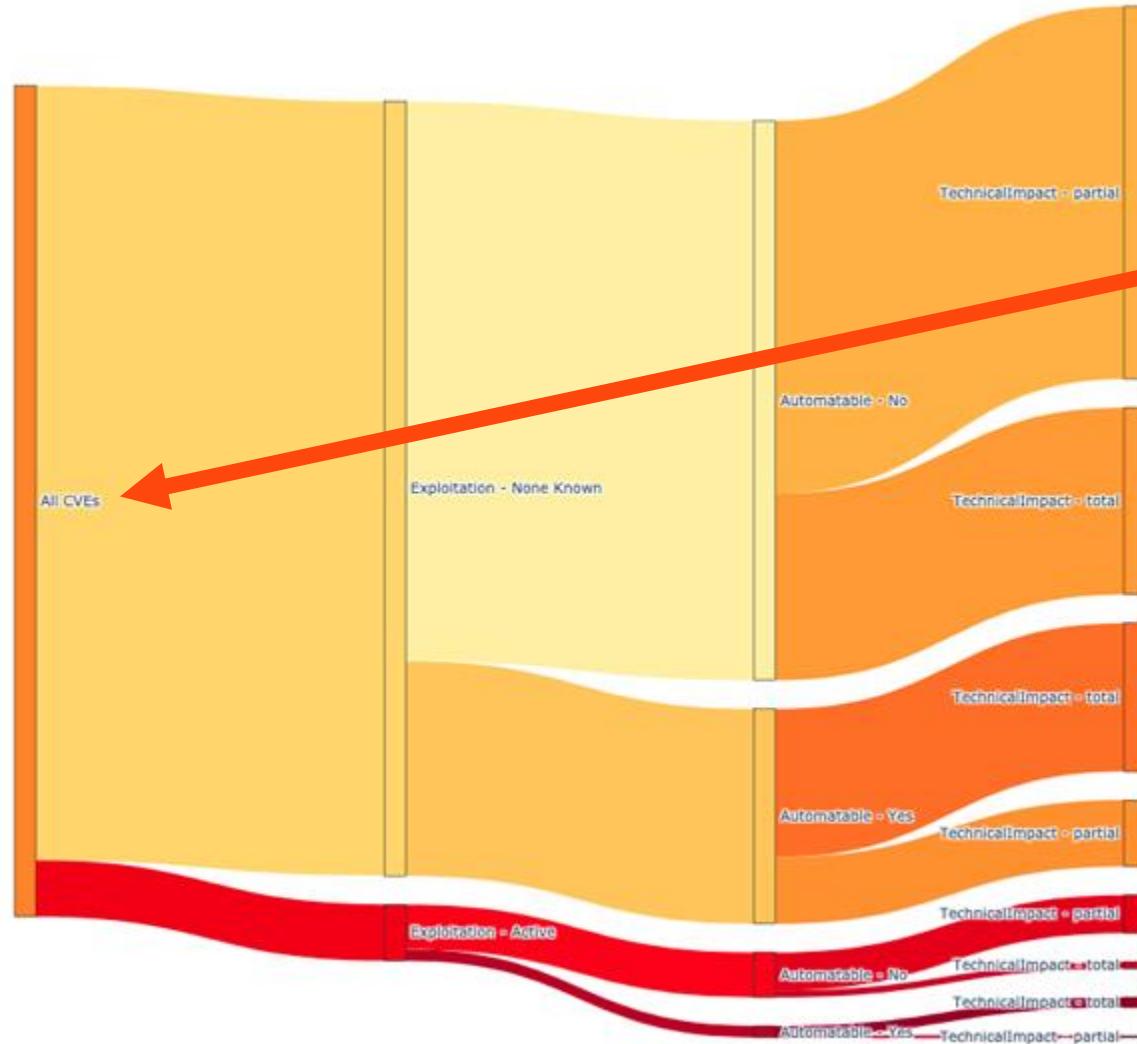


Part 1 - Identify what
to fix first IS
COMPLEX

Current Flow of vulnerabilities only 1% are exploitable



All CVEs DT Sankey Diagram



Can we apply AI here yet?

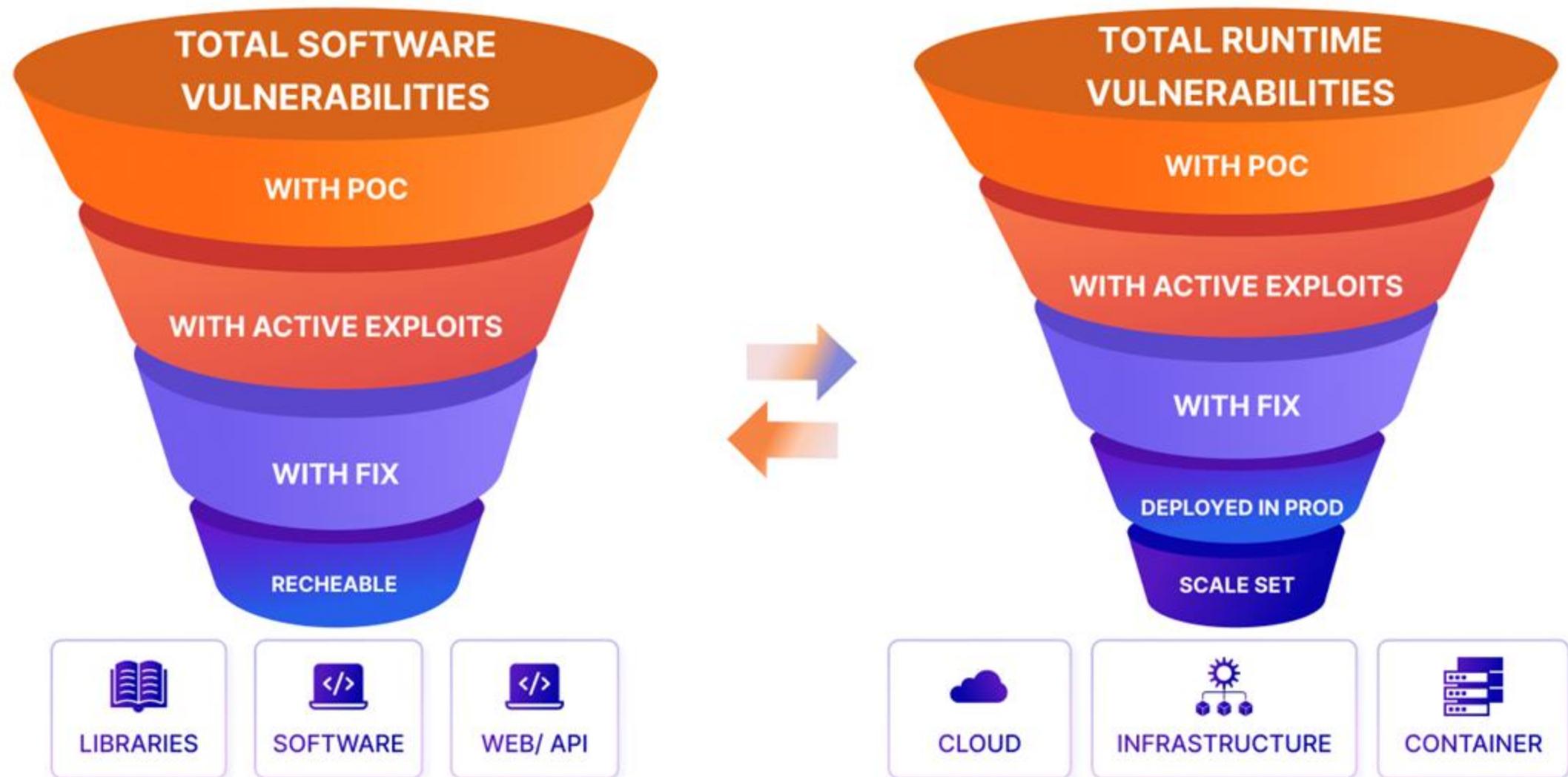
Current Focus

Really important to focus on

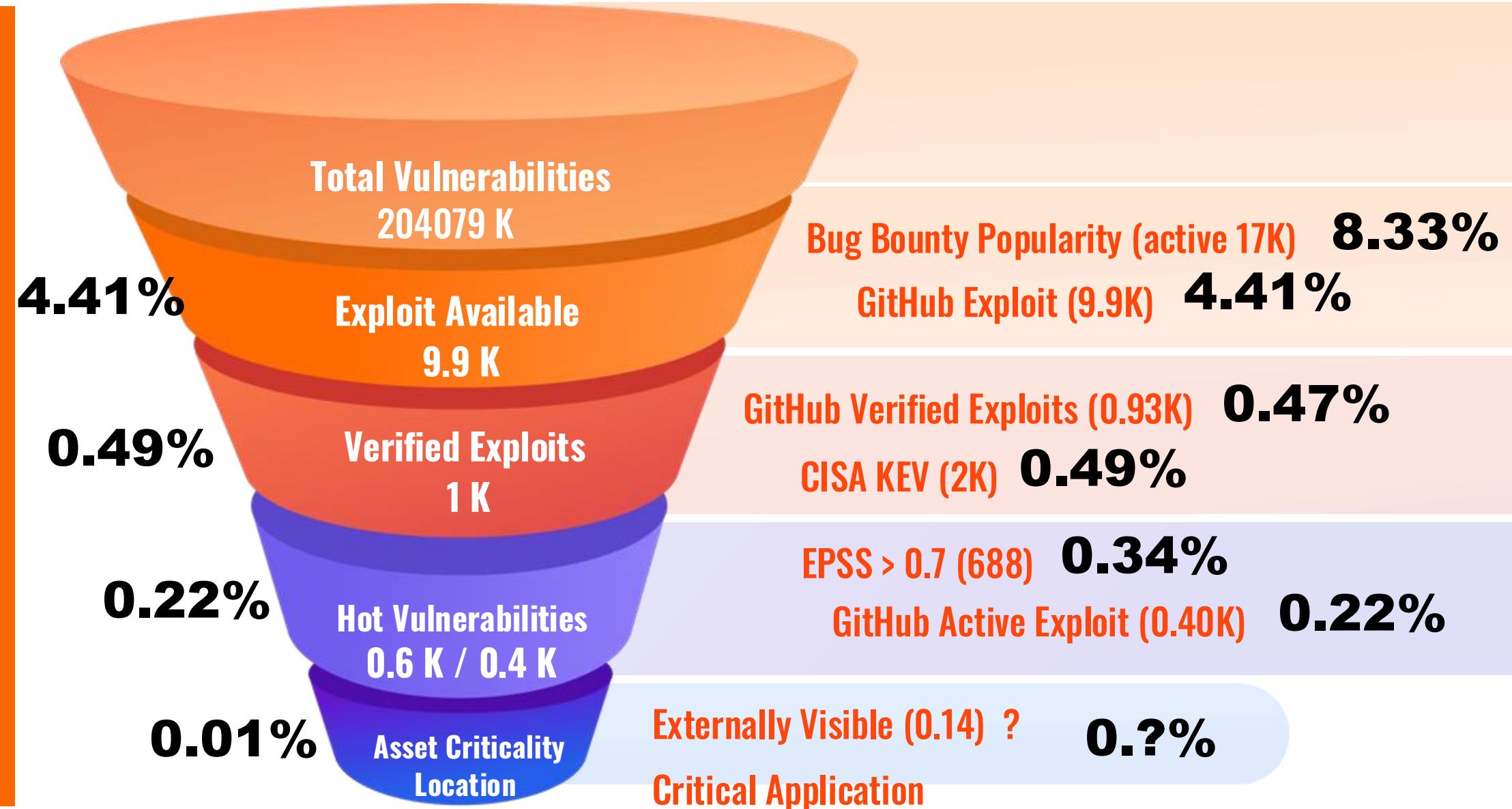
A large, abstract graphic on the left side of the slide consists of several overlapping, semi-transparent orange and red polygonal shapes. These shapes are roughly triangular and have sharp, angular edges, creating a sense of depth and movement. They are positioned in the upper-left quadrant of the slide.

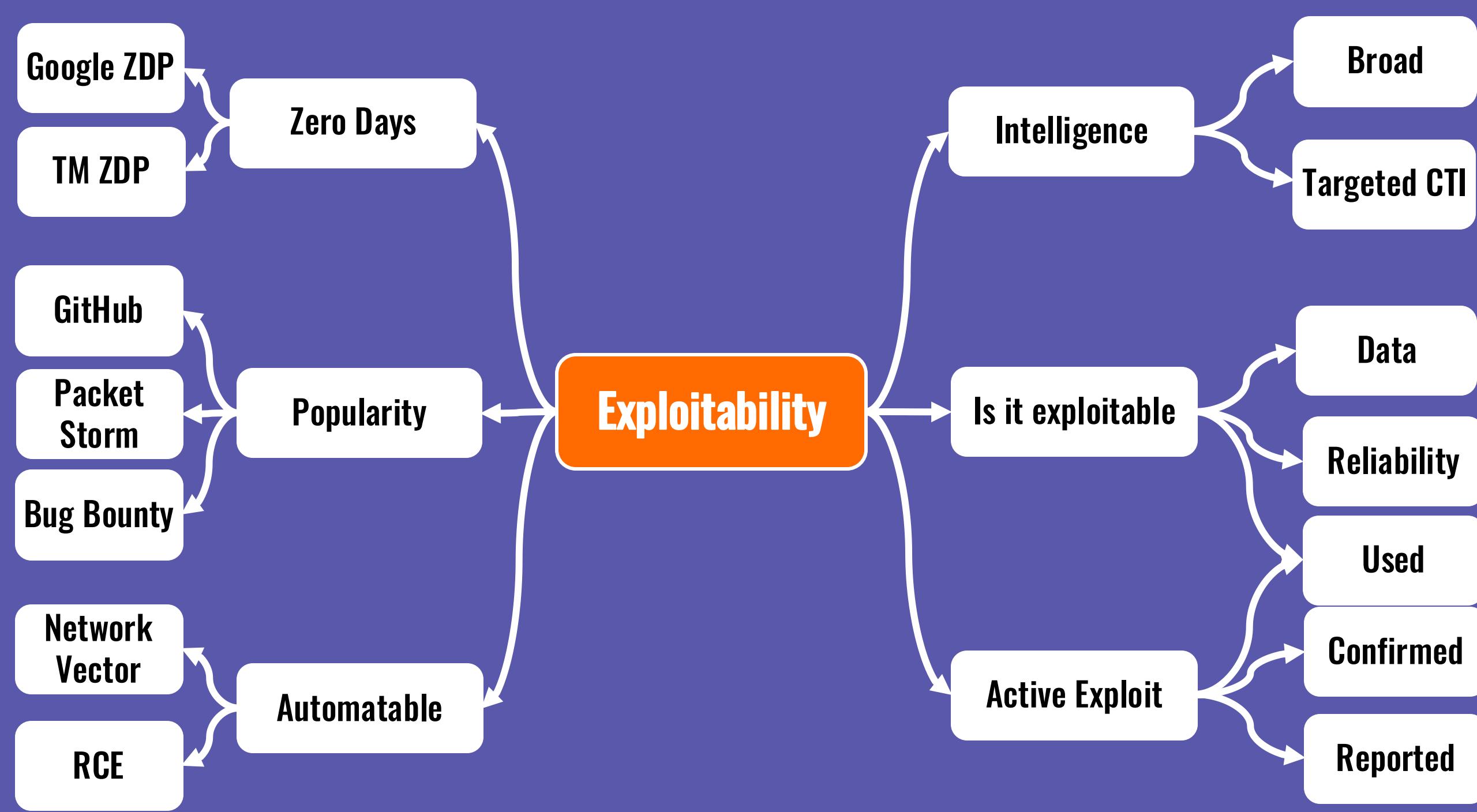
Part 1 - Attributing the right vulnerability with right context

Common Root Cause

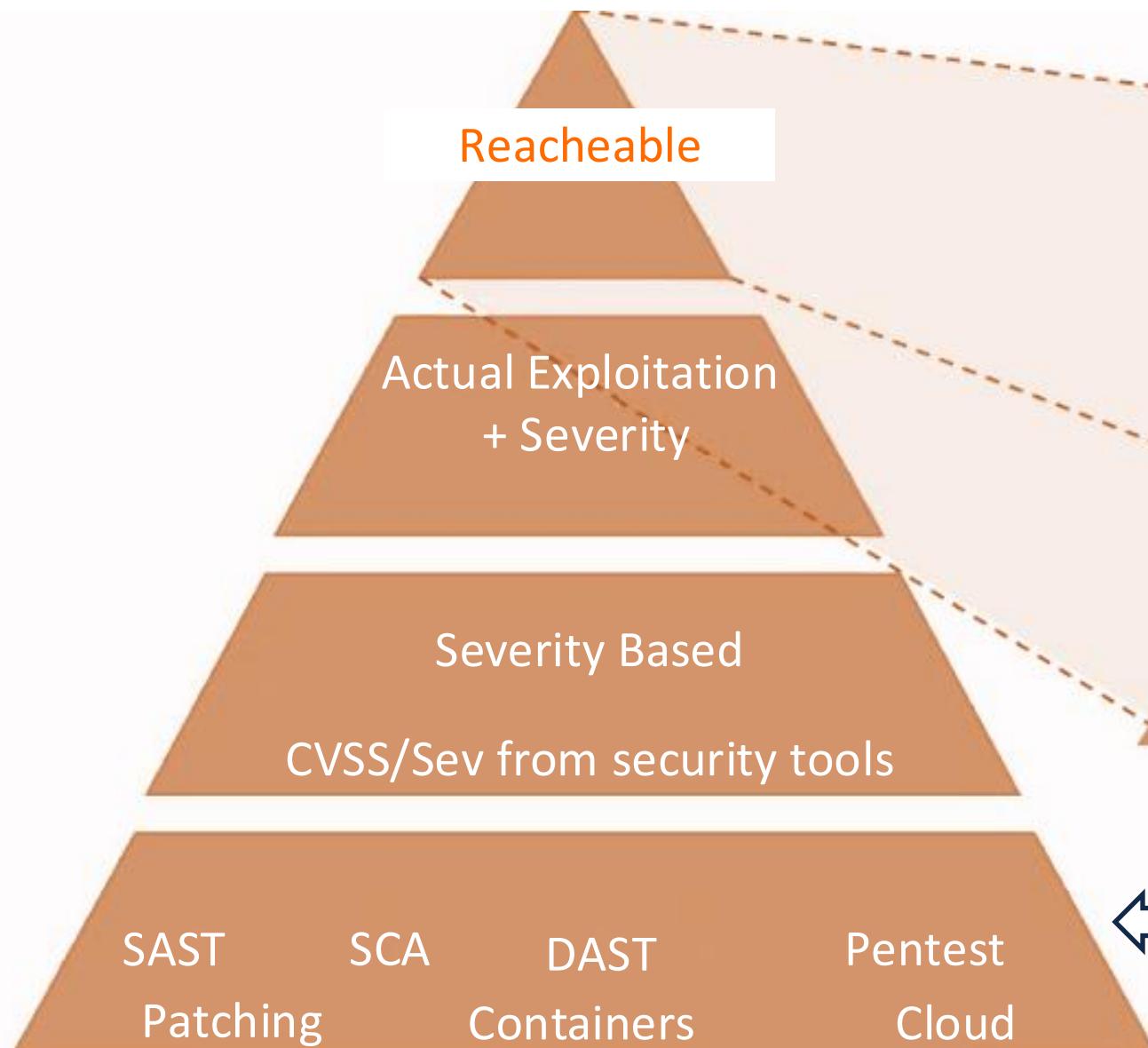


Not all the vulnerabilities require equal attention

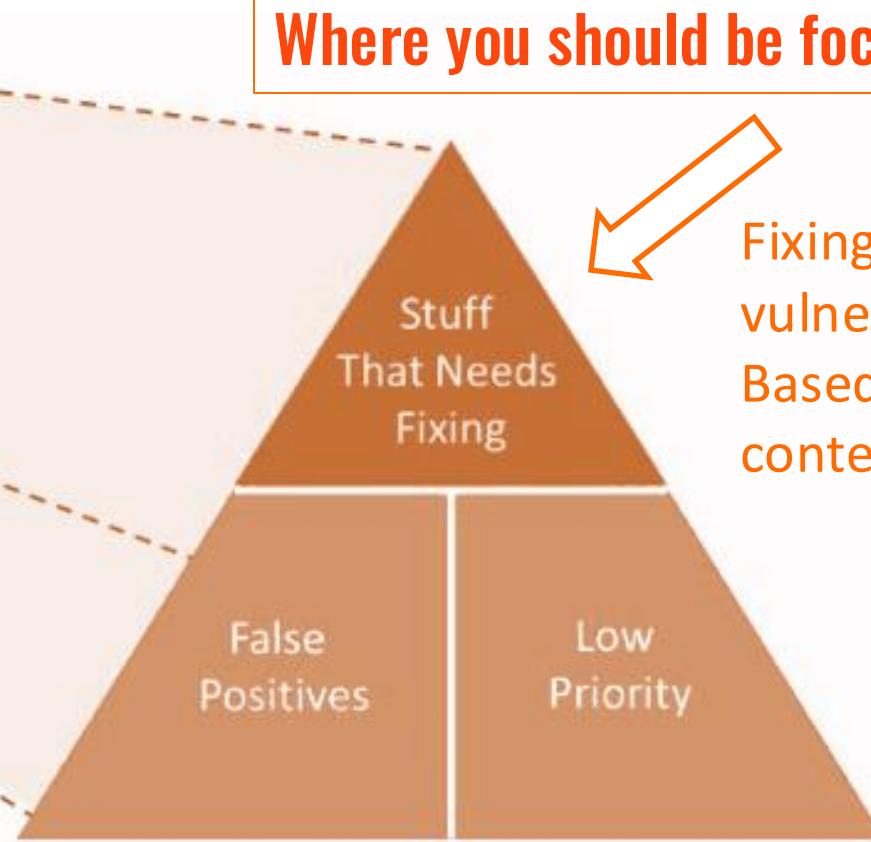




Prioritization is so 90....



Where you should be focusing

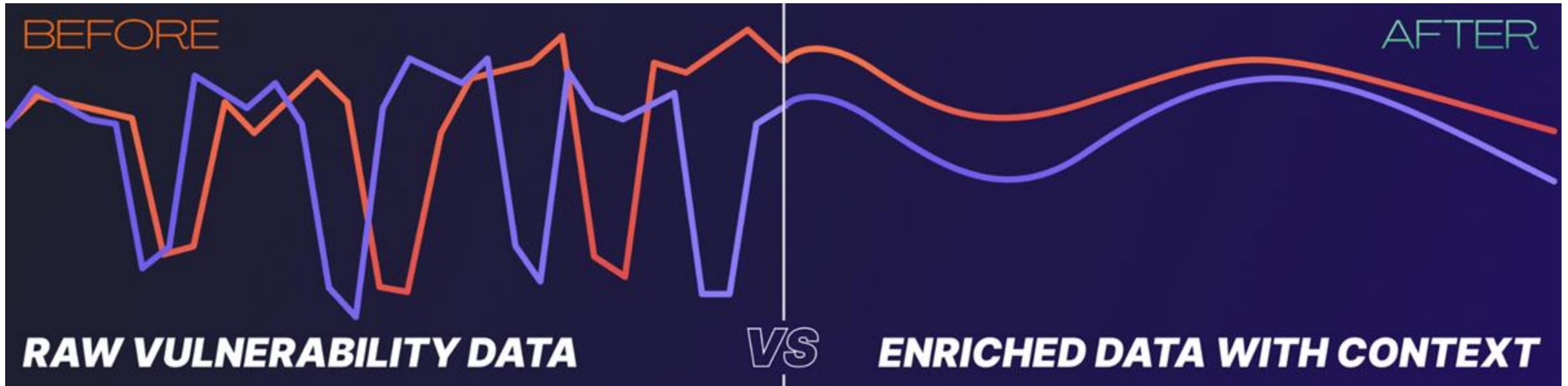


Fixing vulnerabilities
Based on contextual risk



Direct Output from tools

raw vuln data vs. enriched data with context.



But most important: Who owns it, is it important, when does it need to be fixed

Can we use AI/LLM now?



