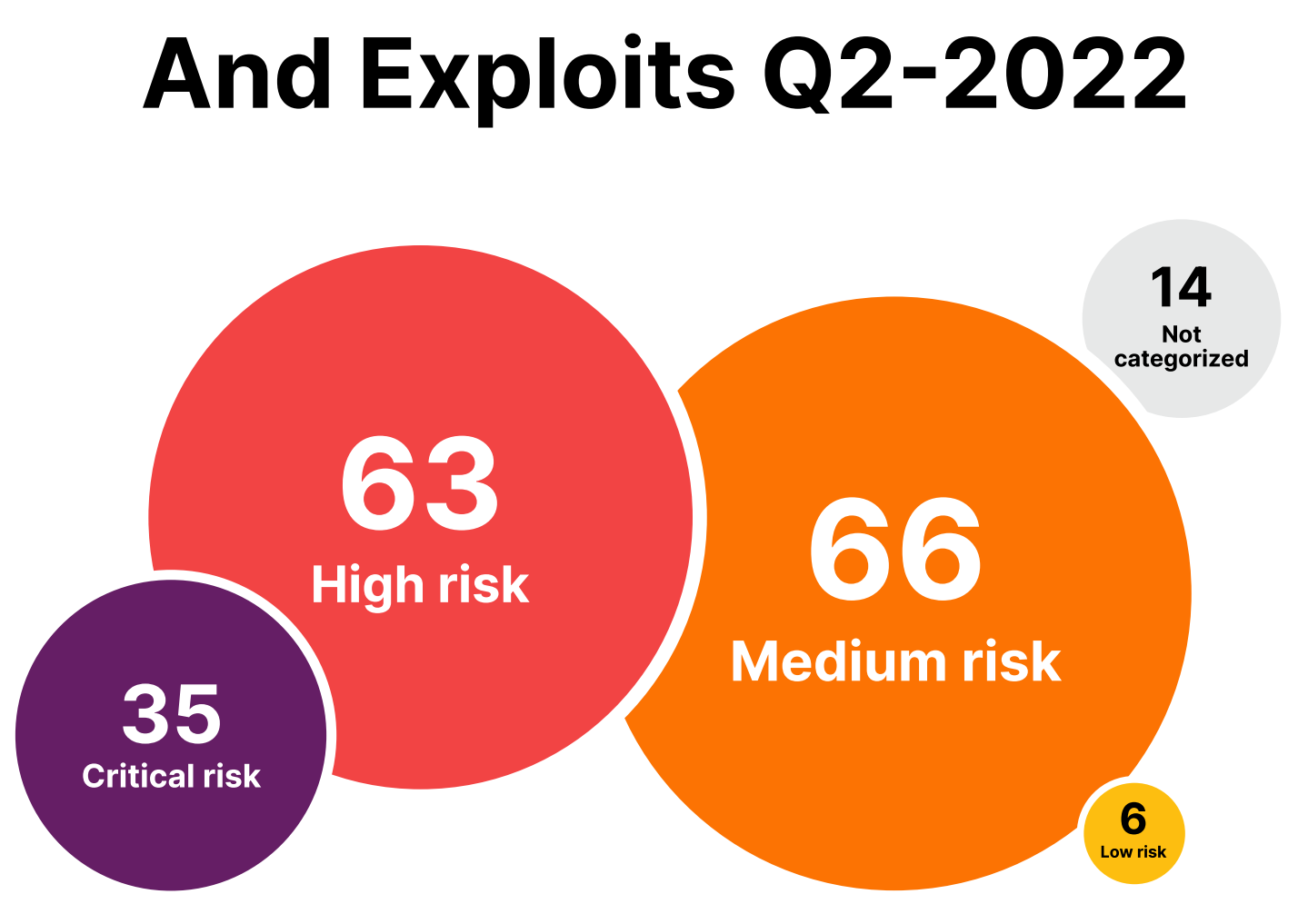
wallarm

INFOGRAPHIC

API Vulnerabilities And Exploits Q2-2022



API Vulnerabilities Up 3.7x!

Read on to Learn Why – and How to Evaluate Your Risk.

Earlier this year, Gartner wrote that "by 2022, API abuses will move from an infrequent to the most frequent attack vector, resulting in data breaches for enterprise web applications." [1] Midway through the year, is this being proved true by the facts on the ground? Is the threat real?