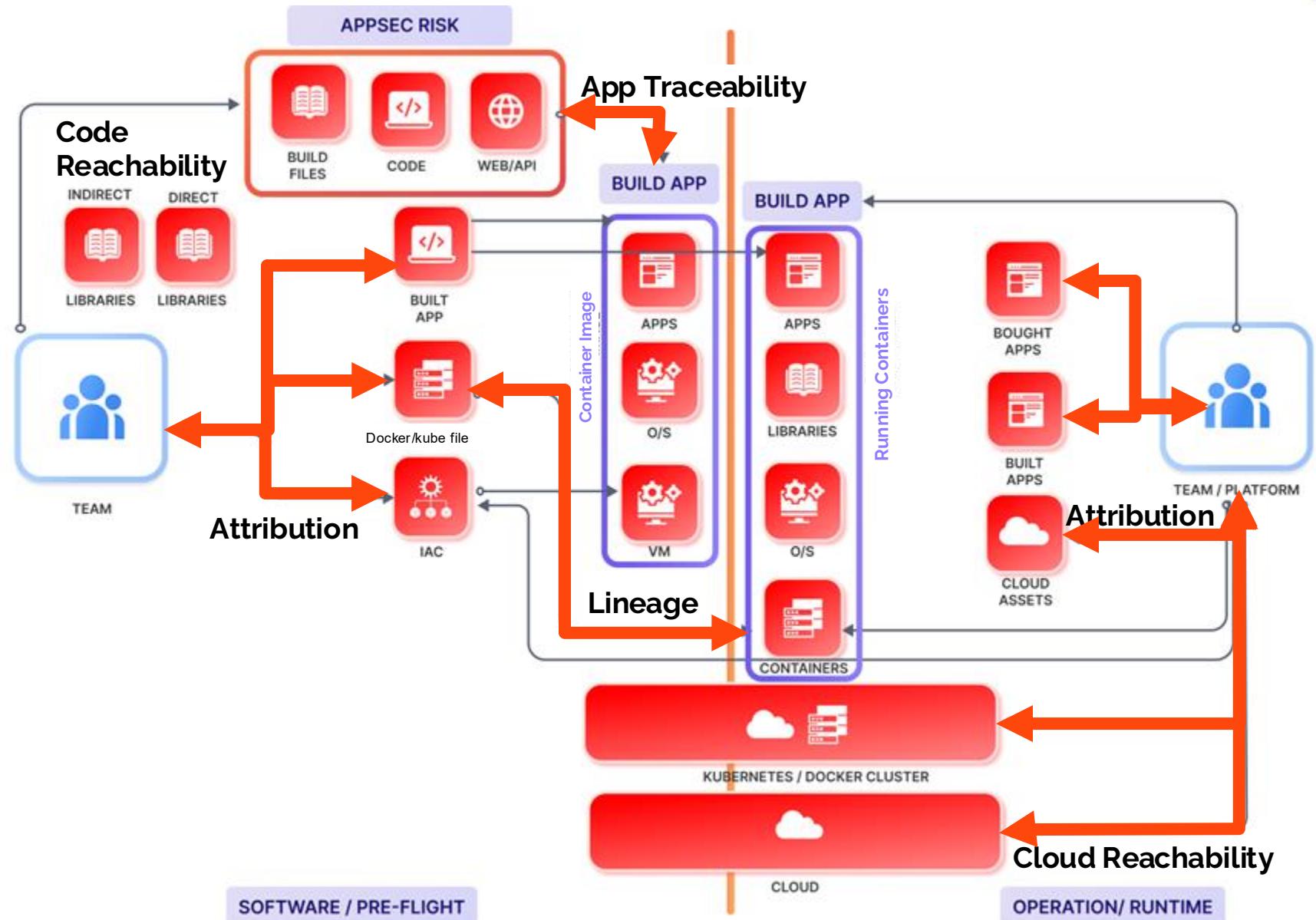
A large, abstract graphic on the left side of the slide features a series of overlapping, curved, orange-colored shapes. These shapes resemble stylized flames or leaves, with some having sharp, angular edges and others being more rounded. The colors range from bright orange to deep red, creating a sense of warmth and motion.

Part 2 - Attributing the right vulnerability team in the right context

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

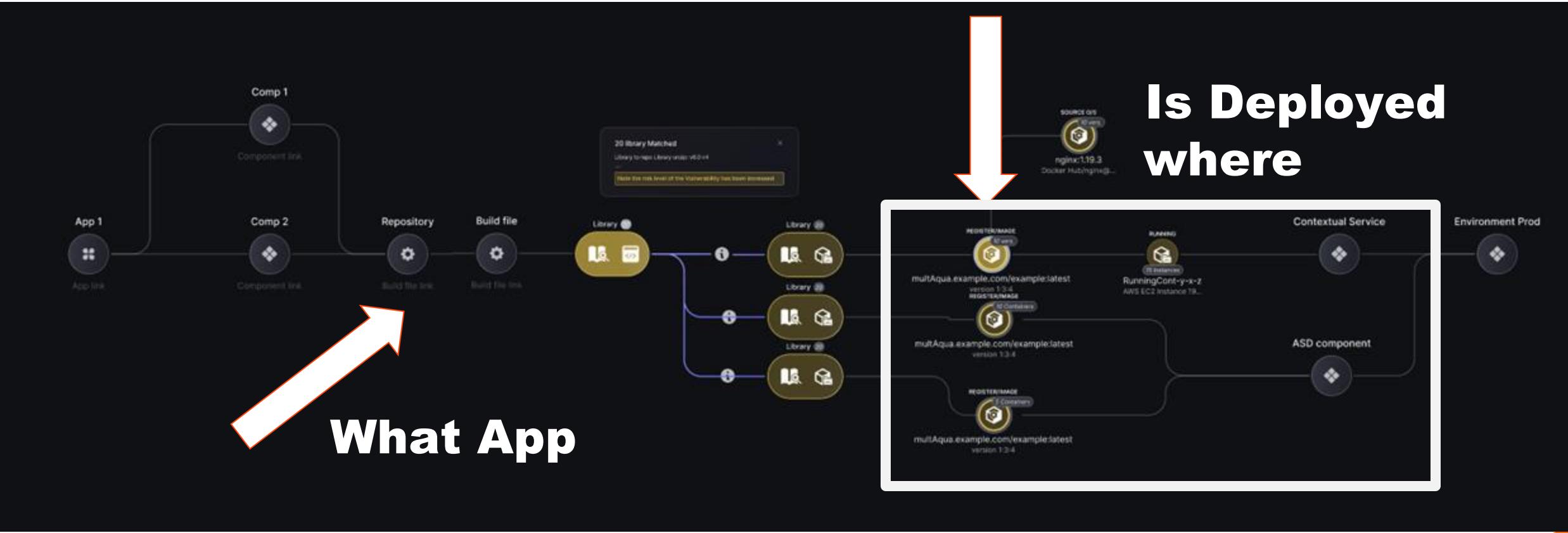


- Attribution
- Lineage
- Traceability
- Code/Cloud Reachability



An example of complexity Container fixing :

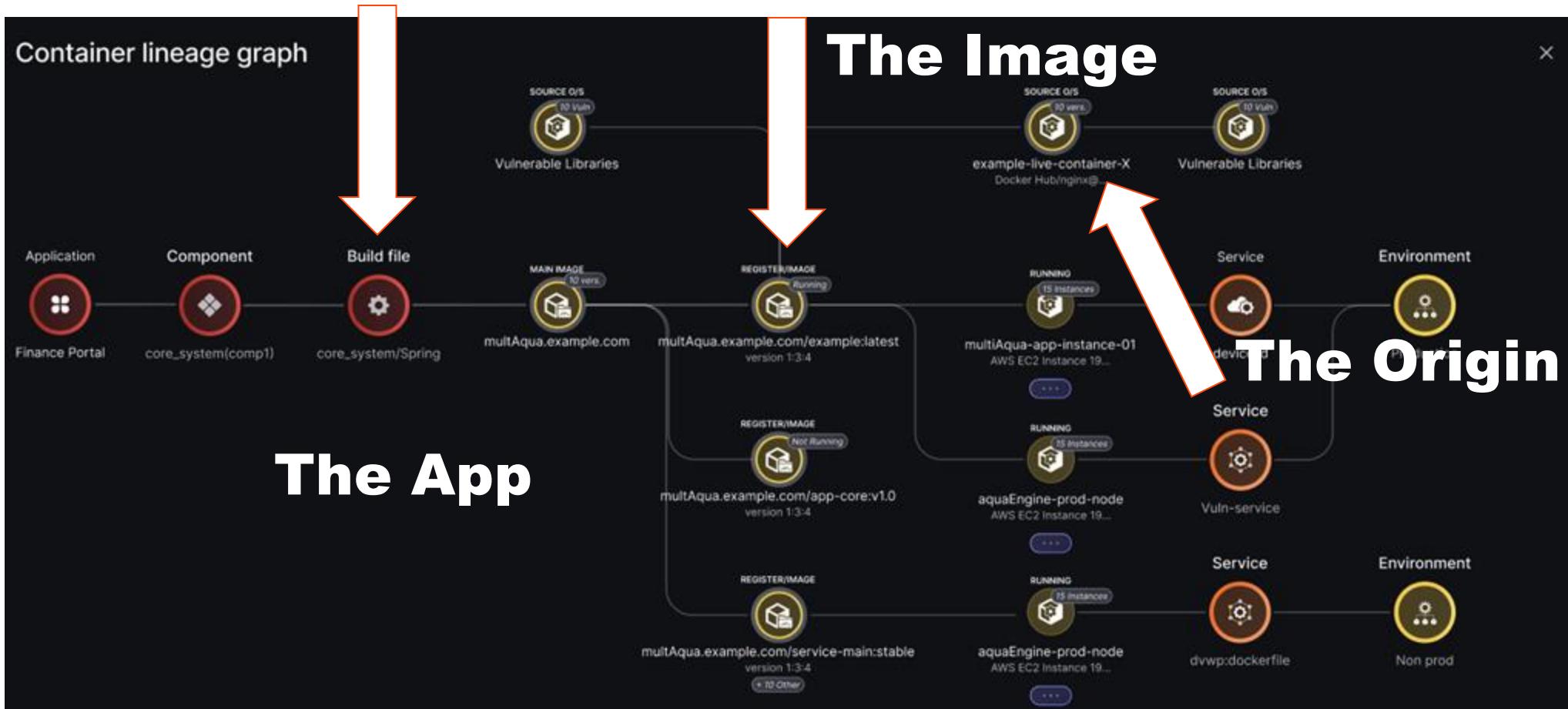
Libraries that are deployed : fix in the library vs fix in containers



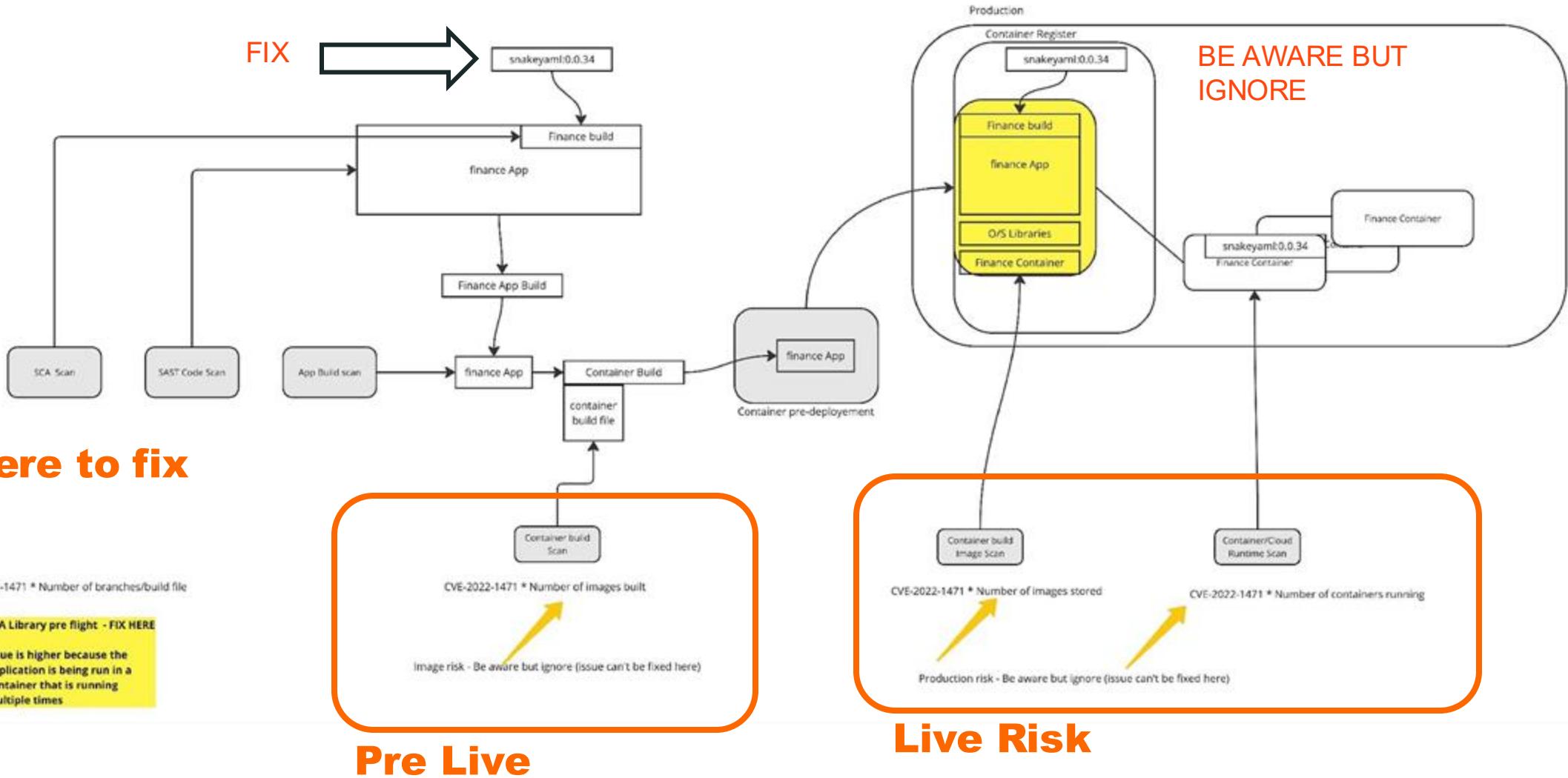
Container Deployment Lineage – What to fix where?



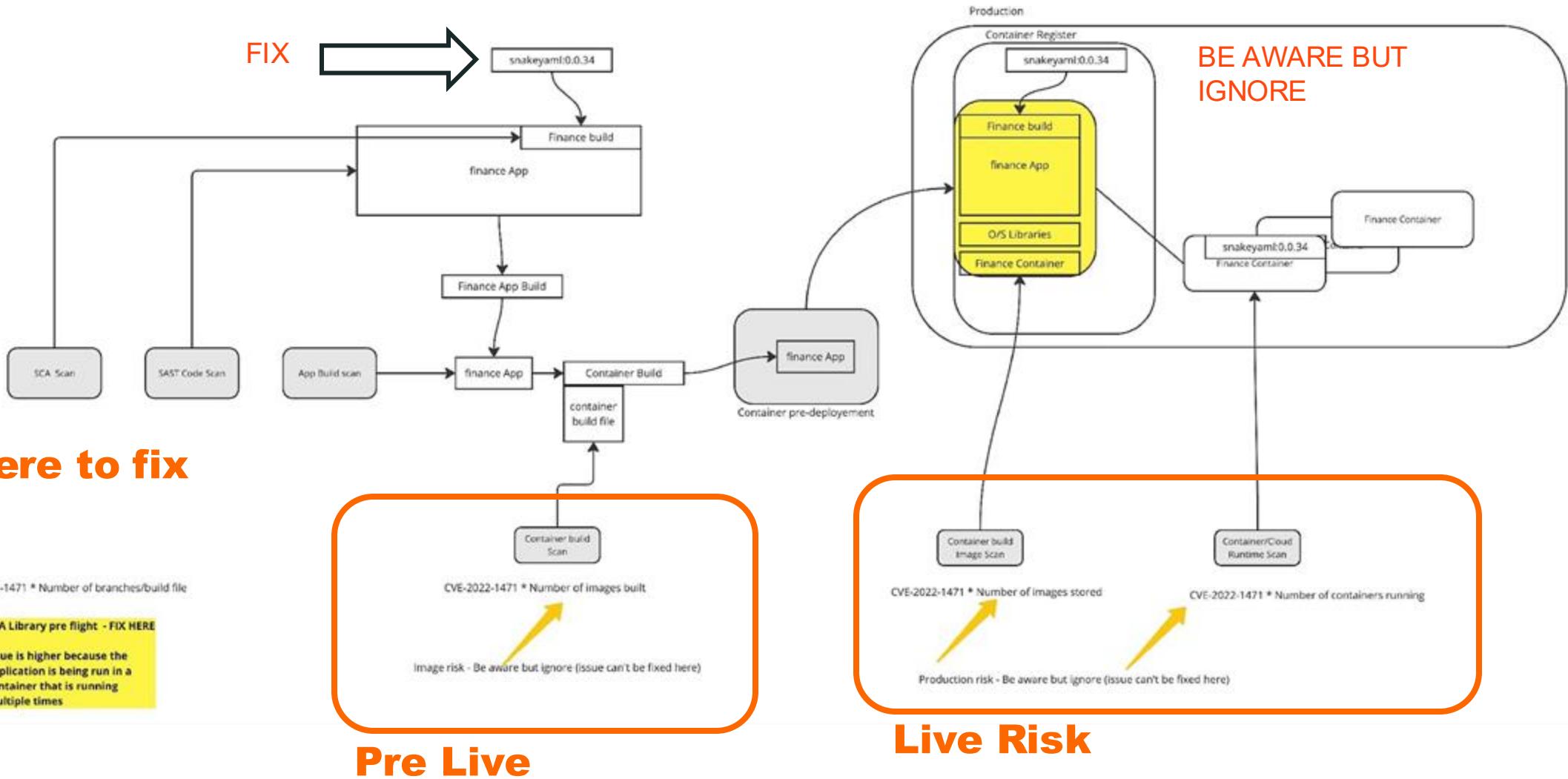
- Why trying to patch all containers when
- ↳ Libraries to patch needs to be fixed by dev team
- ↳ Container images needs to be solved at root
- ↳ Sometimes the problem is with source containers



Real Case Scenario : Deduplicating Contextually Code and Libraries



Real Case Scenario : Deduplicating Contextually Code and Libraries



Where to fix

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

SCA Library pre flight - FIX HERE

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

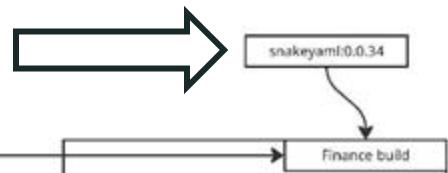
Pre Live

Live Risk

Real Case Scenario : Deduplicating Contextually Code and Libraries



FIX



BE AWARE BUT
IGNORE

Where

A screenshot of the Phoenix Security platform's "Vulnerabilities" page. The sidebar shows navigation links like Dashboard, Risk Explorer, Assets, External Dependencies, Security, and Vulnerabilities (which is selected). The main area shows a search for "CVE-2022-1471" and a filter for "Team: Finance-Fullstack". The results table has columns for Risk, Asset/Location, Name, Type, CVSS / Severity, Discovery Days, Remediation Days, Exploitability (EPSS), Risk Exception, Create Ticket / Ticket Status, and Source. Two rows of results are shown, both related to "snakeyaml:0.0.34" with a severity of 9.8 and a remediation time of N/A. The first row is associated with "Container Finance" and the second with "Finance Backend".

Risk	Asset/Location	Name	Type	CVSS / Severity	Discovery Days	Remediation Days	Exploitability (EPSS)	Risk Exception	Create Ticket / Ticket Status	Source
985	Container Finance	snakeyaml:0.0.34	Arbitrary ...	9.8	9	N/A	2.1%	<input type="checkbox"/>	-	
985	Finance Backend	snakeyaml:0.0.34	Arbitrary ...	8.6	15	N/A	2.1%	<input checked="" type="checkbox"/>	Fix	

Reachability Analysis of a Vulnerability

	1 Code 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 is being loaded in container	Detect if the container is being loaded	Verify if a container's library/node is reachable	Like EPSS identify if a vulnerability is being exploited	Exploit evidence of a vulnerability
WHEN/WERE	Code, Repo, Build	Runtime/ Build	Cluster analysis of container	Cloud/ Operation	Everywhere	Everywhere
BENEFITS	Reduce vulnerabilities in code is being loaded in lib/function not used	Helps identify if the lib/function not used is being loaded in a container, and which container	Image of the container is being used in runtime	Helps identify if the vulnerability can be reached from Remote	Prioritization of vulnerabilities based on exploitation in wild	Prioritization of vulnerabilities based on exploit evidence
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

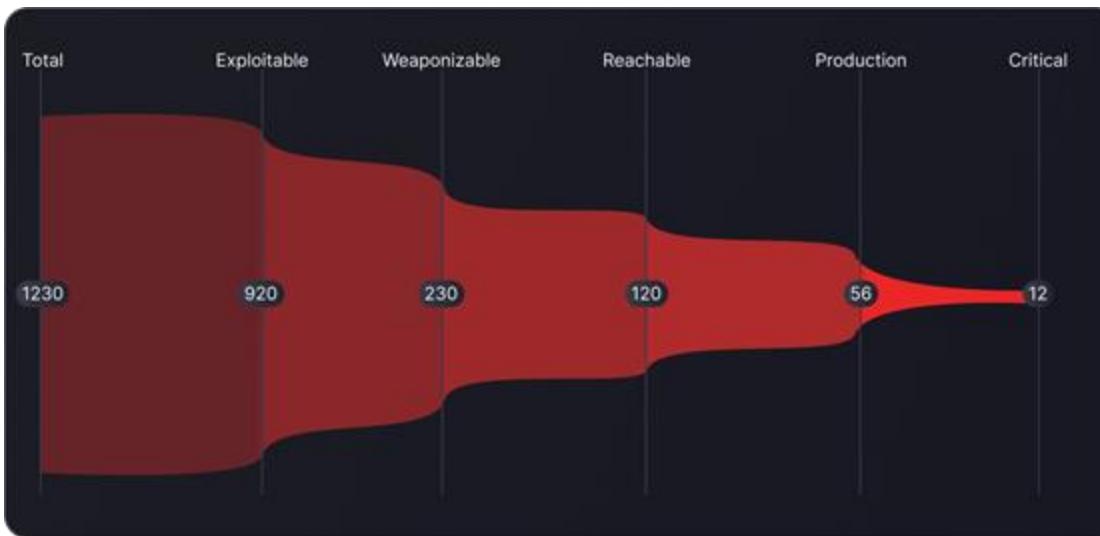
A large, stylized graphic of a flame or leaf, composed of many small orange and red triangles, occupies the left side of the slide. It has a white circular area on its left side.

Part 3 – Risk based Approach

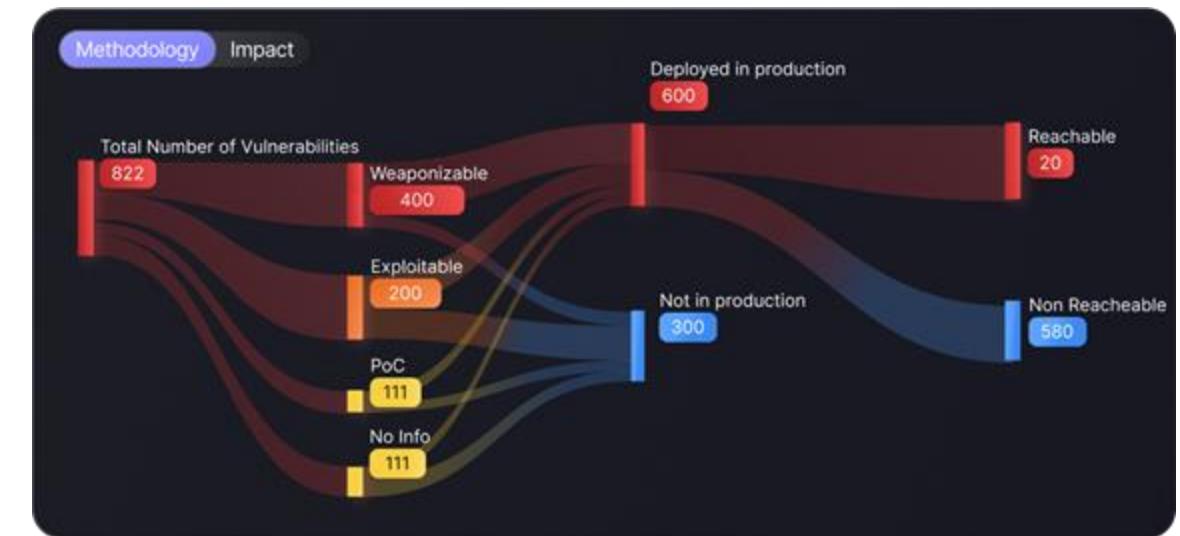
THE GOAL -> ANSWERING QUESTION -> AM I EXPOSED?



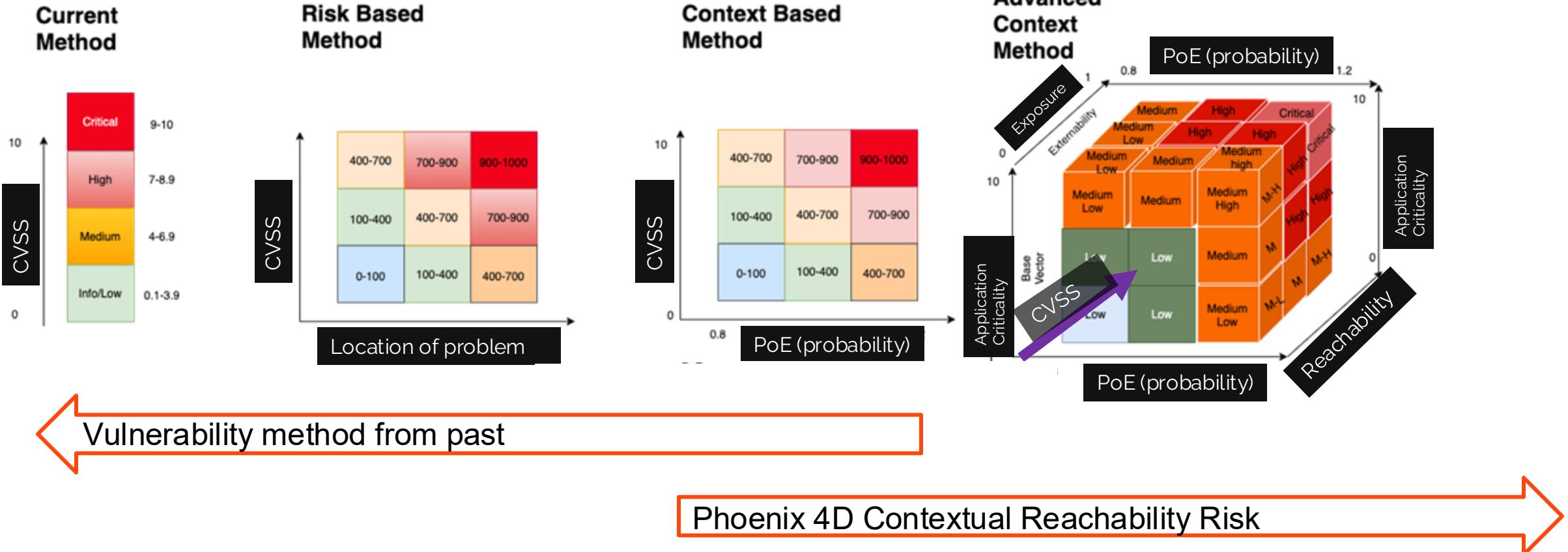
How many Vulnerabilities are actually important



How many exploitable/Weaponizable vulnerability you have



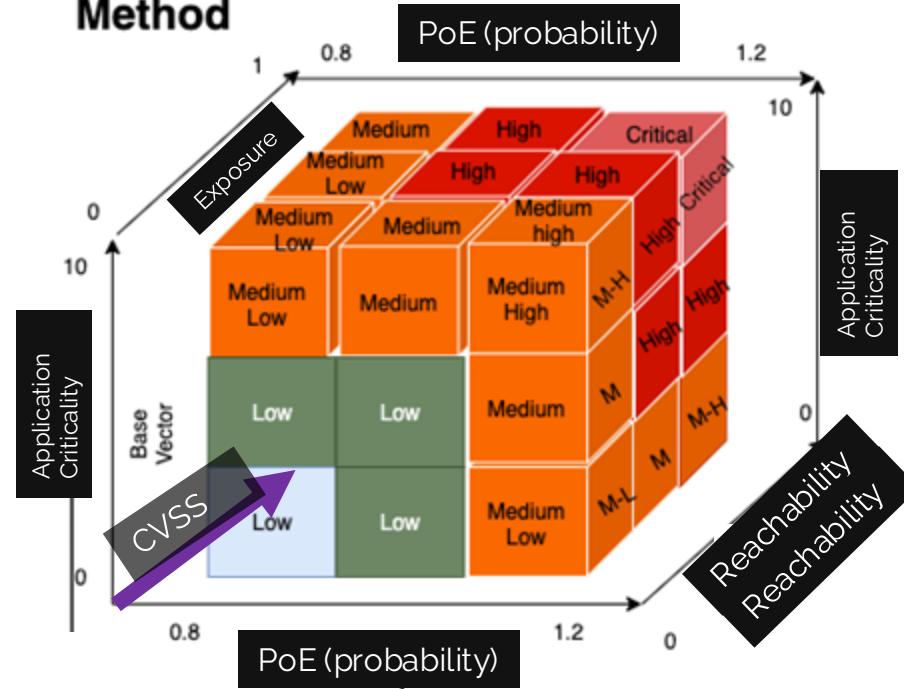
PHOENIX BRINGS OUT THE 4TH DIMENSION OF REACHABILITY



PHOENIX BRINGS OUT THE 4TH DIMENSION OF REACHABILITY

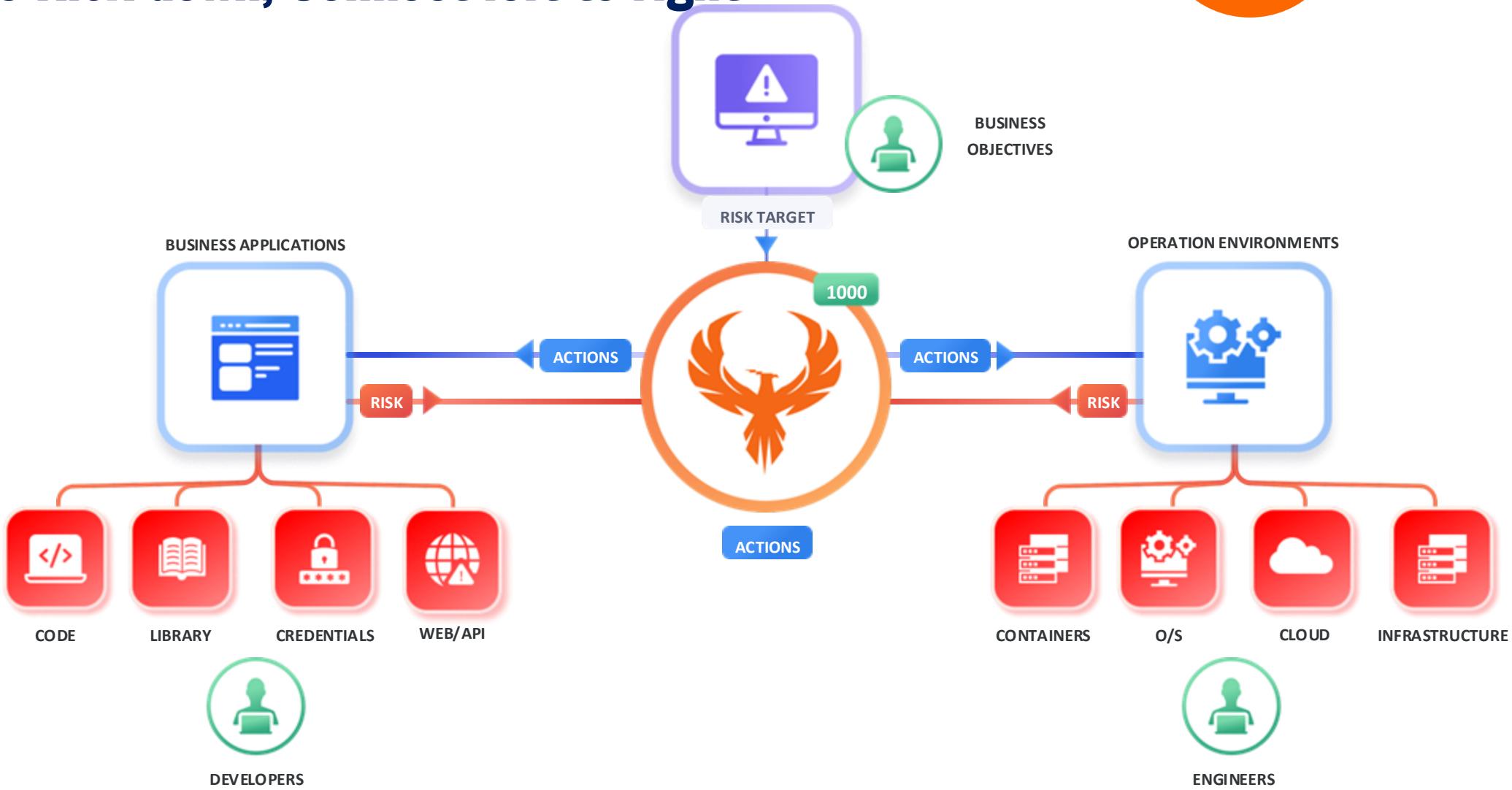


Advanced Context Method



From Number of Vulnerabilities to risk objectives

Drive Risk down, Connect left to right

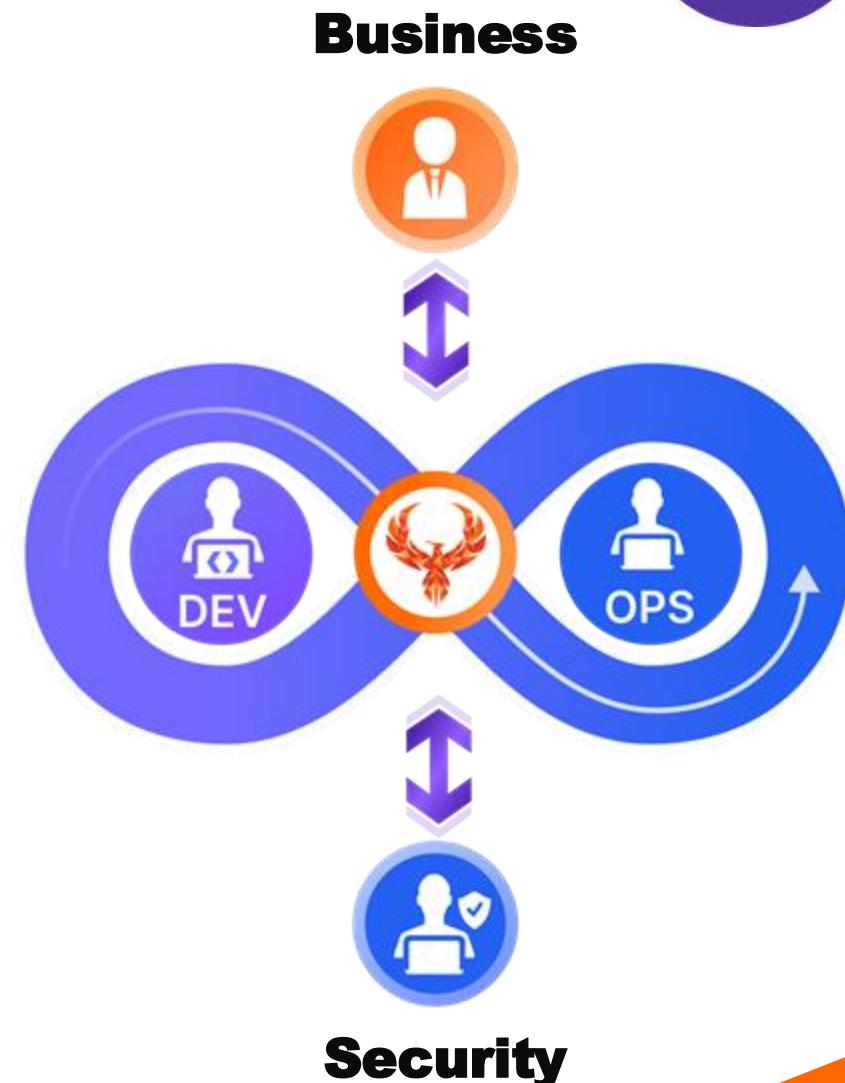


Divide
conundrum



HOW TO SCALE

SET TARGET THAT ARE
ACHIAVABLE INSTEAD OF
SLA NEVER REACHED



RISK COMMON LANGUAGE



Code Runtime Attack Path

Reachability analysis

FIX AVAILABLE

HOW MANY USERS

THREAT INTEL

HOW IMPORTANT

EXPLOITABILITY

PROBABILITY

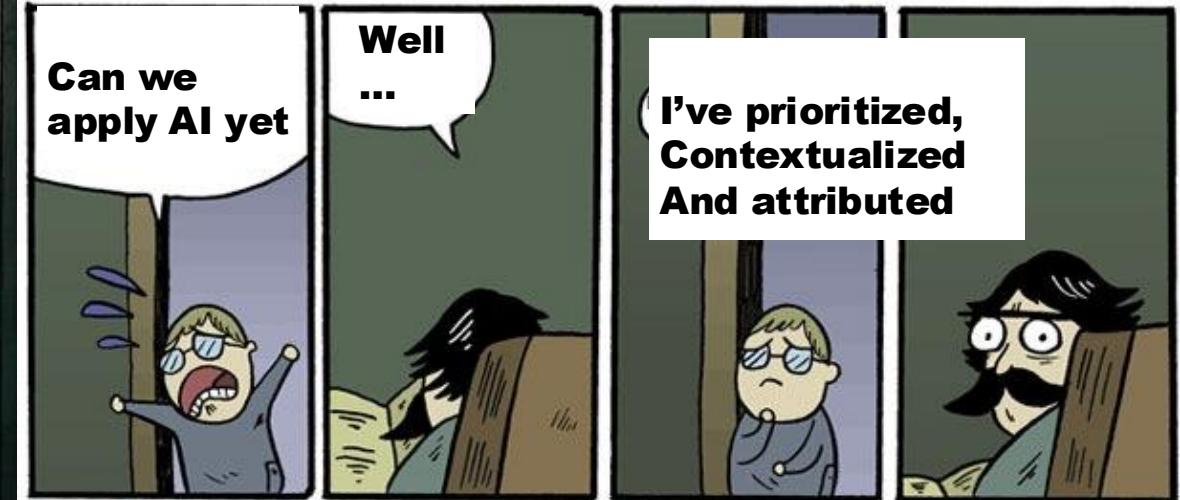
IMPACT

MORE ACCURATE SCORING

SEVERITY

CONTEXT

Can AI/LLM do it for me?





Part 3 – Now can we talk about AI? Agentic Threat Centric

AGENTIC AI - VULNERABILITY ENRICHMENT FROM DETECTION TO RESPONSE

RESEARCH

ANALYZE

RESPOND

Vulnerability details Research

PHOENIX
Research Agent

- Vulnerability Analysis
- Root Cause & Attack Path
- CWE Enhancement
- Threat Actor Mapping



Attack Vectors & Threat Model

PHOENIX
Agentic Analizer

- MITRE Mapping
- Risk Scenarios
- Enhanced Contextual Analysis
- Attack Scenarios
- Threat Modelling



Response Plan

Phoenix
Agentic Remediation Planner

- Grouping suggestions by remedy / Remedy Plan
- Specific Remediation Steps
- Application specific contextual Prioritization



RESEARCHER - AI AGENTIC Appsec

Threat Centric Analysis and description

Vulnerabilities

Vulnerability status: Open, Closed/Ignored, All. Search: log4j

10 results

Risk	Asset / Location	Name	Libraries	Type	CVSS / Severity	Discovery Days	Remediation Days
1000	Security-Phoenix-demo/terrago...	Remote Code Execu...	org.apache.logging.l...	Code	10.0	332	
1000	Security-Phoenix-demo/java-sec...	Remote Code Execu...	org.apache.logging.l...	Code	10.0	331	
1000	Security-Phoenix-demo/java-sec...	log4j-core@2.8.2 is ...	log4j-core: v2.8.2	Code	10.0	69	
989	Security-Phoenix-demo/java-sec...	log4j-api@2.7 is affe...	log4j-api: v2.7	Code	9.8	69	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	-	Code	9.8	516	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j:log4j: v1.2.17	Code	9.8	332	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j:log4j: v1.2.17	Code	9.8	331	
987	Security-Phoenix-demo/WebGoa...	log4j@1.2.17 is affect...	log4j: v1.2.17	Code	9.8	66	

Phoenix remediation copilot

Vector analysis

[WEAKNESS] Improper Control of Generation of Code ('Code Injection')
[PRODUCT] Apache Log4j2
[COMPONENT] log4j-core
[VERSION] 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1)
[ATTACKER] an attacker who can control log messages or log message parameters
[IMPACT] execute arbitrary code loaded from LDAP servers
[VECTOR] when message lookup substitution is enabled
[ROOTCAUSE] JNDI features used in configuration, log messages, and parameters do not protect against attacker-controlled LDAP and other JNDI-related endpoints
[THREAT] Remote Code Execution due to uncontrolled JNDI lookups leading to arbitrary code execution

Agentic Phoenix | 9 m

Exploitability Metrics:

Exploitability:	Weaponized
Probability of exploit:	High
Remediation Level:	Fixable
Base Severity:	High
Privileges Required:	None
Attack complexity:	Low

Threat Intel

```
graph TD; Intelligence --> VulnIntelligence; Intelligence --> Exploitability; Intelligence --> ThreatIntel; Intelligence --> Likelihood; Intelligence --> Exposure; Intelligence --> Popularity; Intelligence --> MITREAttack; Intelligence --> ThreatActor; Intelligence --> ThreatType; VulnIntelligence --> Exploit; Exploitability --> Exploit; ThreatIntel --> Exploit; Likelihood --> Exploit; Exposure --> Exploit; Popularity --> Exploit; MITREAttack --> ThreatActor; ThreatActor --> ThreatType; ThreatType --> APT37; ThreatType --> APT38; ThreatType --> RemoteCode; ThreatType --> BufferOverflow
```



Deep Link Research & Threat Analysis

log4j:log4j version 1.2.13 is vulnerable (CVE-2019-17571) See affected area

987 3.9k |

General Remediation Risk & Threat Actors

Attack graph

```
graph LR; CVE201917571 --> CWE284; CVE201917571 --> CWE306; CVE201917571 --> CWE287; CWE284 --> CAPEC536; CWE284 --> CAPEC546; CWE306 --> CAPEC12; CWE306 --> CAPEC19; CWE287 --> CAPEC115; CAPEC536 --> T1543; CAPEC546 --> T1543; CAPEC12 --> T1543; CAPEC19 --> T1543; CAPEC115 --> T1543; T1543 --> Carbank; T1543 --> APT37; T1543 --> APT38
```

Exploitability metrics:

Exploitability:	Weaponized
Probability of exploit:	High
Remediation Level:	Fixable
Base Severity:	High
Privileges Required:	None
Attack complexity:	Low

Threat intel:

Alert	Ransomware
Actor	APT28
Locations	USA, UK
Industry	Finance
EPSS	0.99 (high)
EPSS Variation	=0% in the last 30 days
Weaponized Exploit	Verified Exploit
IP	169.202.13.25, 110.11.20.32
Total Number of References	154
Threat Actors	154
Exploit Sources	32
Alerting Tags	TIBSI_Inf, TCISA-KEV, AI: 0 Day, AI: Ransomware
Github References	1077

ANALYST - AI AGENTIC Appsec



Threat and Root Cause

Vulnerabilities

Vulnerability status: Open, Closed/Ignored, All. Search: log4j

10 results

Risk	Asset / Location	Name	Libraries	Type	CVSS / Severity	Discovery Days	Remediation Days
1000	Security-Phoenix-demo/terrago...	Remote Code Execu...	org.apache.logging.l...	Remote Code Execu...	10.0	332	
1000	Security-Phoenix-demo/java-sec...	Remote Code Execu...	org.apache.logging.l...	Remote Code Execu...	10.0	331	
1000	Security-Phoenix-demo/java-sec...	log4j-core@2.8.2 is...	log4j-core: v2.8.2	Remote Code Execu...	10.0	69	
989	Security-Phoenix-demo/java-sec...	log4j-api@2.7 is affe...	log4j-api: v2.7	Remote Code Execu...	9.8	69	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	-	Remote Code Execu...	9.8	516	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j;log4j: v1.2.17	Remote Code Execu...	9.8	332	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j;log4j: v1.2.17	Remote Code Execu...	9.8	331	
987	Security-Phoenix-demo/WebGoa...	log4j@1.2.17 is affect...	log4j: v1.2.17	Remote Code Execu...	9.8	66	
943	Security-Phoenix-demo/java-sec...	Incomplete fix for Ap...	-	Remote Code Execu...	9.0	516	



Deep analysis / Threat Modelling MITRE/Attack Simulation



Phoenix remediation copilot

Start deep analysis on CVE-2021-44228.

You | 10 m

Based on the provided information, here's a deep analysis of CVE-2021-44228:

Root cause of vulnerability:
The vulnerability stems from a flaw in the communication design of Yokogawa's CAMS for HIS (Consolidation Alarm Management Software for Human Interface Station). This is categorized as a violation of secure design principles. Specifically, the system allows for the potential misuse of stored credentials.

Weaknesses/vulnerabilities present:

- Violation of Secure Design Principles (CWE-657): the core weakness lies in how CAMS for HIS handles credentials. When a system is compromised, the stored credentials can be used to access and manipulate data on other CAMS for HIS systems.
- Insecure Credential Handling: the vulnerability arises from the fact that credentials stored on one system can be used to authenticate against other systems, leading to unauthorized access.

Impact of exploitation:
Successful exploitation can lead to:

- Data Breach/Leakage: an attacker can access data managed by another CAMS for HIS, leading to potential exposure of sensitive information.
- Data Tampering / Modification: attackers may alter the data managed by other CAMS for HIS instances.
- Resource Exhaustion: an attacker can create unnecessary files on another CAMS for HIS, potentially leading to resource exhaustion and service disruption.
- Denial of Service: ultimately, the attack can disable CAMS for HIS functionality on affected machines due to resource exhaustion.

Attack vectors:

- Compromised System: the primary attack vector involves compromising a computer that has CAMS for HIS software installed.
- Adjacent Network Access: the attacker needs to be on the adjacent network.

Required attacker capabilities/position:

- Initial Access: the attacker needs to gain access to a system running the vulnerable CAMS for HIS software.
- Knowledge of Credentials: once a system is compromised, the attacker needs to be able to access and utilize stored account and password information to access other systems.
- Adjacent Network: the attacker must be on the same network or a network segment that is

Enter your message

Simulate attack scenario.

Agentic Phoenix | 10 m

What would you like to do next?

Get a risk assessment

Get a mitigation plan based on my context

Provide remediation plan

We are GlobalFinance Ltd., a UK-based financial services provider catering to international clients with a focus on secure and seamless transactions. We are a regulated financial institution offering banking, investment, and payment processing services. Our headquarters is in London, UK, with regional offices in Europe, North America, and Asia. We operate a Windows-based enterprise environment running critical financial applications. Our infrastructure includes Active Directory for identity management, single sign-on (SSO) for seamless employee access, and various security tools such as WAF, endpoint detection, and SIEM solutions.

REMEDIATION - AI AGENTIC Appsec

Remediation Plan

Vulnerabilities

Filtrage: Vulnerabilities, Findings, Group By Location

Vulnerability status: Open, Closed/Ignored, All

Search: log4j

10 results

Risk	Asset / Location	Name	Libraries	Type	CVSS / Severity	Discovery Days	Remote
1000	Security-Phoenix-demo/terraogo...	Remote Code Execu...	org.apache.logging.l...	Dependency	10.0	332	
1000	Security-Phoenix-demo/java-sec...	Remote Code Execu...	org.apache.logging.l...	Dependency	10.0	331	
1000	Security-Phoenix-demo/java-sec...	log4j-core@2.8.2 is...	log4j-core: v2.8.2	Dependency	10.0	69	
989	Security-Phoenix-demo/java-sec...	log4j-api@2.7 is affe...	log4j-api: v2.7	Dependency	9.8	69	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	-	Dependency	9.8	518	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j:log4j	Dependency	9.8	332	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j:log4j; v1.2.17	Dependency	9.8	331	
987	Security-Phoenix-demo/WebGoa...	Deserialization of Un...	log4j:log4j	Dependency	9.8	518	
987	Security-Phoenix-demo/WebGoa...	log4j@1.2.17 is affect...	log4j: v1.2.17	Dependency	9.8	66	
943	Security-Phoenix-demo/java-sec...	Incomplete fix for Ap...	org.apache.logging.log4j:log4j-	Dependency	9.0	518	



Remediation and WAF Rules & Compensating Actions



Phoenix remediation copilot

See affected area Chat Remediation Summary

A sensible remediation for CVE-2021-44228 in the Security-Phoenix-demo/java-sec-code:pom.xml project, which currently uses log4j-core v2.8.2, is to upgrade the vulnerable dependency to a non-vulnerable release of Log4j. The safest bet is to update to Log4j 2.17.2 (or newer) if possible:

1. Update the pom.xml
In your projects pom.xml, locate the Log4j dependency:

```
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.8.2</version>
</dependency>
```

Replace the version with a secure release (e.g. 2.17.2): [Copy](#)

```
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.17.2</version>
</dependency>
```

2. Check for Transitive Dependencies
Sometimes other libraries drag in older Log4j versions behind your back. Use a dependency analysis tool (mvn dependency:tree) to confirm you haven't any hidden references to out-of-date Log4j components lurking in the shadows. If you spot older versions, enforce overrides in your pom.xml to ensure they're replaced by the updated one.

3. Remove or Disable JNDI Lookups (if immediate upgrading is impossible)
• Delete the JndiLookup class: From your Log4J JAR (zip -q -d log4j-core-*.*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class).
• Set the system property: -Dlog4j2.formatMsgNoLookups=true to disable lookups at runtime.
This is more of a temporary plaster, though—long-term safety relies on using a fixed version.