To address this, Wallarm examined API vulnerabilities and exploits that were publicly disclosed in Q2-2022, and what types of software from which vendors are involved.

We also analyzed publicly disclosed exploit POCs to determine where the risk lies.

In addition, we map these issues across industry standards, including both **OWASP Top-10 (2021)** for web apps and **OWASP API Security Top-10 (2019)**, **CVSS scores**, and **CWEs**.

Use this data both to assess your exposure and to reduce the risk in your API portfolio.

[1] Gartner, Magic Quadrant for Application Security Testing (ID G00733839)

Huge Jump in reported Vulnerabilities!

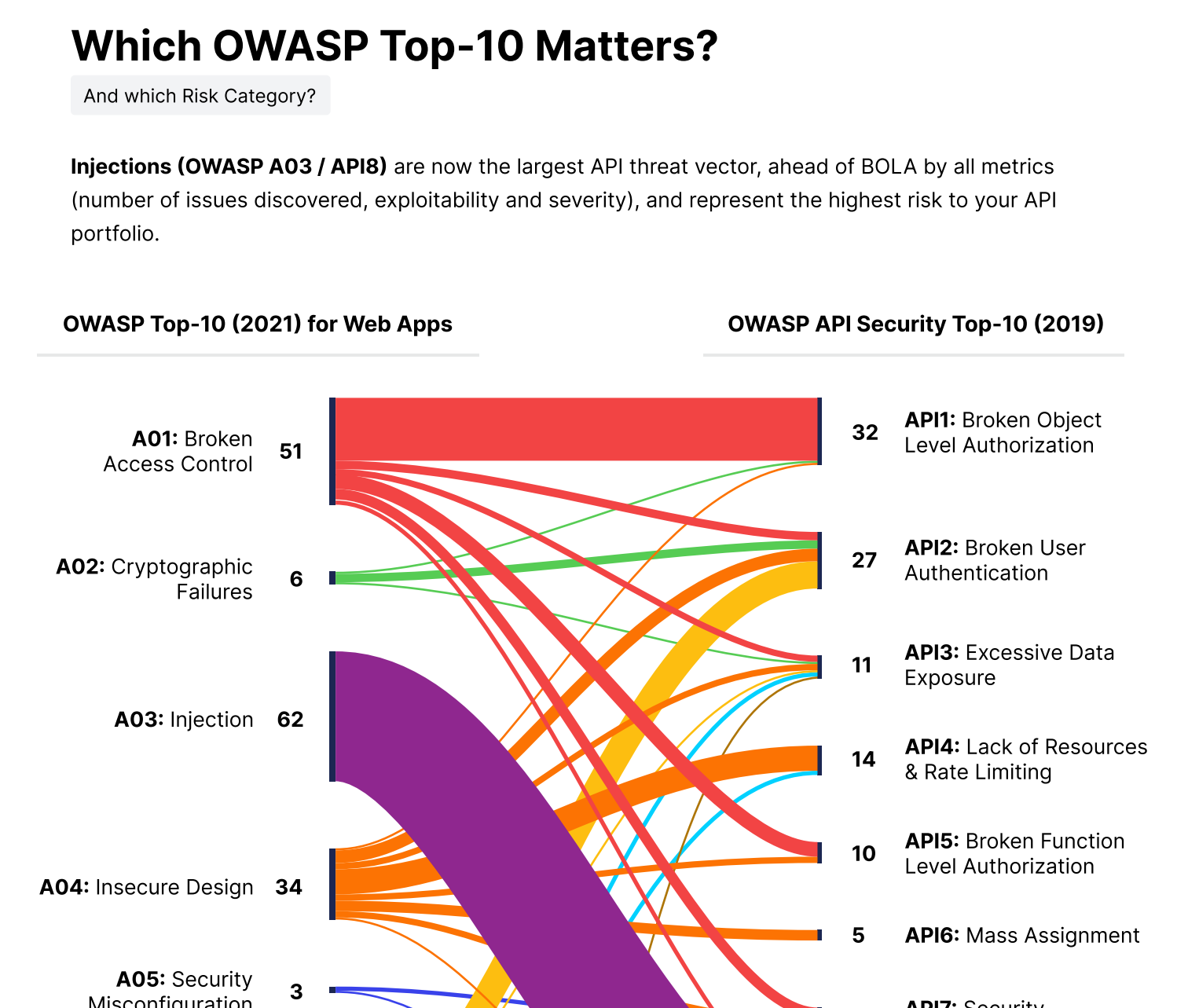
+268%	+270%	+90%
Vulnerabilities	Vendors	Critical & High rated vulnerabilities
Q2: 184	Q2: 111	Q2: 53%
Q1: 50	Q1: 30	Q1: 28%

Which OWASP Top-10 Matters?