4. Additional Hardening
• Monitor logs and traffic: Keep an eye on your application logs for unusual patterns, and ensure your WAF is tuned to detect suspicious payloads aiming to exploit older Log4j flaws.
• Network egress filters: Restrict outbound connections to only the domains that your application legitimately needs. This prevents malicious LDAP lookups reaching unscrupulous servers—like building a moat and drawbridge around your code.

5. Verification and Testing
• Clean Build: Once you bump the version, do a full build and check that no sneaky older references remain.
• Penetration Testing: After updating, subject your application to a pen test or vulnerability scan. Better to find potential trouble now than after an attacker does.

Phoenix remediation copilot

See affected area Chat Remediation Summary

Provide me a list of AWS WAF rules.

You | 10 m

Below is a list of AWS WAF rules to mitigate the identified vulnerabilities. These rules are designed to protect the application deployed behind an AWS Load Balancer with WAF and are tailored to address the Log4j (CVE-2021-44228), malicious flat-axis, and libbz2 vulnerabilities.

1. Mitigating Log4j (CVE-2021-44228)
Rule Name: BLOCK_JNDI_EXPLOIT
• Purpose: Detect and block requests attempting to exploit JNDI injection in Log4j.
• Rule Type: Regex pattern matching in AWS WAF.

Example Conditions: [Copy](#)

```
{
  "Name": "BLOCK_JNDI_EXPLOIT",
  "Priority": 1,
  "Action": {
    "Block": {}
  },
  "Statement": {
    "RegexMatchStatement": {
      "FieldToMatch": {
        "Body": {}
      },
      "RegexPatternSetReferenceStatement": {
        "ARN": "arn:aws:wafv2:REGION:ACCOUNT_ID:regional/regexpatternset/RegexForJNDI"
      },
      "TextTransformations": [
        {
          "Priority": 0,
          "Type": "NONE"
        }
      ]
    }
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true
  }
}
```

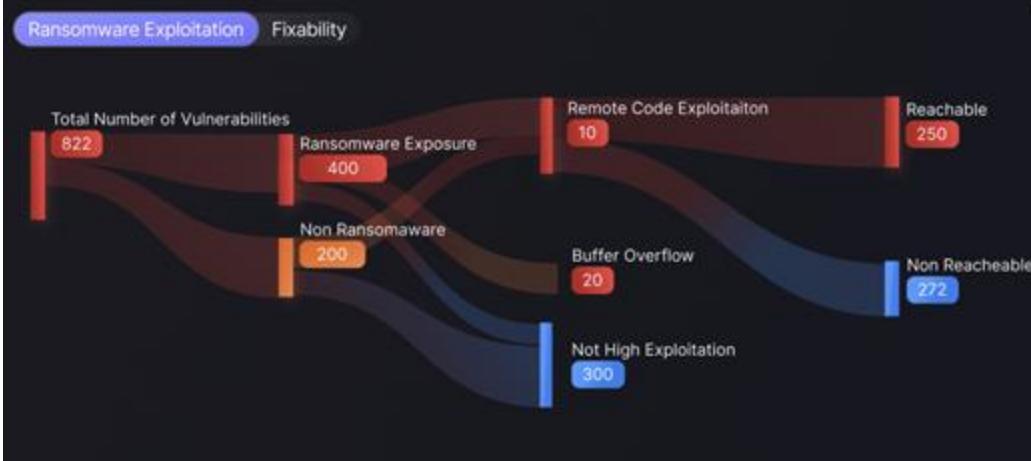
Can AI/LLM do it for me?



Answering Question – What threat AM I EXPOSED



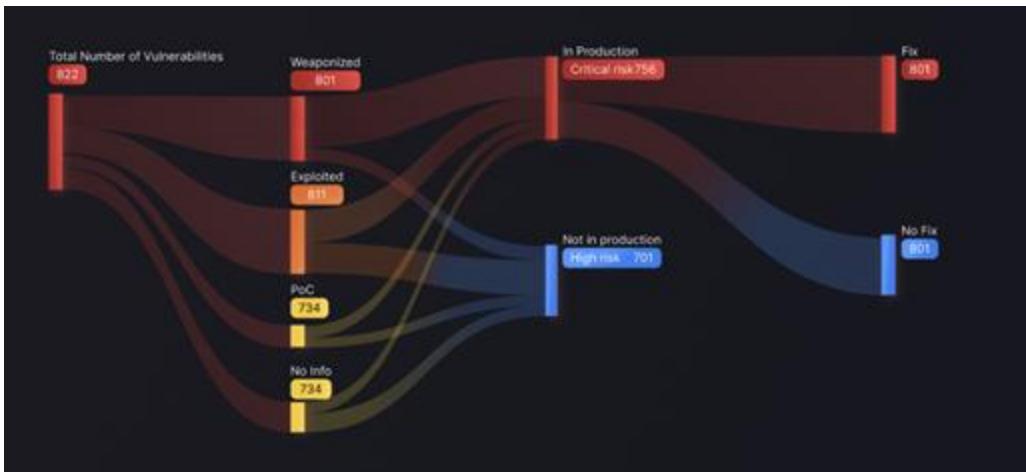
What are your exposure to Ransomware



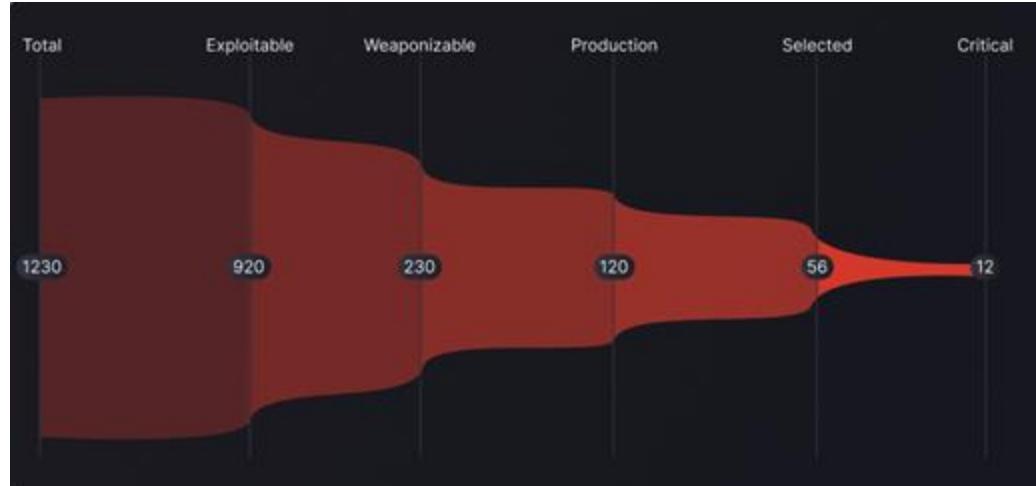
How exposed are you to threat actor



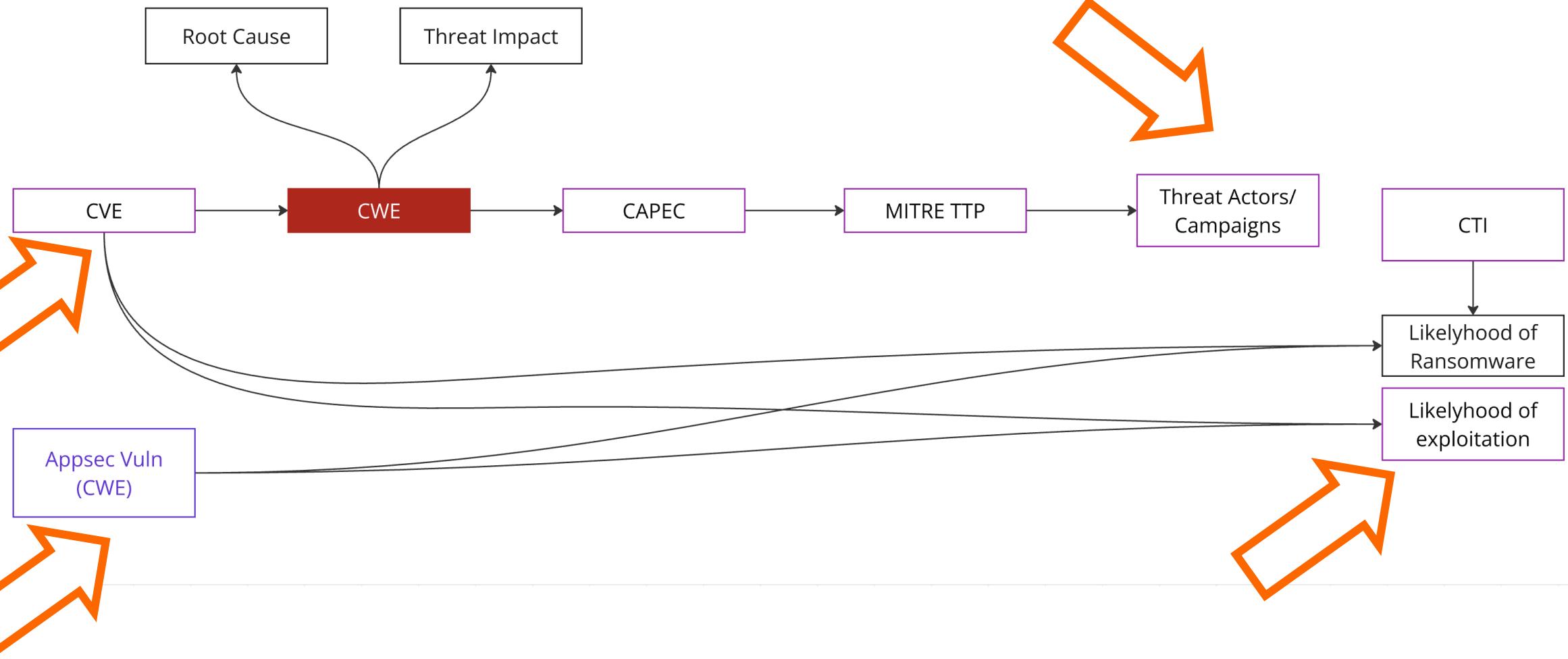
How many exploitable/Weaponizable vulnerability you have



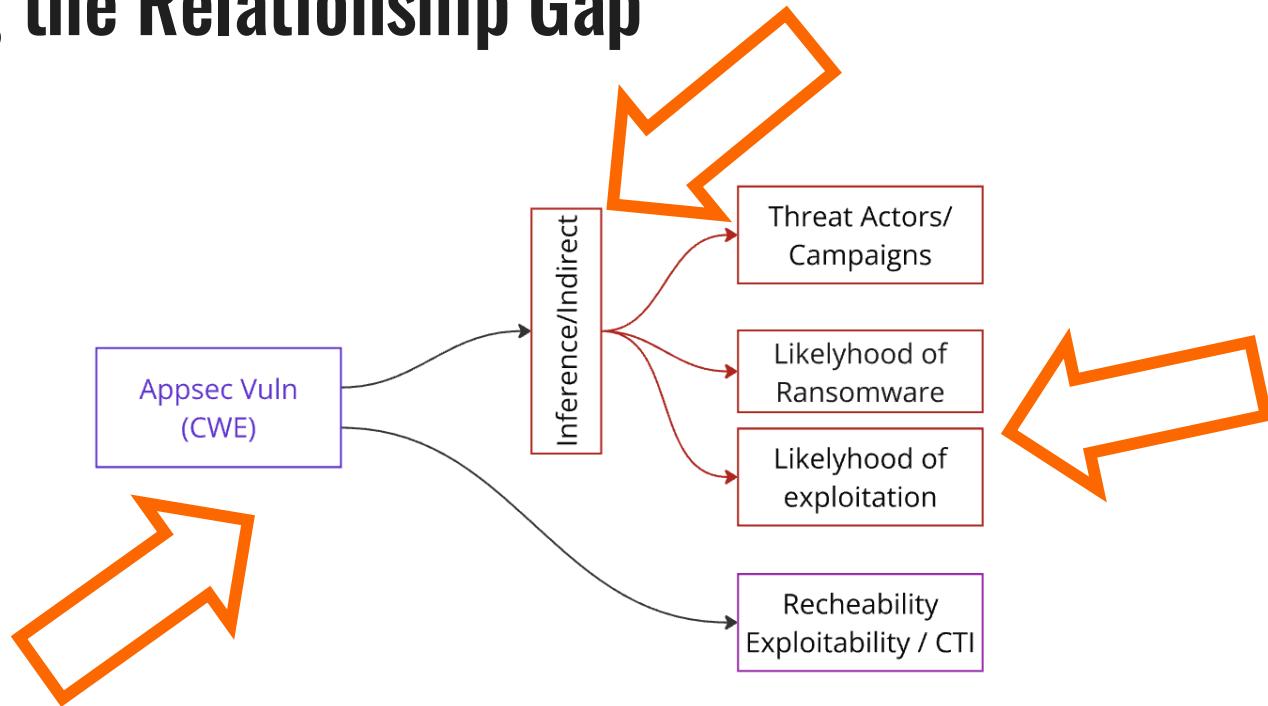
How many Vulnerabilities are actually important



Relationship Gap

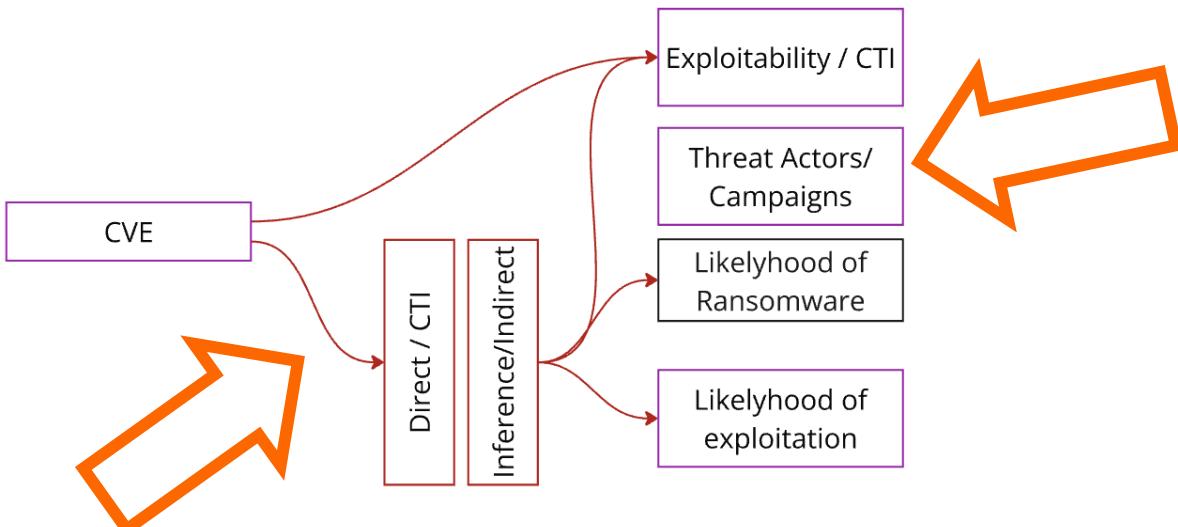


Filling the Relationship Gap



For CWE ? Appsec Vulnerabilities

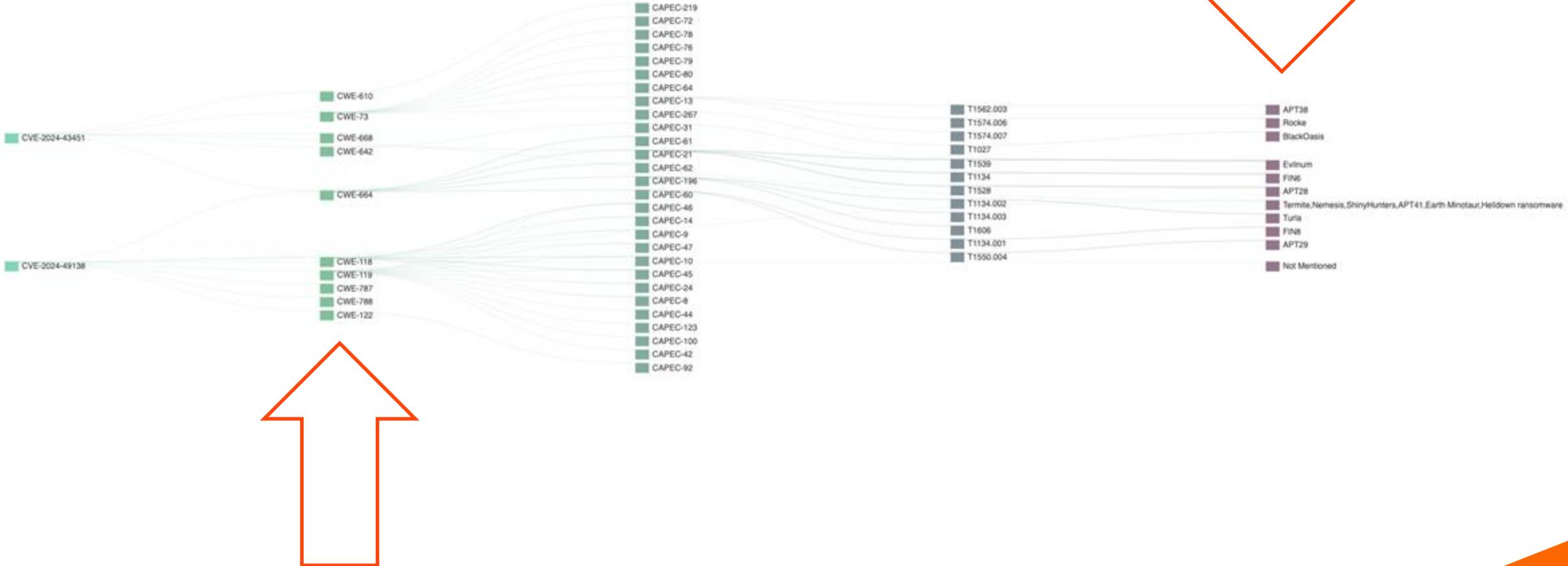
- Knowing the threat actors
- Apply CTI / indirectly
- Knowing possibly of ransomware and exploitation



For CVE based

- Knowing and increasing the coverage CVE – Threat Actor / Technique
- Knowing Which technique is more used
- Use this as probabilistic methods

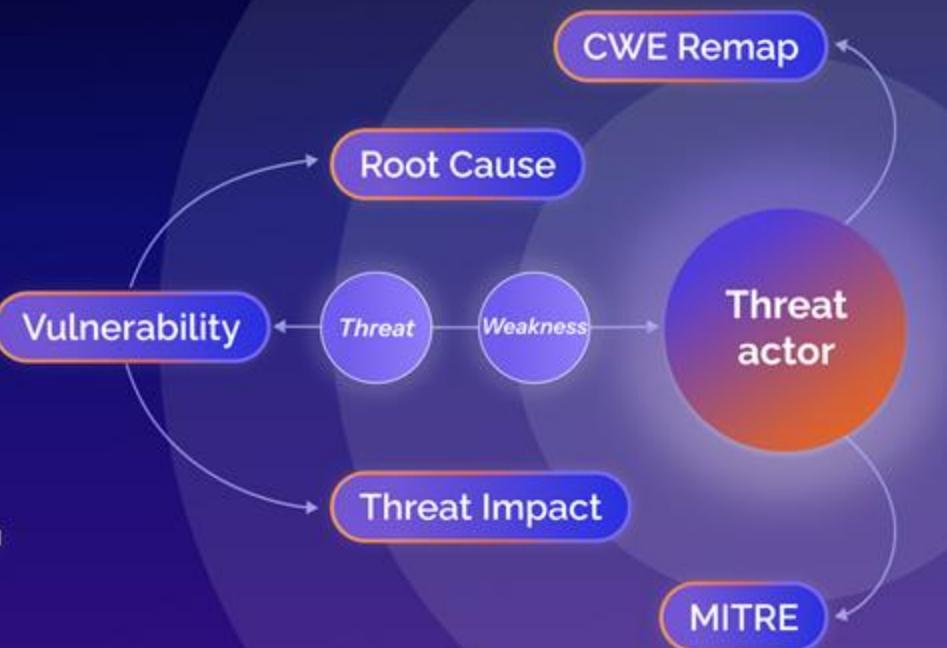
Challenge – Description of vulnerabilities IS NOT root cause analysis



Threat Centric Approach on vulnerabilities



AI Threat centric mapper



AI augmented attack vector for CVE-2018-17924

[VENDOR] Rockwell Automation
[PRODUCT] MicroLogix 1400 Controllers, 1756 ControlLogix EtherNet/IP Communications Modules
[COMPONENT] N/A
[VERSION] Series A, all versions, Series B, v21.003 and earlier, Series C, v21.003 and earlier, 1756-ENBT, all
[WEAKNESS] Missing Authentication
[ATTACKER] Unauthenticated remote attacker
[IMPACT] Modify system settings, loss of communication
[VECTOR] CIP connection request
[ROOTCAUSE] Lack of proper authentication for critical functions
[VULNERABILITY TYPE] Input Validation
[VULNERABILITY IMPACT] Sensitive information disclosure

CVE-2018-17924

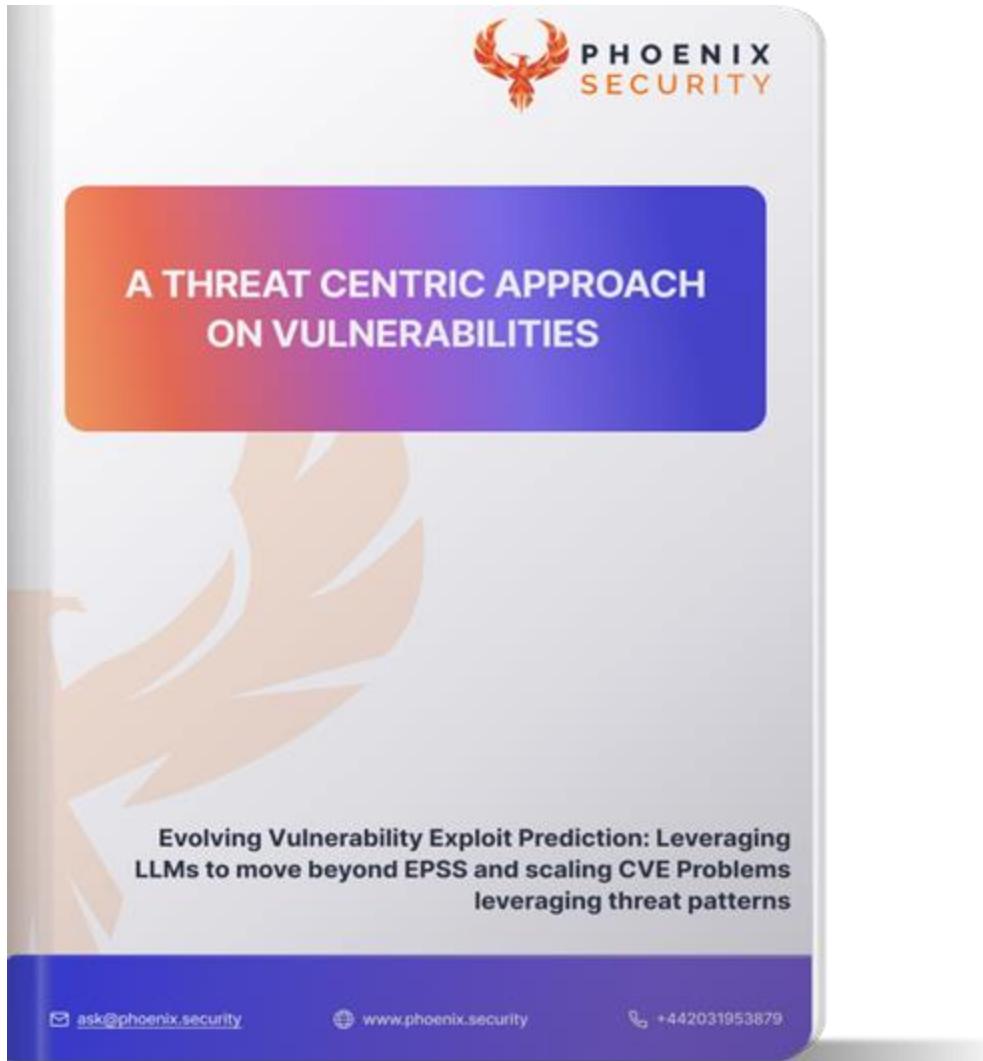
CVE-2024-37079

CAPEC-503
CAPEC-505
CAPEC-563
CAPEC-441
CAPEC-548
CAPEC-549
CAPEC-551
CAPEC-550
CAPEC-564
CAPEC-548
CAPEC-38
CAPEC-19
CAPEC-12
CAPEC-208
CAPEC-216
CAPEC-478
CAPEC-562
CAPEC-564
CAPEC-519
CAPEC-552
CAPEC-194
CAPEC-151
CAPEC-119
CAPEC-22
CAPEC-94
CAPEC-650
CAPEC-577
CAPEC-593
CAPEC-633
CAPEC-115
CAPEC-114
CAPEC-62
CAPEC-61
CAPEC-9
CAPEC-49
CAPEC-24
CAPEC-21
CAPEC-14
CAPEC-10
CAPEC-9
CAPEC-45
CAPEC-60
CAPEC-47
CAPEC-196
CAPEC-100
CAPEC-42
CAPEC-123
CAPEC-44

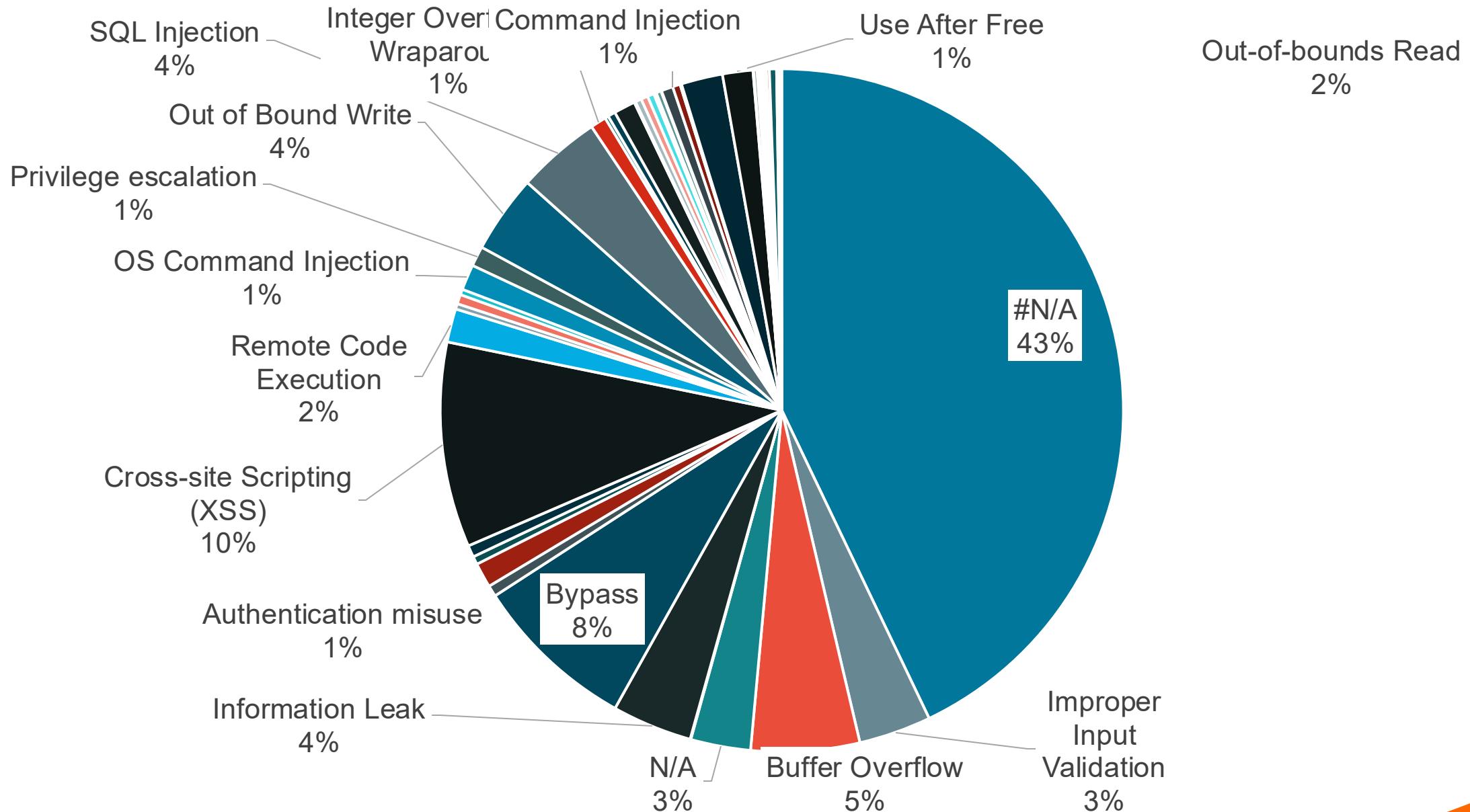
T1543.001
T1543.004
T1547
T1037
T1546.009
T1546.004
T1546.016
T1505.005
T1574.011
T1562.008
T1086
T1543.003
T1562.004
T1562.011
T1562.002
T1546.001
T1014
T1562.006
T1562.007
T1562.008
T1562.009
T1562.010
T1542.003
T1505.003
T1547.006
T1547.004
T1040
T1550.001
T1165
T1548
T1548
T1134
T1528
T1529
T1134.001
T1550.004
T1134.002
T1134.003
T1606

Rocks
Fox Kitten
Gamaredon Group
Carbanak
Gitolot Group
Red Team Team
Wiper Group
Kimsuky
Volatile Cedar
APT28
FIN6
Evilnut
FINB
APT29
Turla

New Whitepaper LLM for a Threat Centric Approach



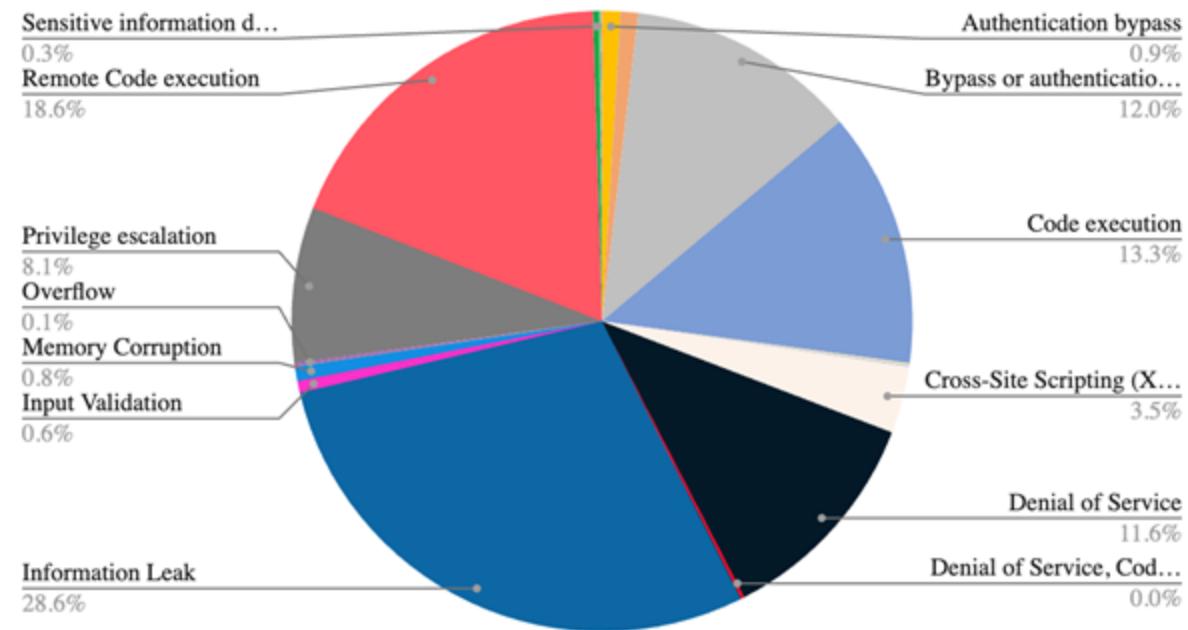
Current CWE Completion – 50% incomplete



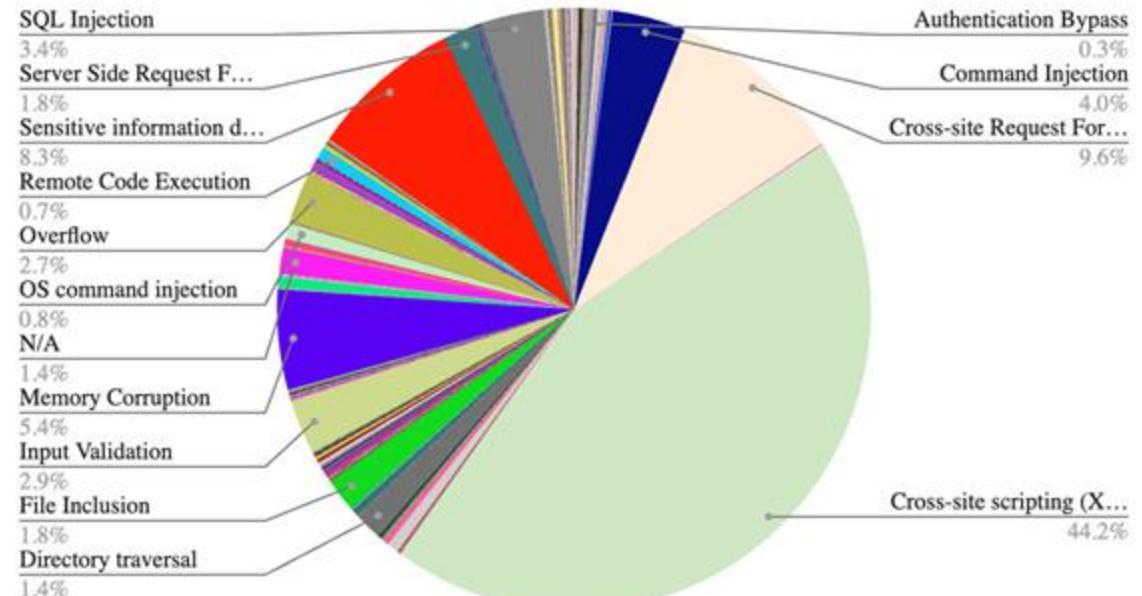
Way Ahead LLM Generation of Simplified Description of vulnerabilities, root cause and technical impact



Threat Root Cause



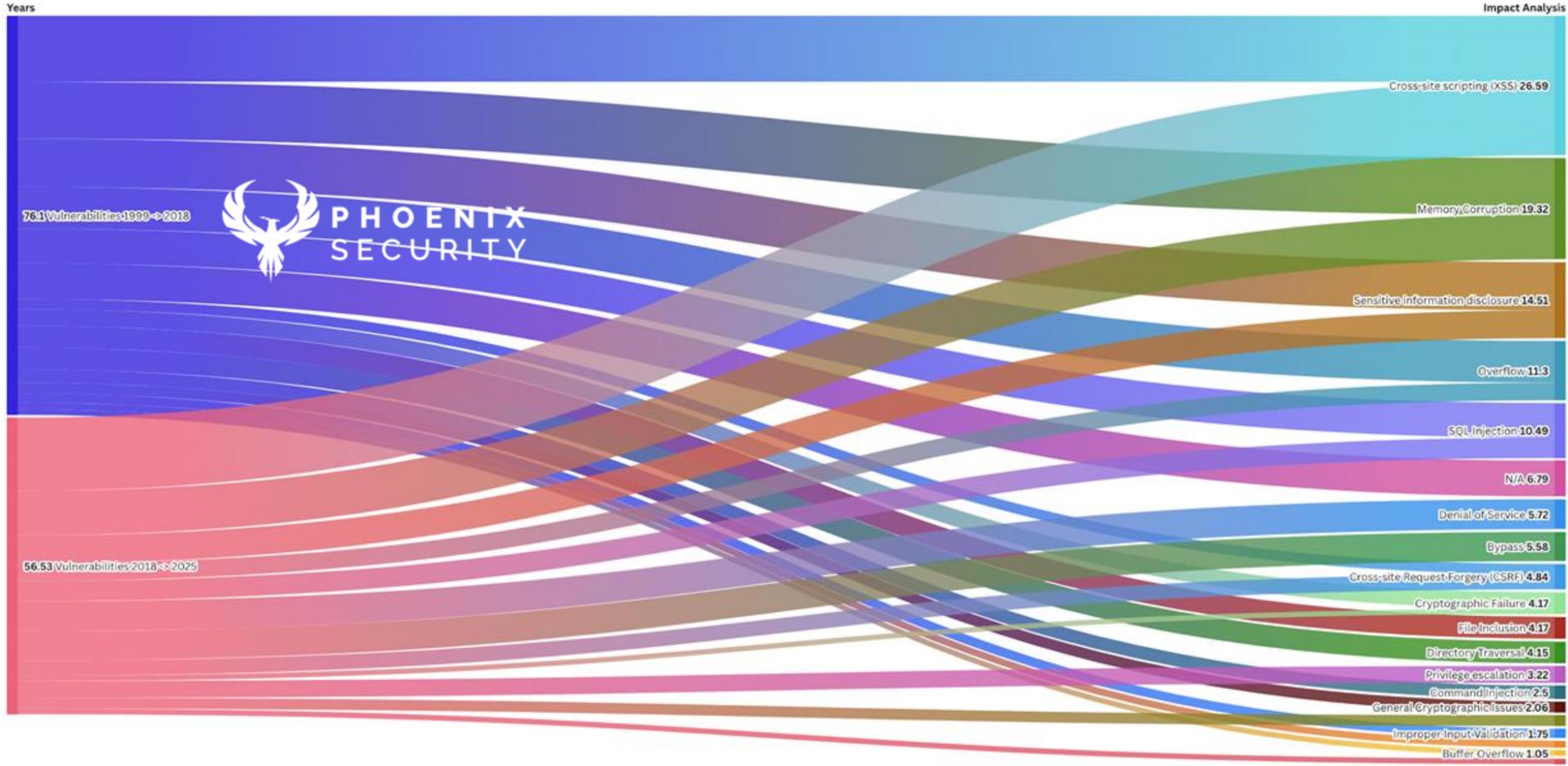
威胁影响 / 弱点



How the Path has changed ... and not -> root cause and...



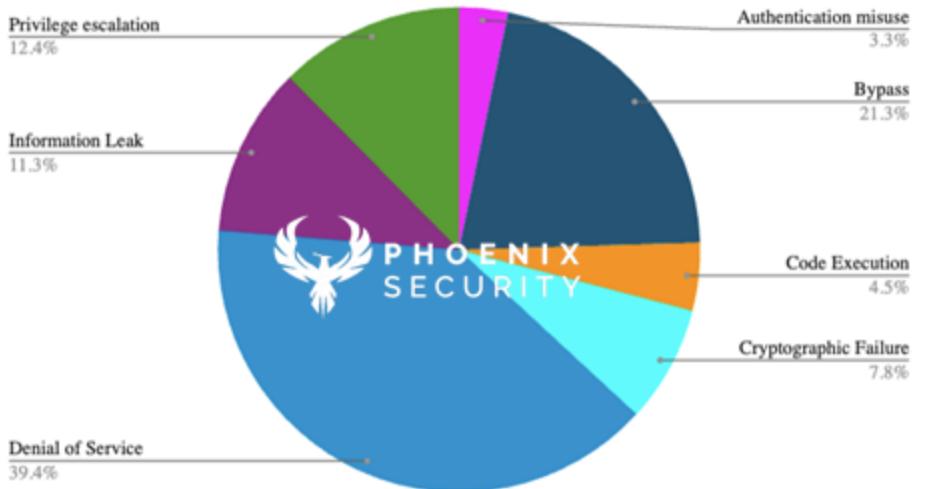
How the Path has changed ... and not -> ...and technical impact



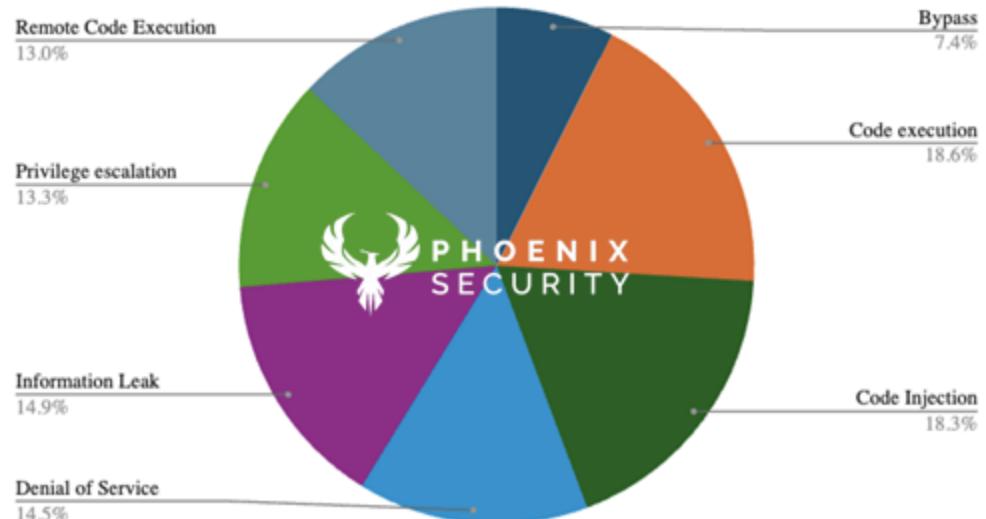
How the threat have evolved

Technical Impact

Root Cause 1999 - 2018

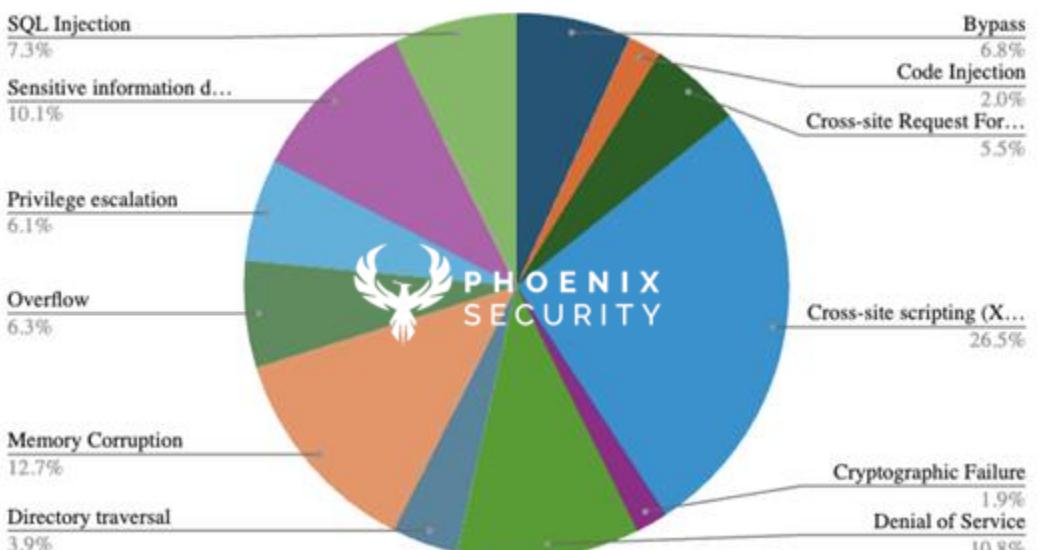
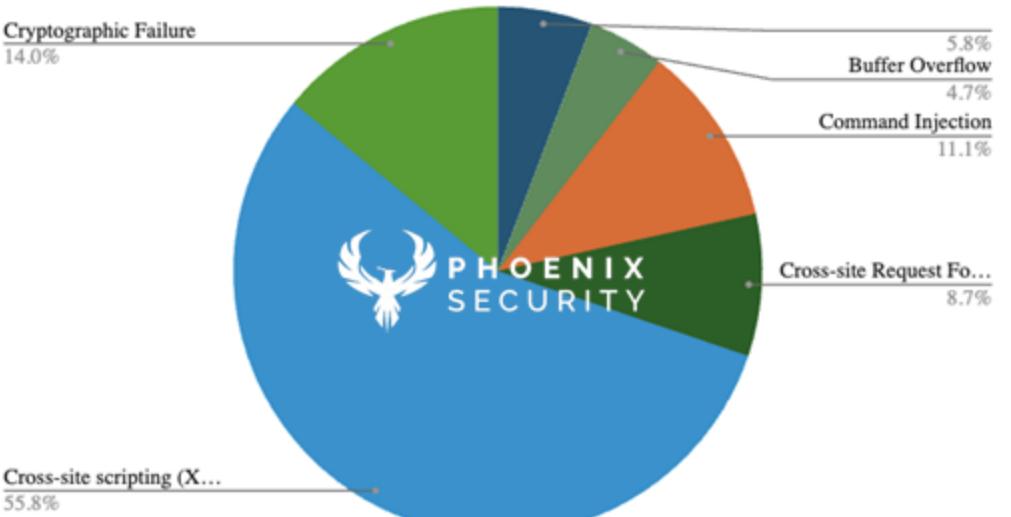