And which Risk Category?

Injections (OWASP A03 / API8) are now the largest API threat vector, ahead of BOLA by all metrics (number of issues discovered, exploitability and severity), and represent the highest risk to your API portfolio.

OWASP Top-10 (2021) for Web Apps | **OWASP API Security Top-10 (2019)**



Lower CVSS Average, but Less Secure?

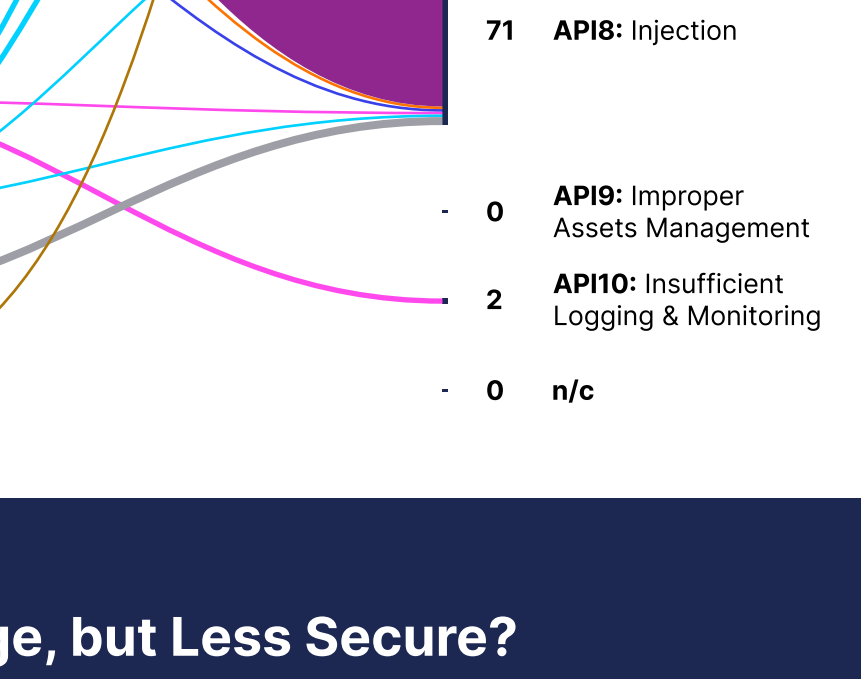
Careful Assessment Required

The average CVSS score in Q2-2022 is 7.1 – compared to 7.6 in Q1. But a deeper analysis shows more Critical and High rated vulnerabilities – which suggests that overall risk actually went up in Q2.

Q2: 7.6

Q1: 7.1

Average CVSS Score



Are CWEs Better Indicators of Risk?

Software Weaknesses Result in More Risk

2022 CWE Top 25 Most Dangerous Software Weaknesses

Rank	ID	Q2 count*	Name
1	CWE-787	n/a	Out-of-bounds Write
2	CWE-79	24	Cross-site Scripting
3	CWE-89	6	SQL Injection
4	CWE-20	11	Improper Input Validation
5	CWE-125	n/a	Out-of-bounds Read
6	CWE-78	15	OS Comand Injection
7	CWE-416	n/a	Use After Free
8	CWE-22	6	Path Traversal
9	CWE-352	4	Cross-Site Request Forgery (CSRF)
10	CWE-434	4	Unrestricted Upload of a File with Dangerous Type
11	CWE-476	n/a	NULL Pointer Dereference
12	CWE-502	1	Deserialization of Untrusted Data
13	CWE-190	n/a	Integer Overflow or Wraparound
14	CWE-287	4	Improper Authentication
15	CWE-798	1	Use of Hard-coded Credentials
16	CWE-862	1	Missing Authorization
17	CWE-77	1	Command Injection
18	CWE-306	2	Missing Authentication for Critical Function
19	CWE-119	n/a	Memory Buffer Overflow
20	CWE-276	2	Incorrect Default Permissions
21	CWE-918	3	Server-Side Request Forgery (SSRF)
22	CWE-362	0	Race Condition
23	CWE-400	7	Incontrolled Resource Consumption
24	CWE-611	2	Improper Restriction of XML External Entity Reference
25	CWE-94	0	Code Injection