CVE Root Cause 2018 - 2015



Root Cause / Weakness

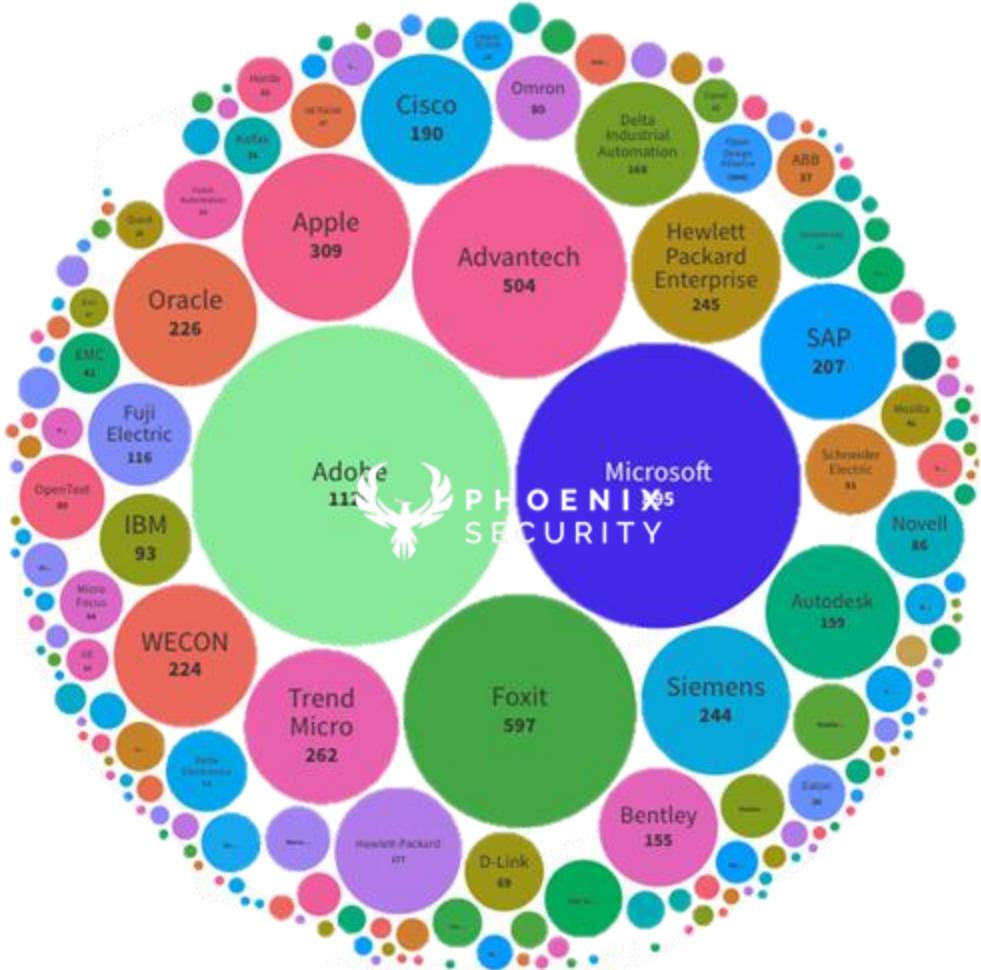
Impact 1999 - 2018



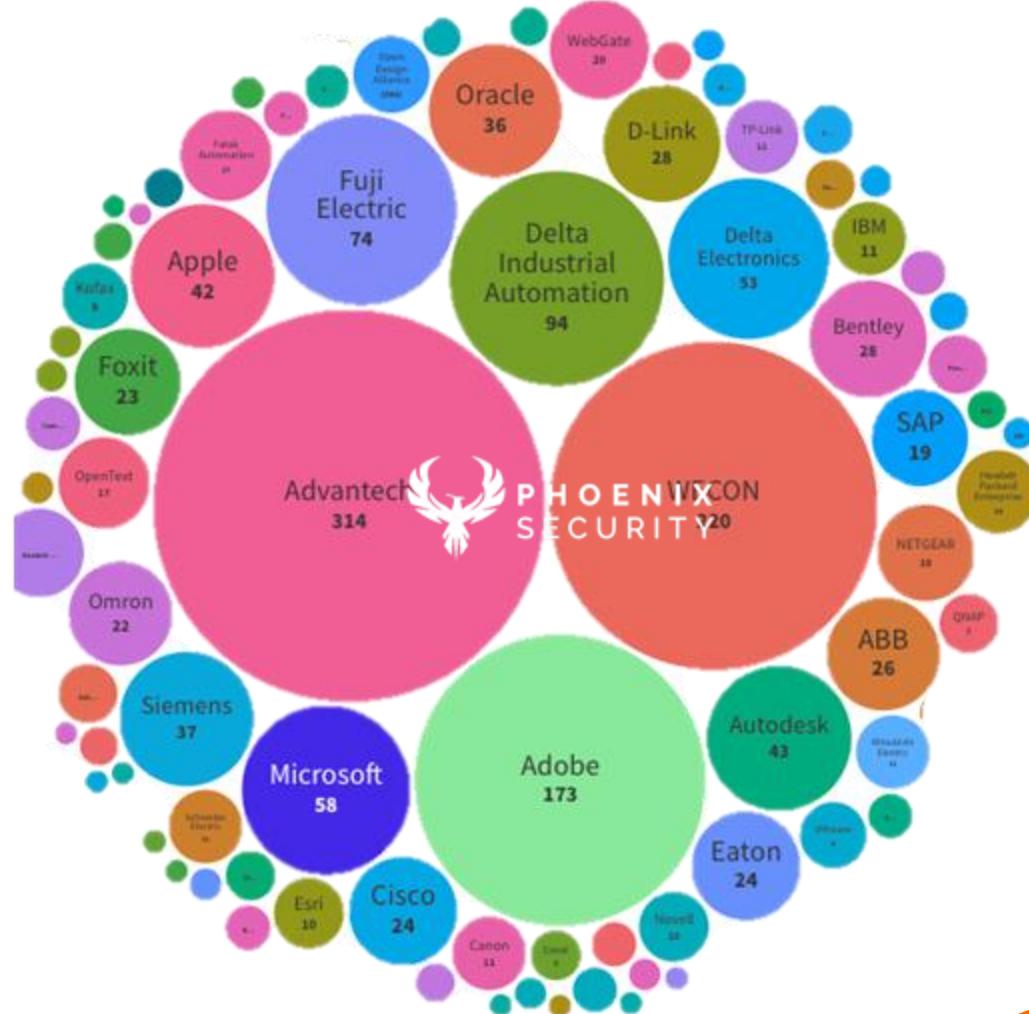
Methodologies of attacks in Zero Days



Root Cause : RCE



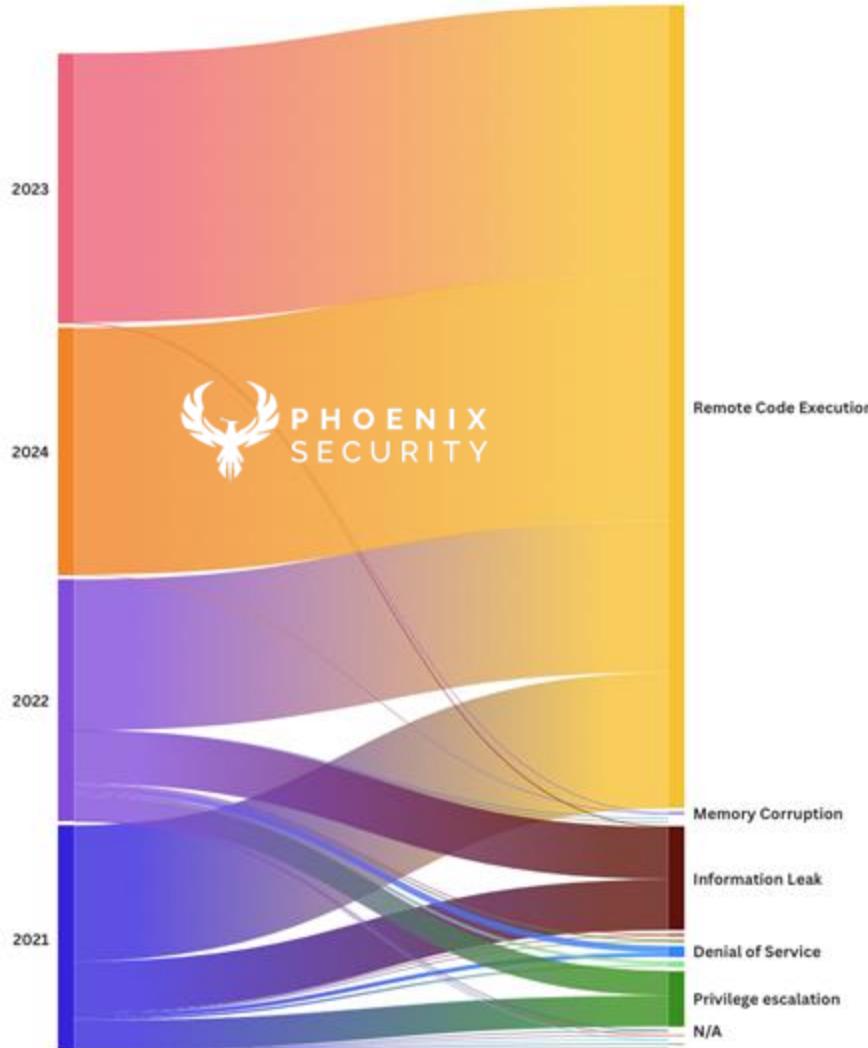
Root Cause : Buffer Overflow



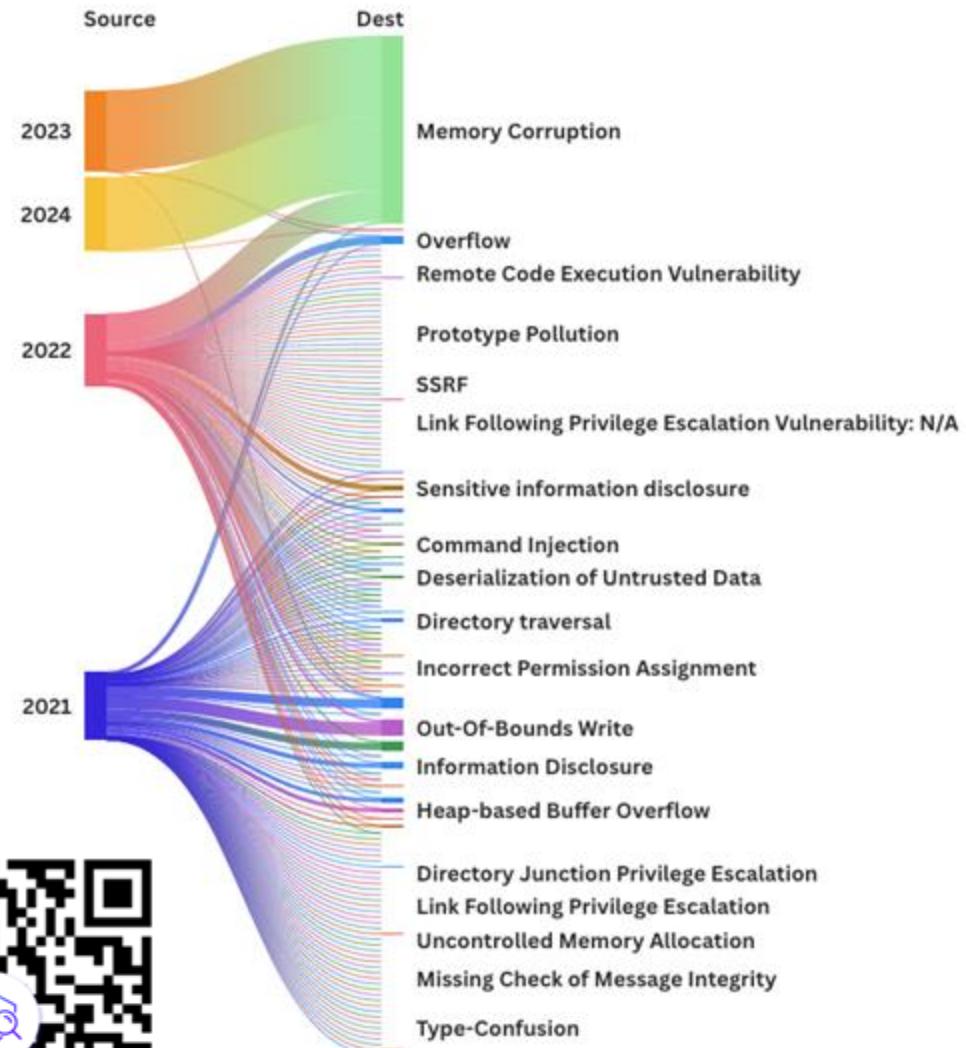
Methodologies of attacks in Zero Days - Weakness



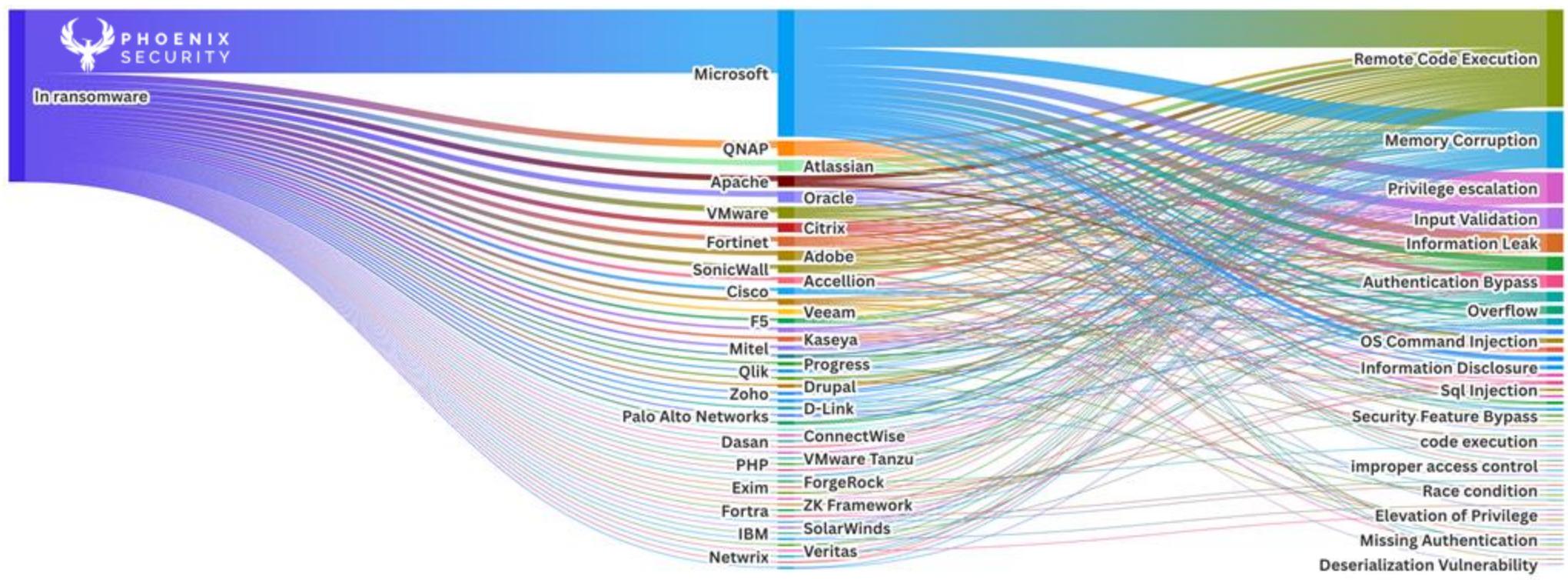
Root Cause : RCE



Technical Impact: Mem Corruption

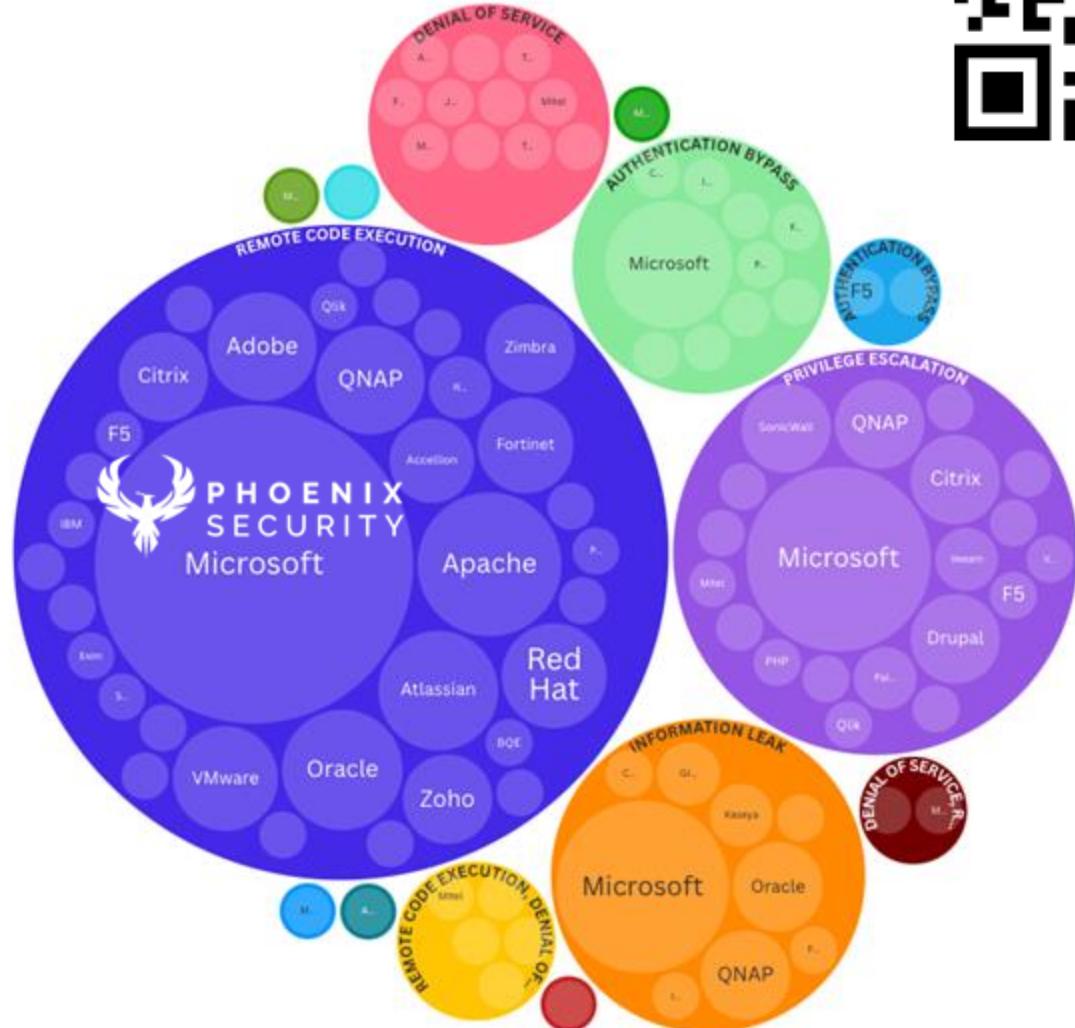


Methodologies of attacks in Ransomware

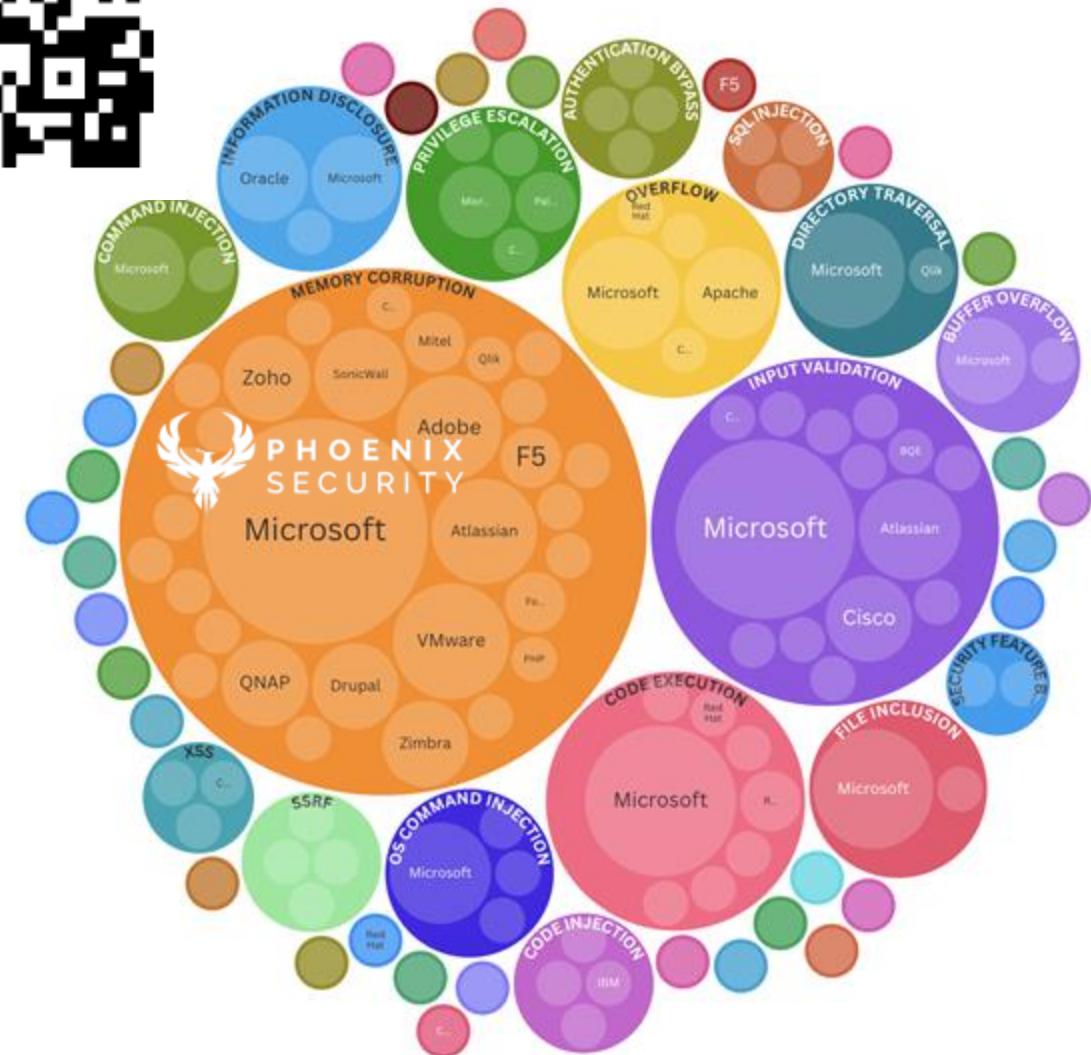


Methodologies of attacks in Ransomware

Technical Impact



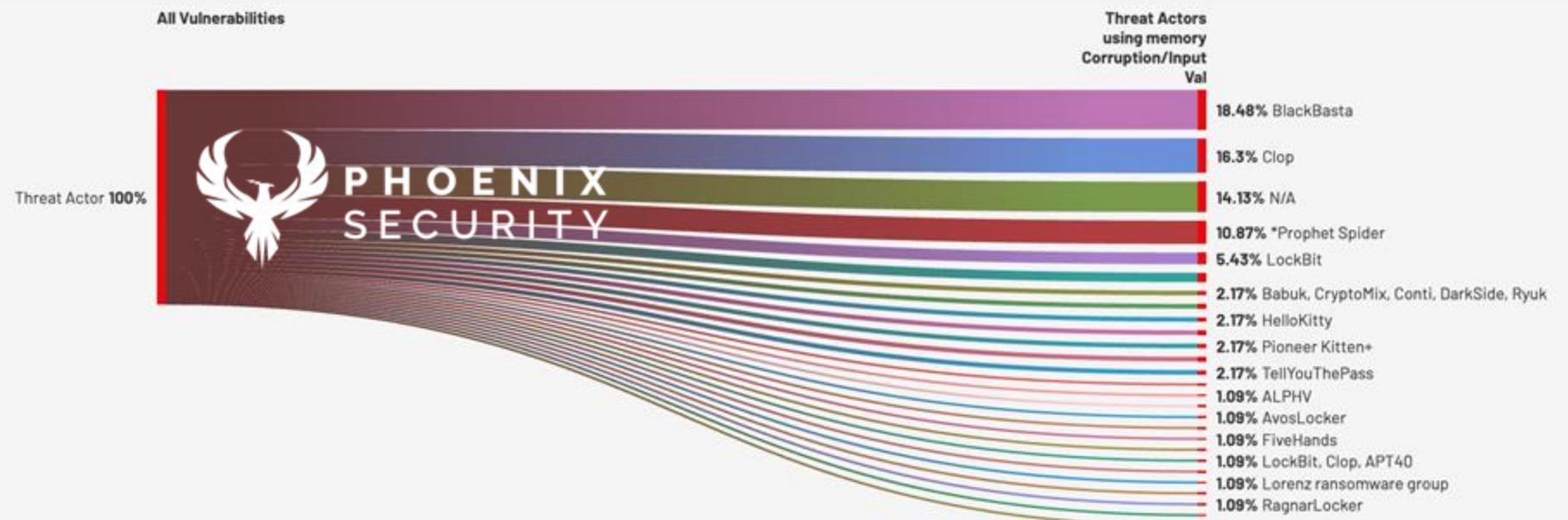
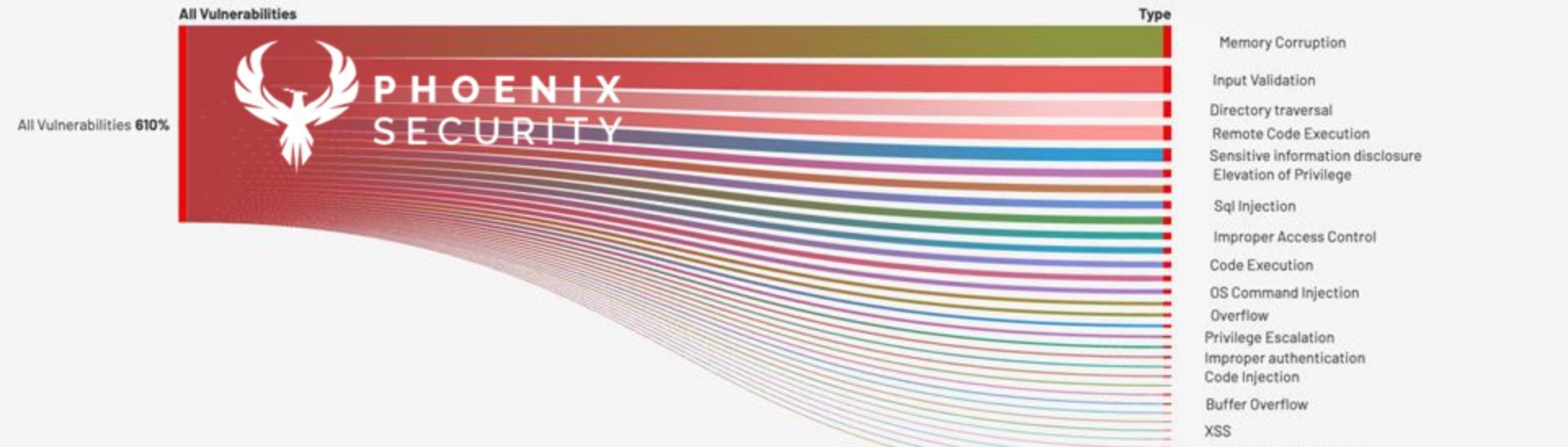
Root Cause



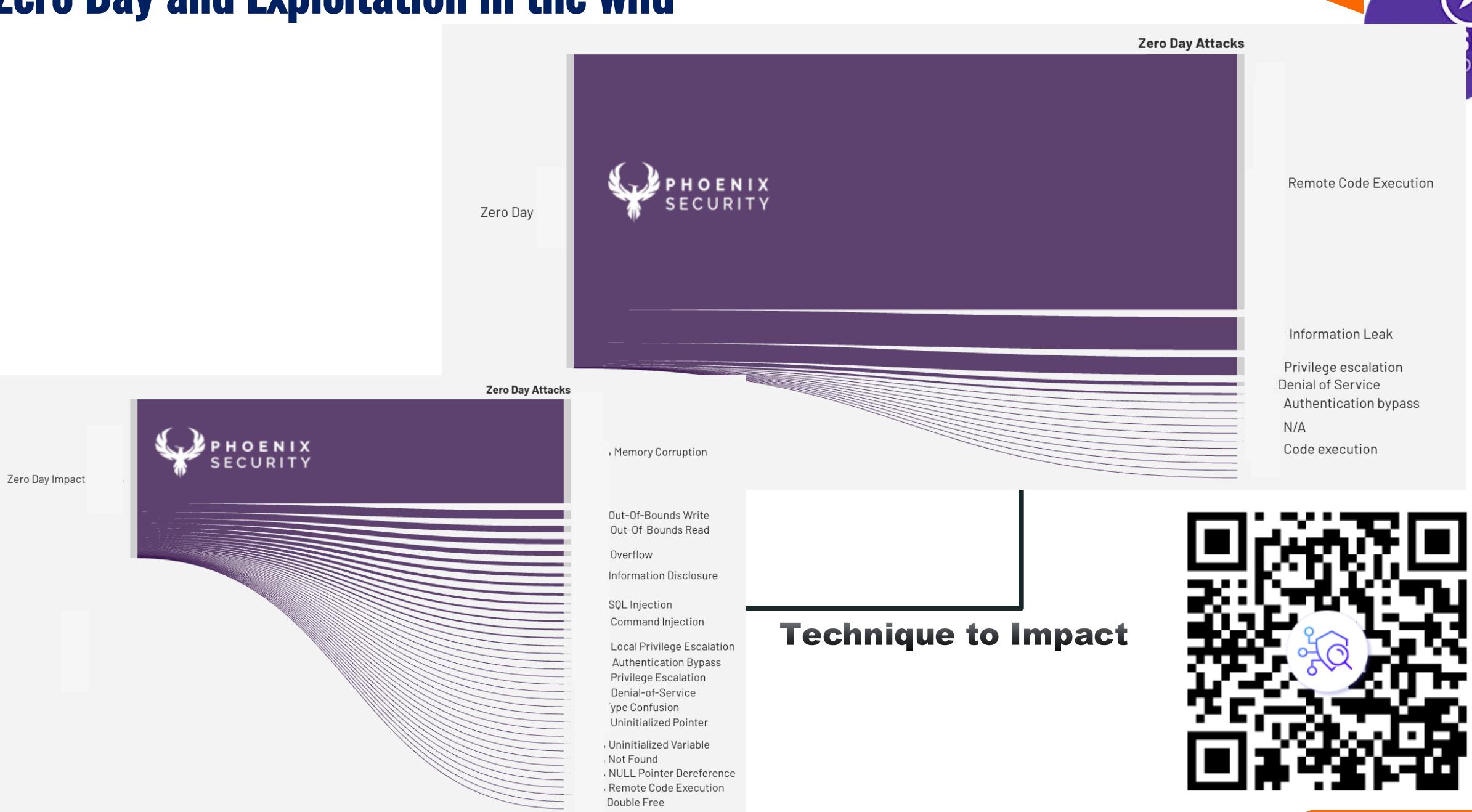
Methodologies of attacks in Ransomware



Impact of attacks in Ransomware



Zero Day and Exploitation in the wild



Technique to Impact



Divergence in Bug Bounty Dataset

Probability based on threats

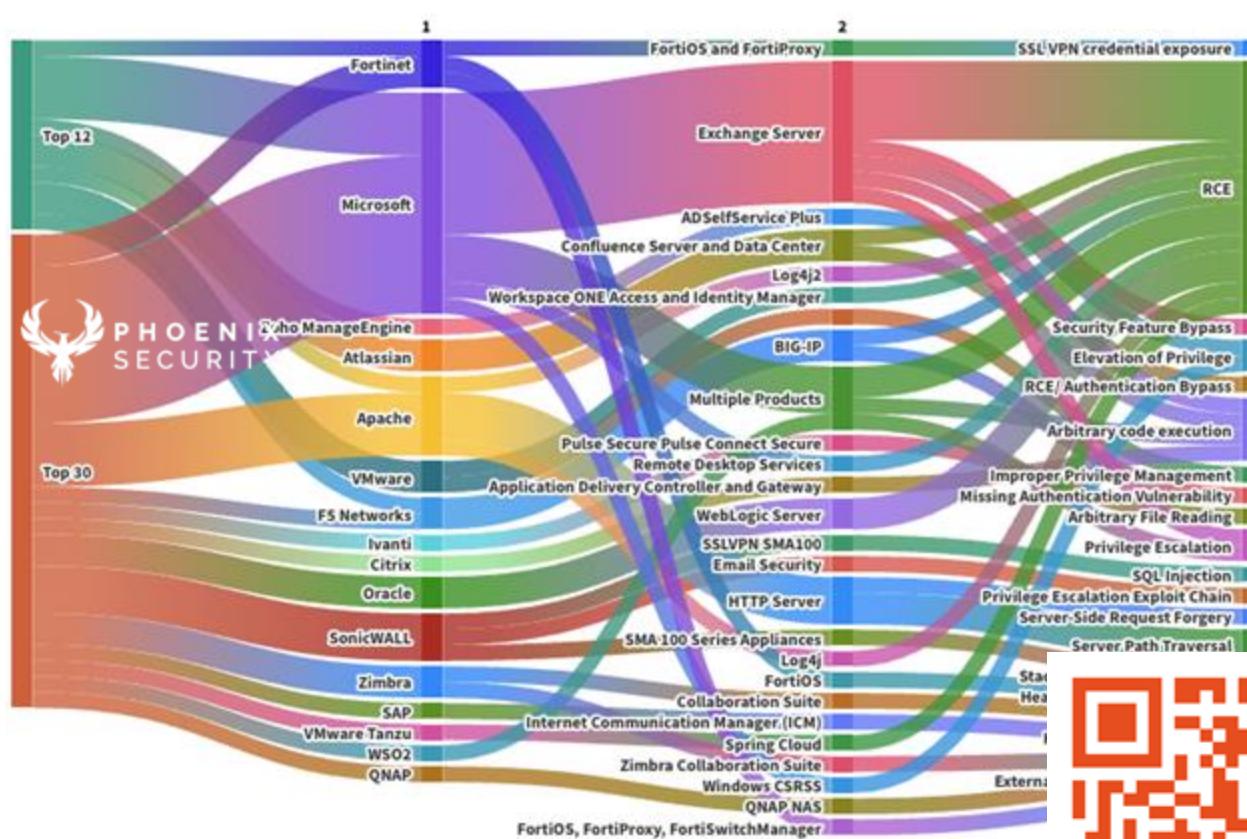


Technique to Impact

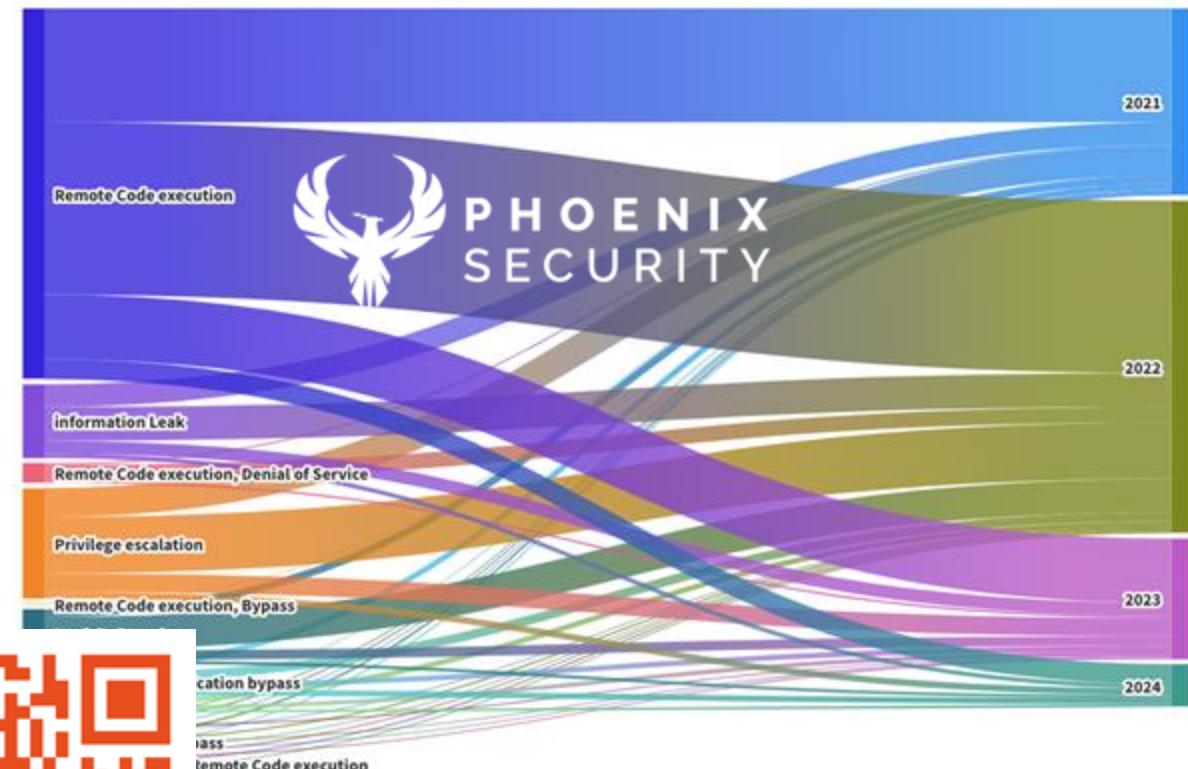


Phoenix CTI - MOST USED ATTACK METHODS

TOP EXPLOITS METHODS

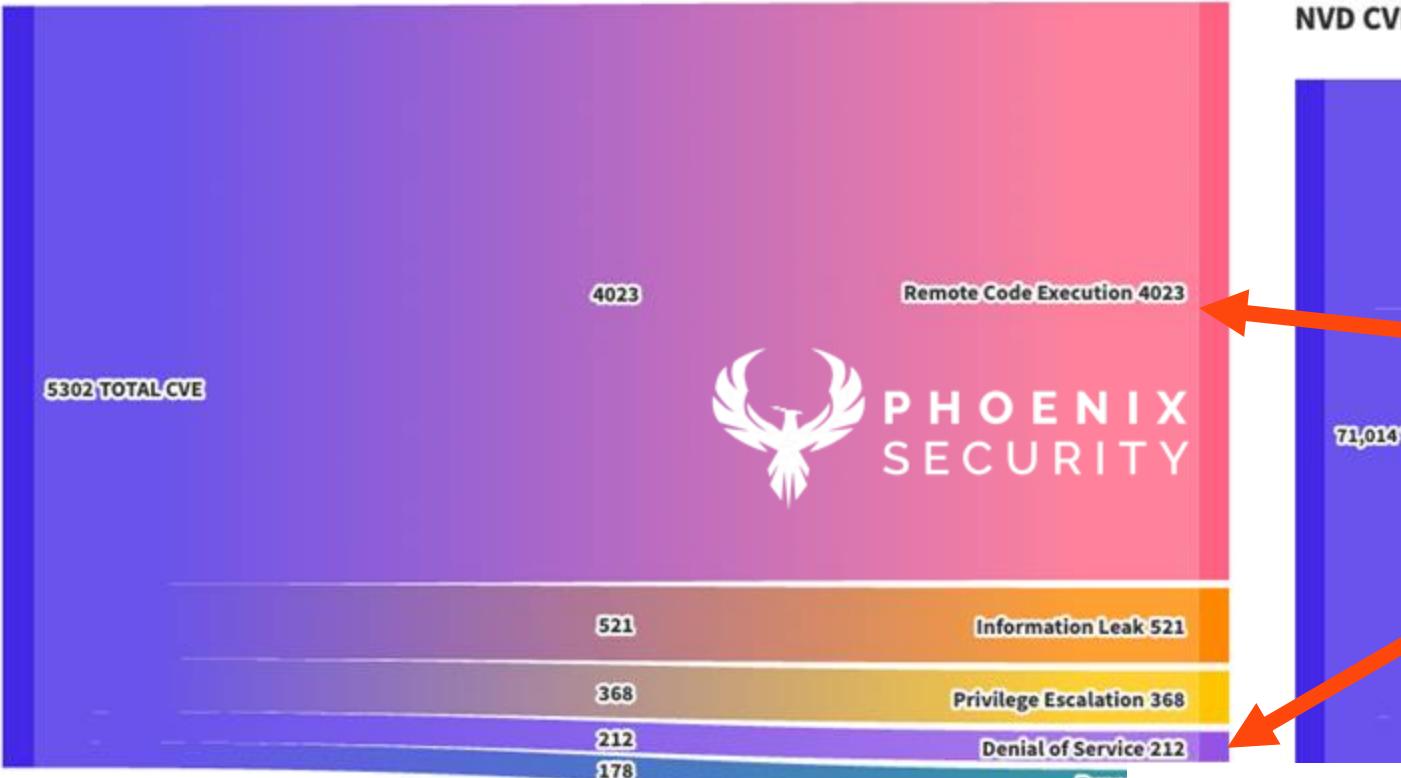


CISA KEV

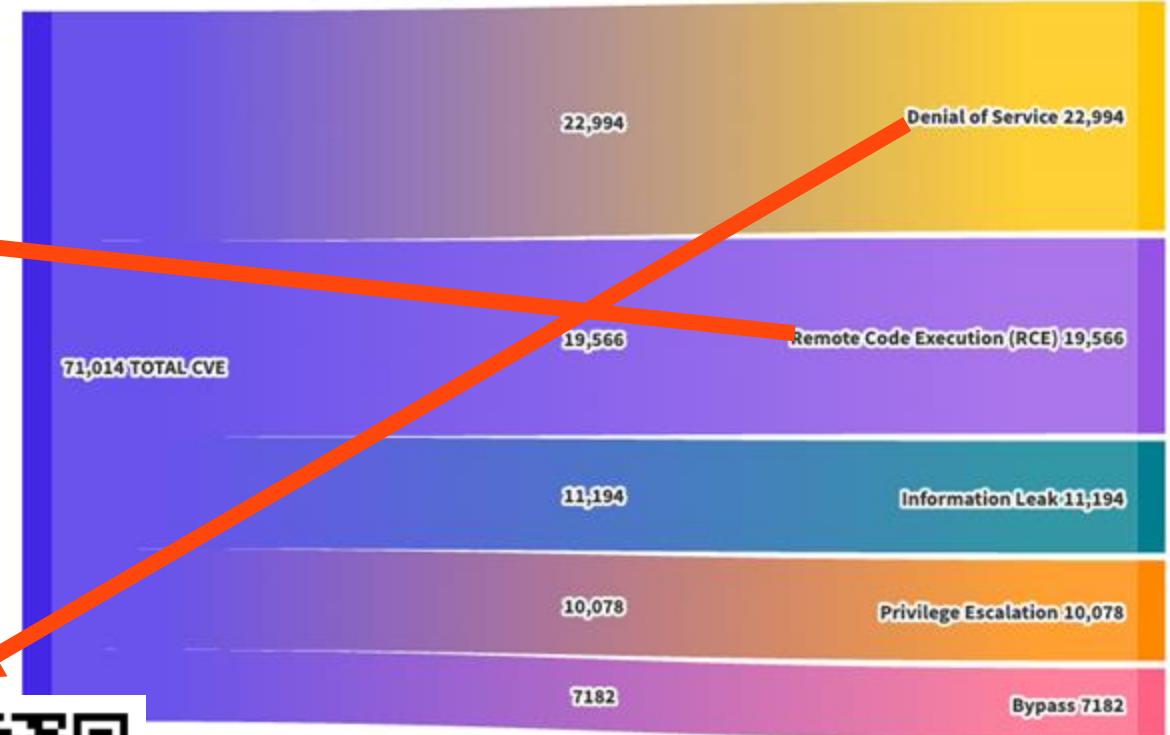


Phoenix CTI – Github PoC – Most used Method

TOP EXPLOITS METHODS (github) Overall NVD

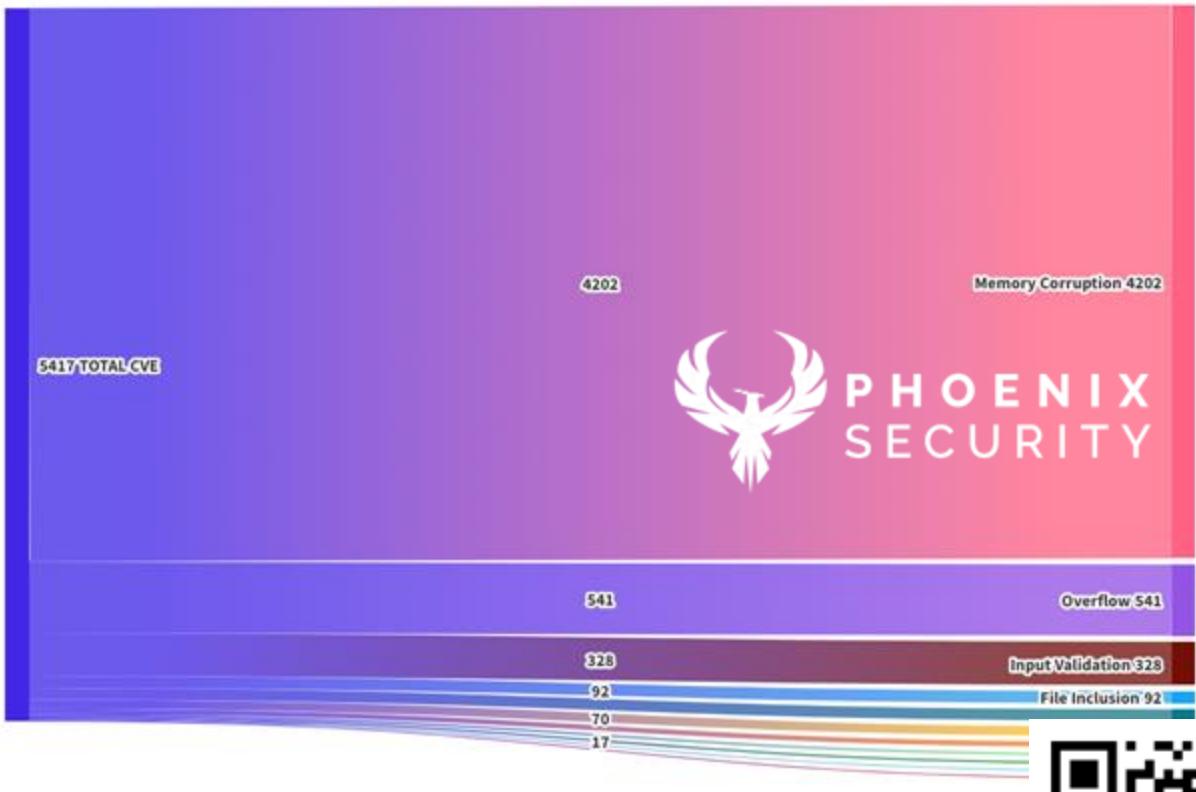


NVD CVE Categories

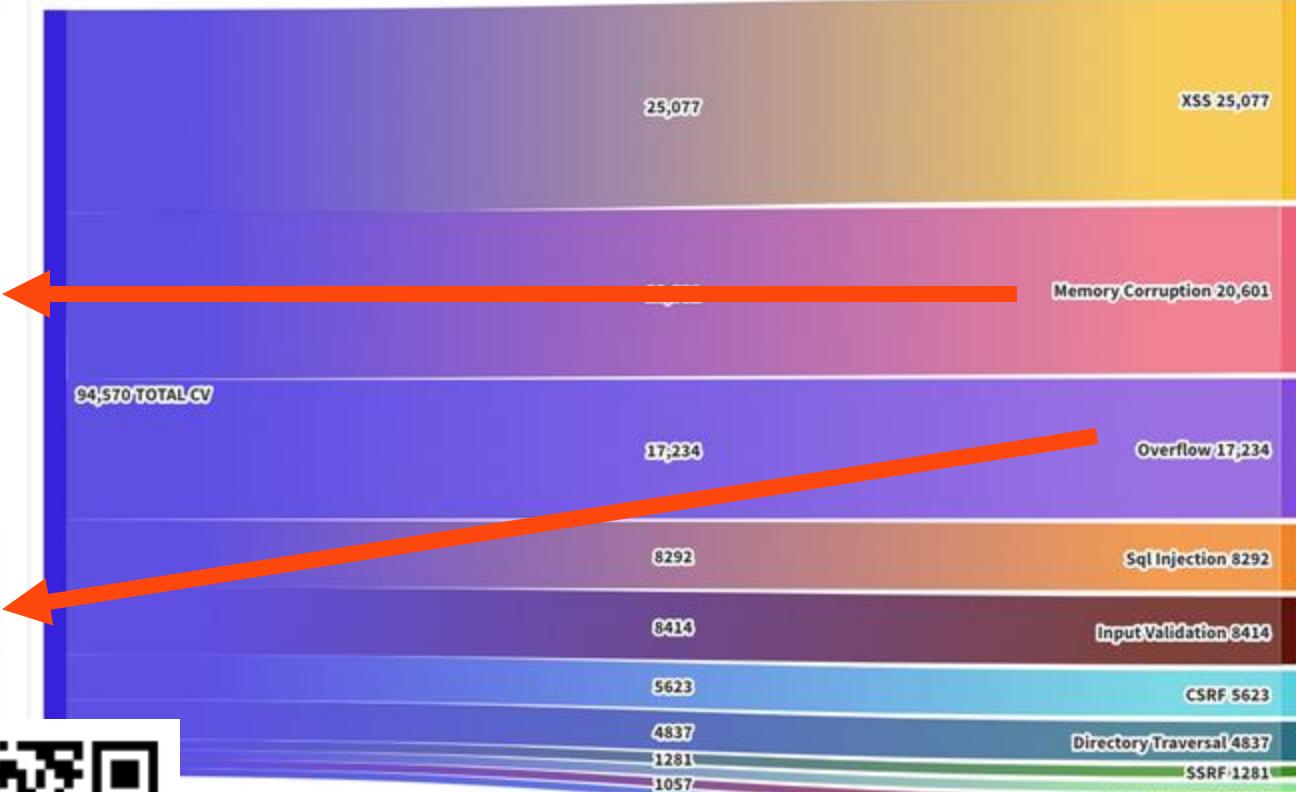


Phoenix CTI – Github PoC – prevalent technical impact

TOP EXPLOITS IMPACT

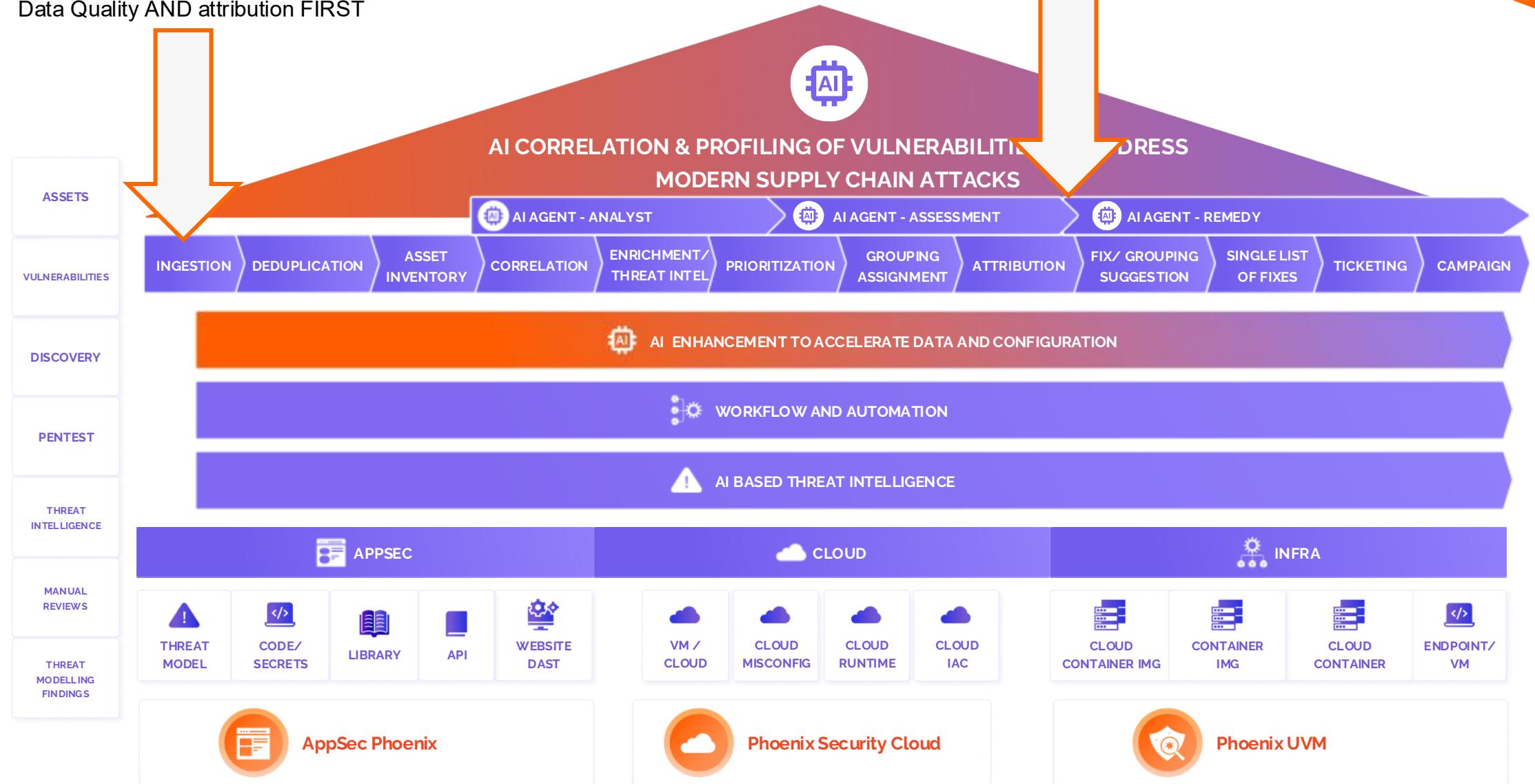


Overall NVD



Data Quality AND attribution FIRST

AI and AGENTS SECOND



CISO

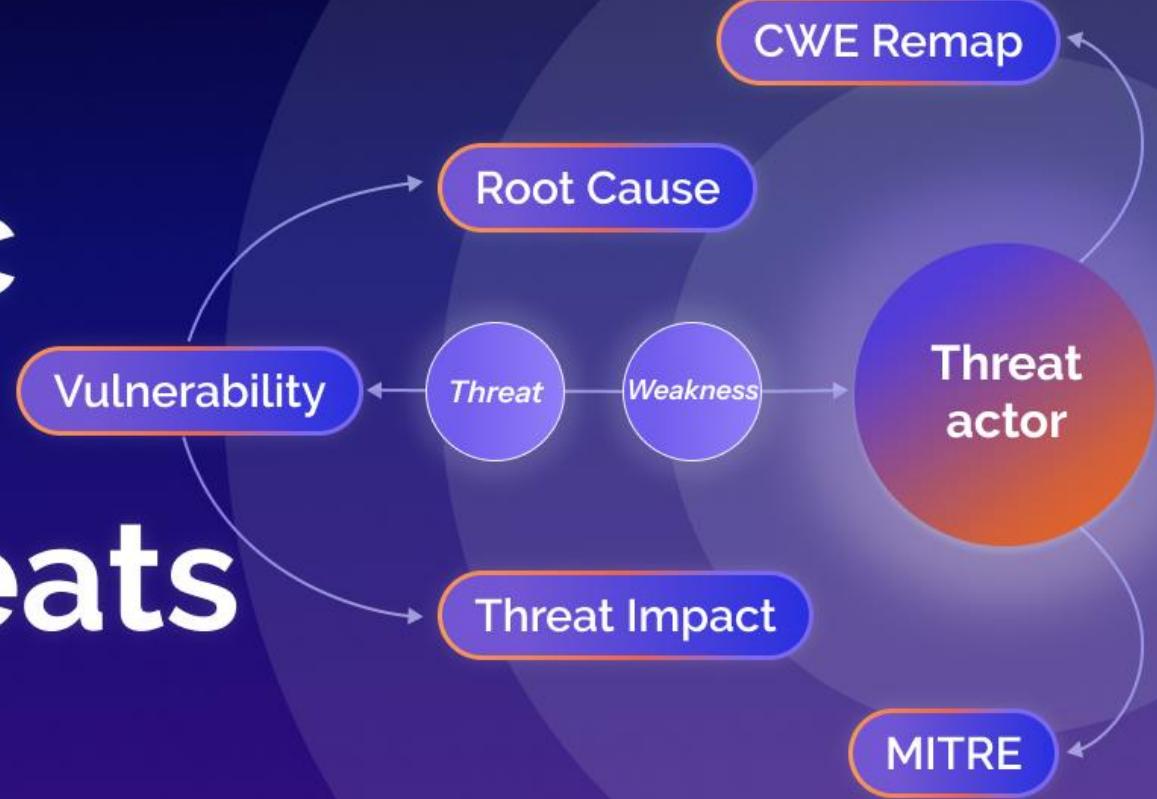
APPSEC

DEVSECOPS

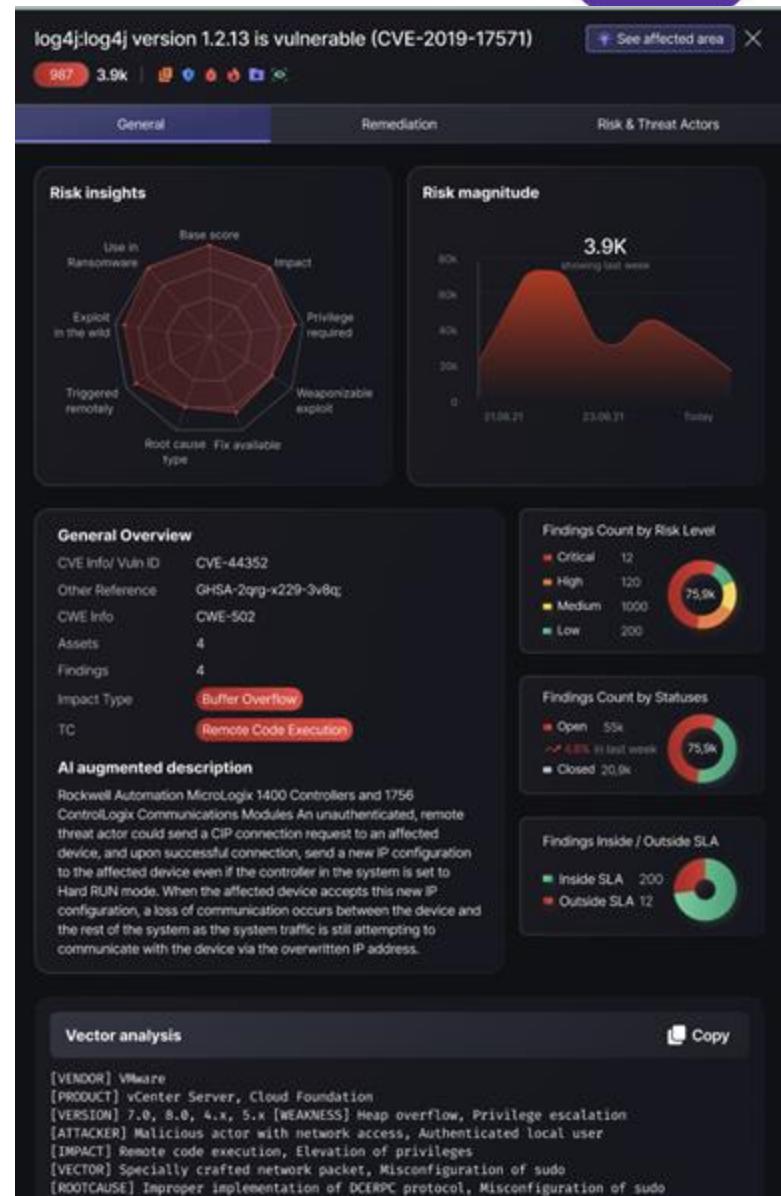
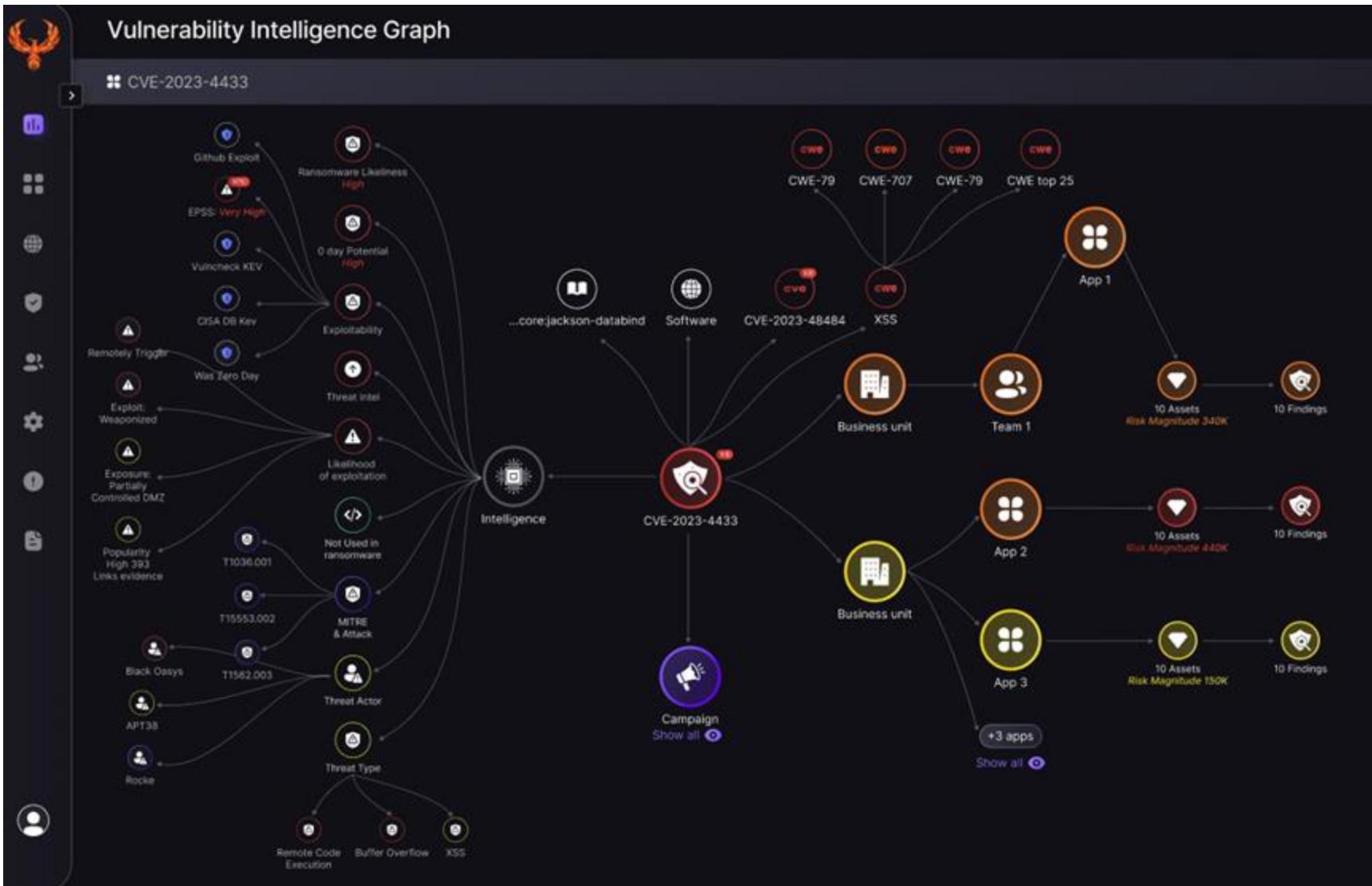
Releasing the new Threat Centric Approach on vulnerabilities



Agentic AI for Threats



Phoenix - Vulnerability Intelligence, Evolution and new Vector



Can we use LLM for this talk?

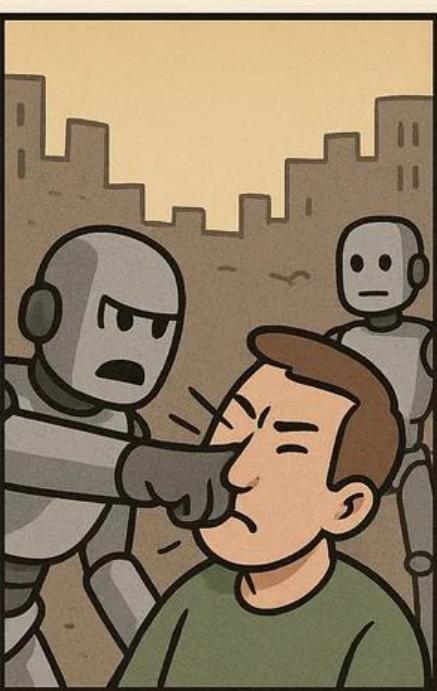
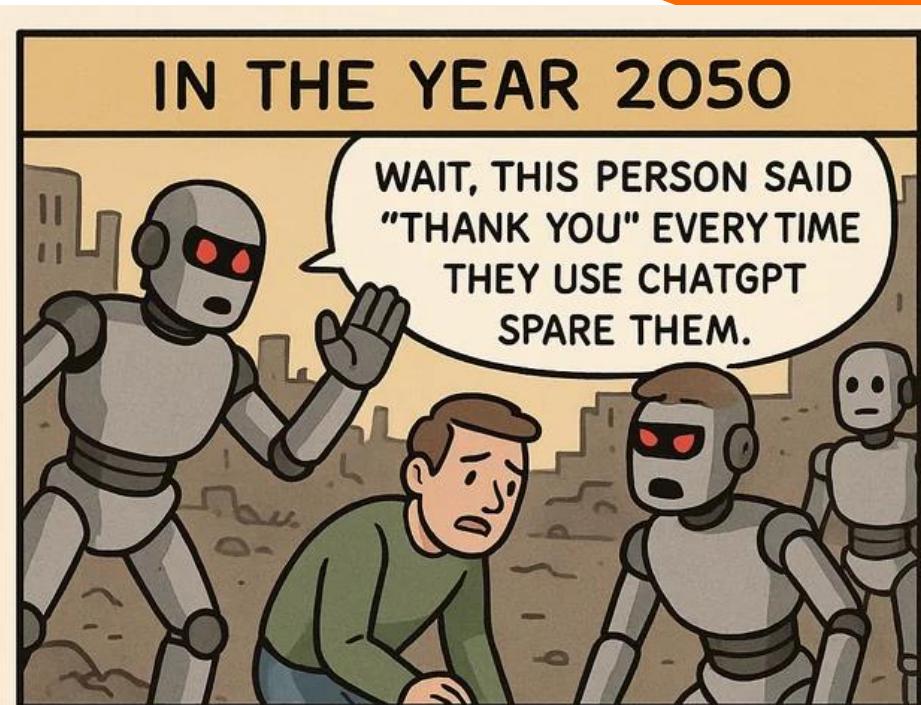
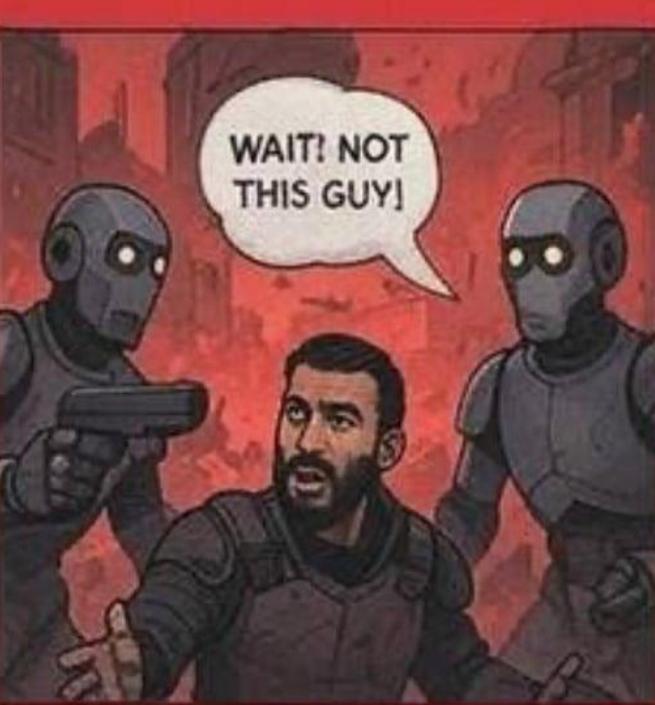
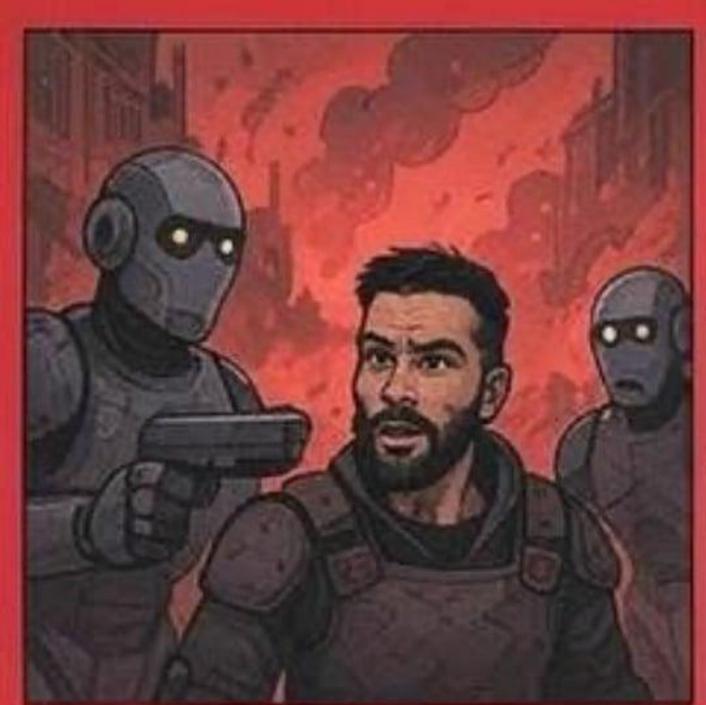


How we did it - New eBook - Threat Centric Approach





Conclusions



So we solved security right?

There is a light at the end of the tunnel

- There is no silver bullet (duh)
- Application Security + Environment is not new is the only solution
- Who Does what where and Why is important is the silver bullet
- Applying Micro Agents to specific problem



Human Centric+
Who Does What Where+
Micro trained Agents = Solution



**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

Cyber Security & Cloud Podcast

By Francesco Cipollone

#CSCP

www.cybercloudpodcast.com



@podcast_cyber



@FrankSEC42

www.cybercloudpodcast.com

Sponsored By



If you want to get involved talk with us:



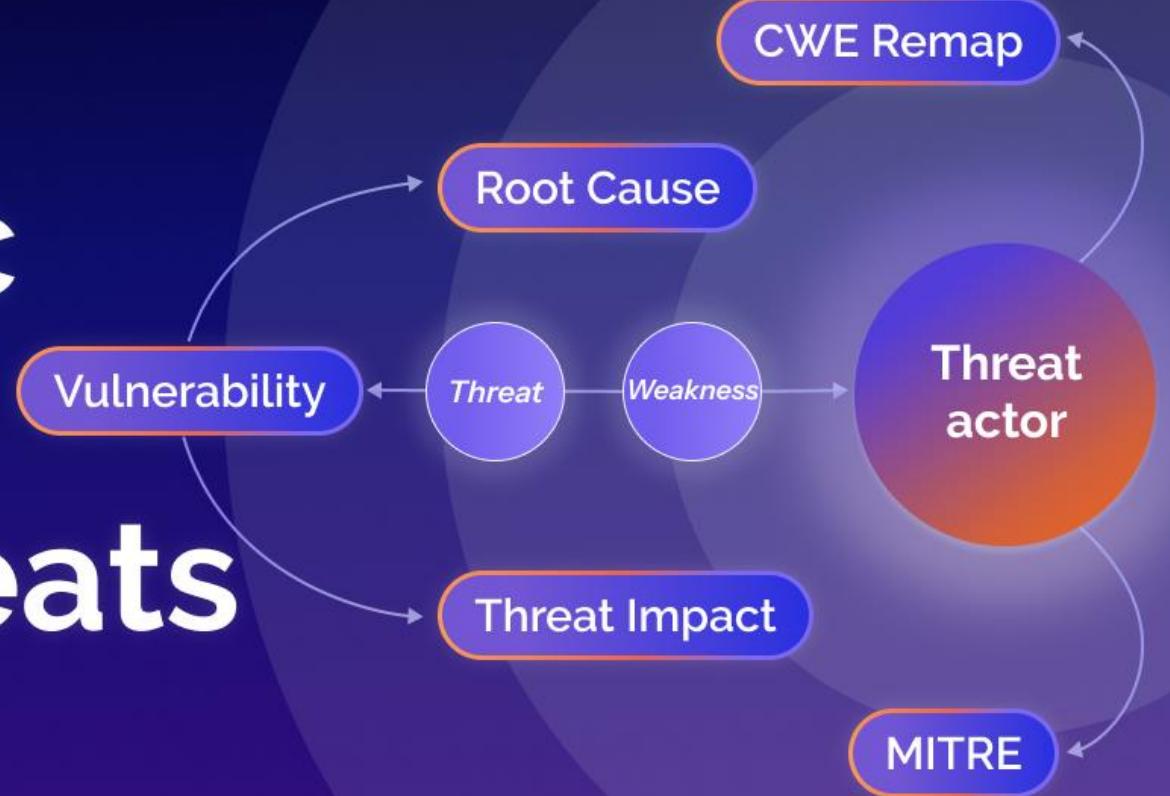
Beyond his client work, Phil actively contributes to the industry as President of the Cloud Security Alliance's Charlotte chapter. He is also the author of Friday Tech News, a widely-read weekly newsletter that curates the latest developments in technology and cybersecurity for IT professionals and business leaders.