*n/a means this CWE is **not** related to API Security, only for binary software / memory corruption bugs

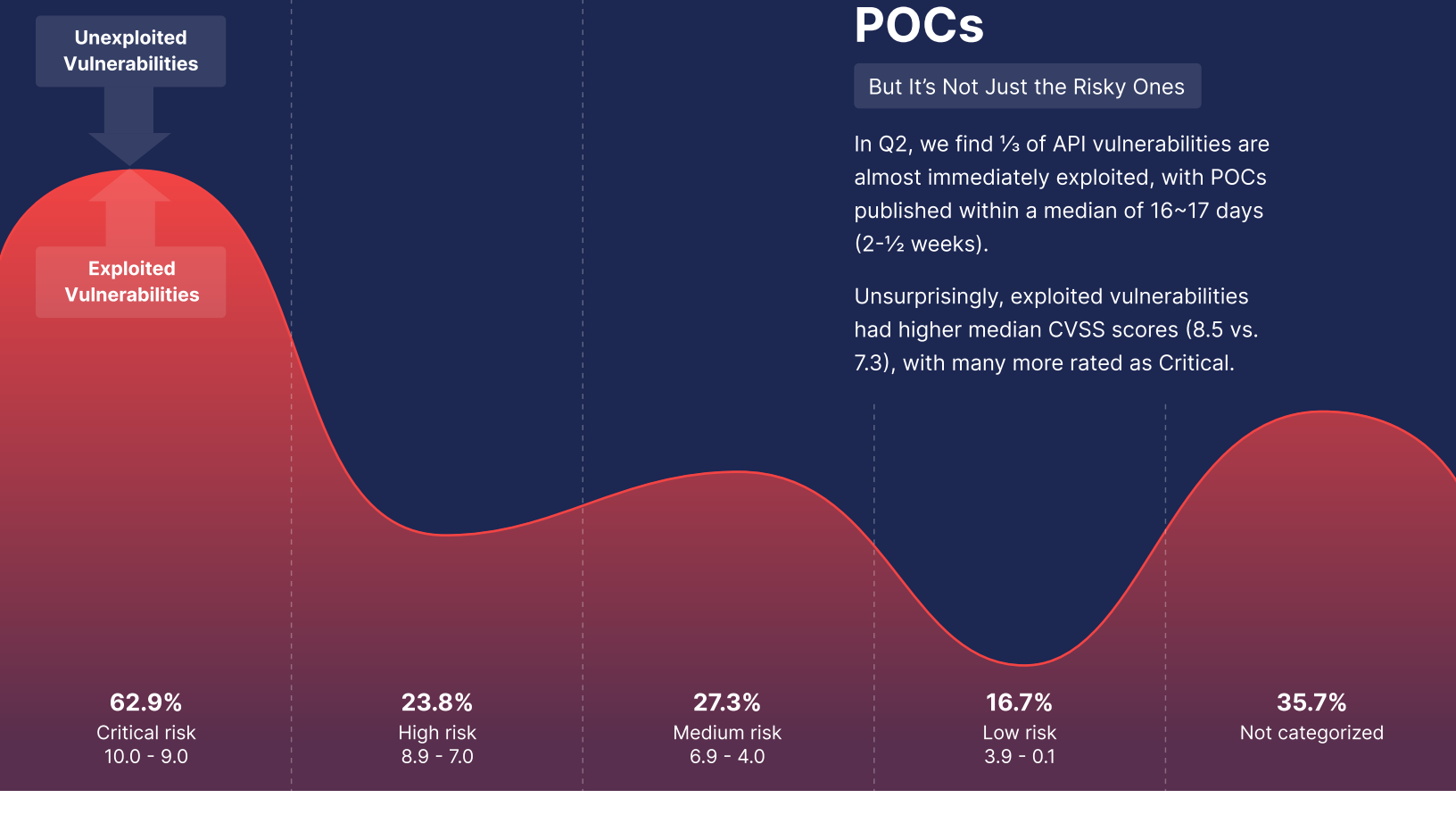
50% of the Q2 vulnerabilities analyzed referenced CWEs which are included in the 2022 CWE Top 25 Most Dangerous Software Weaknesses list from MITRE / CISA[2].

67 unique CWEs found in Q2 reports

17 of these are considered "most dangerous"

Most seen: CWE-79, CWE-78 and CWE-20

[2] https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html



Published Exploit POCs

But It's Not Just the Risky Ones

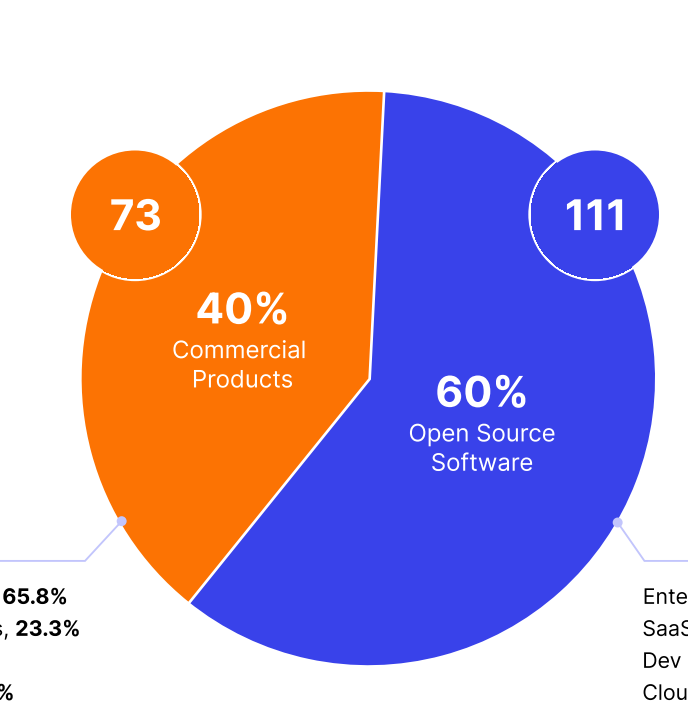
In Q2, we find ⅓ of API vulnerabilities are almost immediately exploited, with POCs published within a median of 16~17 days (2~½ weeks).

Unsurprisingly, exploited vulnerabilities had higher median CVSS scores (8.5 vs. 7.3), with many more rated as Critical.

Is Open Source Really More Secure?

Maybe not.

60% of reported vulnerabilities are related to Open Source software with the remainder related to Commercial products. Overall, the Dev Tools and Enterprise HW / SW subcategories accounted for almost 80% of the total.

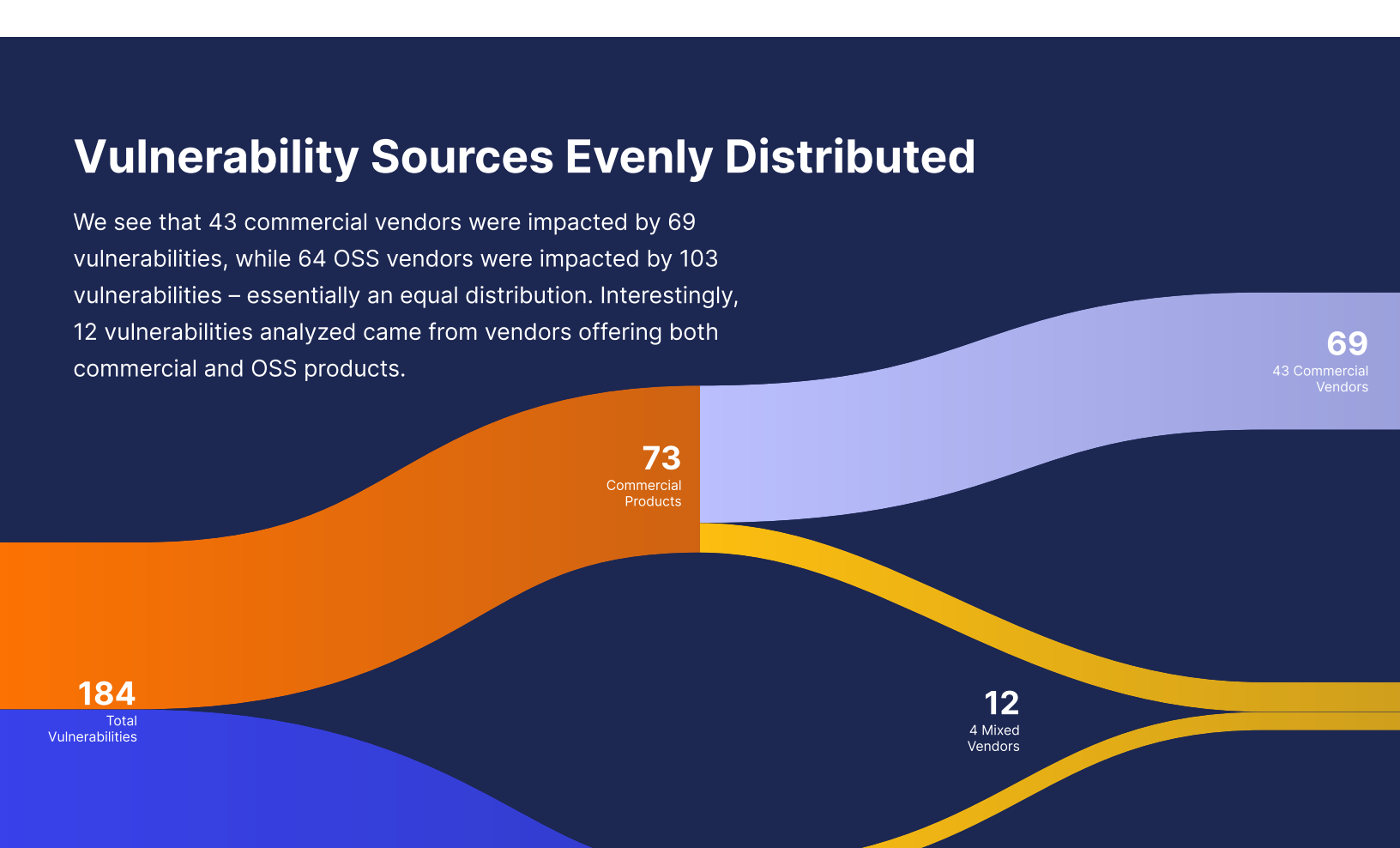


Enterprise HW / SW, 65.8%
SaaS / Web Services, 23.3%
Dev Tools, 6.8%
Cloud Platforms, 4.1%

Enterprise HW / SW, 20.7%
SaaS / Web Services, 11.7%
Dev Tools, 63.1%
Cloud Platforms, 4.5%

Vulnerability Sources Evenly Distributed

We see that 43 commercial vendors were impacted by 69 vulnerabilities, while 64 OSS vendors were impacted by 103 vulnerabilities – essentially an equal distribution. Interestingly, 12 vulnerabilities analyzed came from vendors offering both commercial and OSS products.



What's In Your API Portfolio?

More Vulnerabilities Impacting More Vendors

We see a huge increase in the number of vendors with reported API vulnerabilities, up 3.7x or 270% (from 30 in Q1 to 111 in Q2), mirroring the overall growth.

While a vast majority of vendors (71%) are impacted by only 1 vulnerability, we find 9 vendors (8%) which were impacted by 4 or more vulnerabilities.

Top-5 Most Impactful API Vulnerabilities

We assess these to be the most impactful API vulnerabilities due to both the severity and reach of the product. Notably, while the CVSS score for the GitLab vulnerability is low, it's used by almost every developer in the world so it had to be included.

f5 — CVSSv3: **9.8**
CVE-2022-1388 BIG-IP iControl REST authentication bypass
CWE-306 Missing Authentication for Critical Function
A07 OWASP API Top-10
API2 OWASP API Security Top-10

WSO2 — CVSSv3: **9.8**
CVE-2022-29464 WSO2 products unrestricted file upload with resultant remote code execution
CWE-434 Unrestricted Upload of File with Dangerous Type
A04 OWASP API Top-10
API4 OWASP API Security Top-10

vmware — CVSSv3: **9.8**
CVE-2022-22980 Spring Data MongoDB SpEL Injection
CWE-917 Improper Neutralization of Special Elements used in an Expression Language Statement
A03 OWASP API Top-10
API8 OWASP API Security Top-10

GitLab — CVSSv3: **2.7**
CVE-2022-1783 GitLab CE/EE Improper Privilege Management
CWE-269: Improper Privilege Management
A04 OWASP API Top-10
API5 OWASP API Security Top-10

argo — CVSSv3: **10.0**
CVE-2022-29165 Argo CD will trust invalid JWT claims if anonymous access is enabled
CWE-287 Improper