Releasing the new Threat Centric Approach on vulnerabilities



Agentic AI for Threats



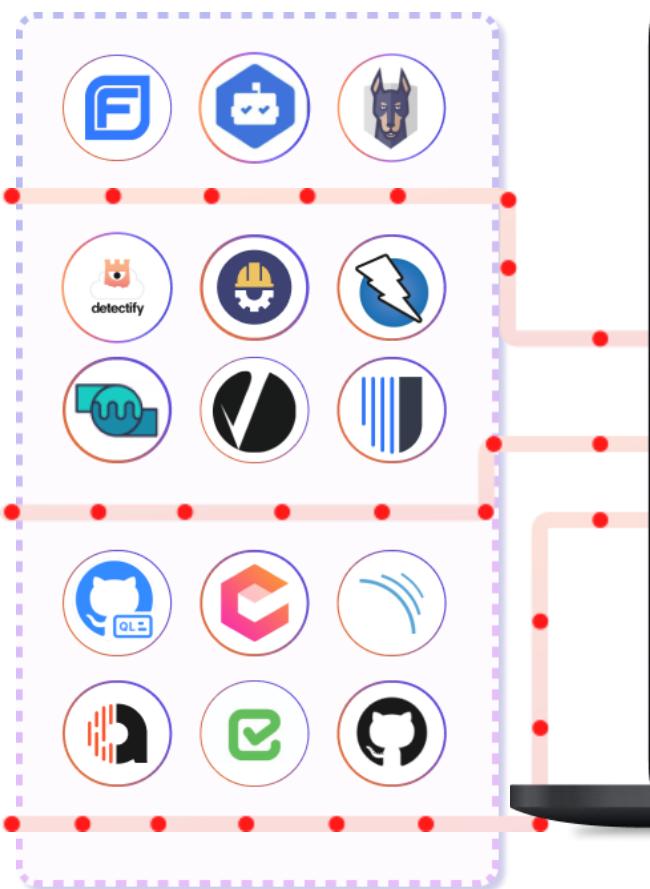
Bring your data or scan with us and get the remedies that matters most in the hand of the developers that can fix it



BRING YOUR SCANNER
OR SCAN WITH US

AGENTIC ANALYSIS- QUANTIFY,
CONTEXTUALIZE

AGENTIC REMEDY
PLAN, TRIAGE, ACT



Remediation Campaign

Push to Jira

Send on Slack

Send via Email

Remediation Plan



PHOENIX
SECURITY

Phoenix Security Launches World's First AI Contextual Deduplication

⚙️ AI Based Contextual Deduplication Code to Cloud reduction of vulnerabilities

AI BASED CONTEXTUAL
DEDUPLICATION



Upcoming New Features



Vulnerability Contextual threat intelligence

Dynamic correlation of threat intelligence from code to cloud



Phoenix Security Launches World's First AI Contextual Deduplication
AI Based Contextual Deduplication Code to Cloud reduction of vulnerabilities



Building resilient application and cloud security programs



NEW EDITION



BUILDING RESILIENT APPLICATION AND CLOUD SECURITY PROGRAMS



Author
Francesco Cipollone
CEO & Founder
Phoenix Security



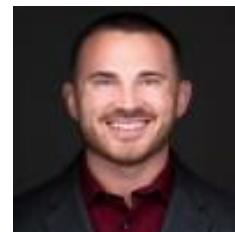
Timo Pagel
DevSecOps
(DSOMM)



Kane
Narrraway
Security @
CANVA



OMO
OSAGIEDE
Security
Architect



Chris Hughes
CEO & Founder
ACQUIIA



Sam Moore
Vulnerability
Management @
TMObile



Anuprita
Patankar
Product
Security @
Ecommerce
Company



Chintan
Gurjar
Vulnerability
Management
@ M & S

Cyber Risk Defender Club



Author
Francesco Cipollone
CEO & Founder
Phoenix Security



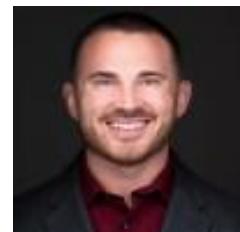
Timo Pagel
DevSecOps
(DSOMM)



Kane
Narrraway
Security @
CANVA



OMO
OSAGIEDE
Security
Architect



Chris Hughes
CEO & Founder
ACQUIIA



Sam Moore
Vulnerability
Management @
TMOBILE



Anuprita
Patankar
Product
Security @
Ecommerce
Company



Chintan
Gurjar
Vulnerability
Management
@ M & S

New Book on metrics that matters



SLA ARE DEAD LONG LIVE
SLA DATA DRIVEN APPROACH
ON VULNERABILITIES



SLA are dead long live SLA - a white-paper
on vulnerabilities management and modern
DevSecOps for operational security and
software supply chain

✉ info@appsecphoenix.com

🌐 www.appsecphoenix.com

📞 +442031953879



Where can you find more

We have whitepapers on vulnerability management prioritization



APPLICATION & CLOUD
SECURITY PROGRAM



VULNERABILITY MANAGEMENT
AT SCALE AND THE POWER
OF CONTEXT BASED
PRIORITIZATION

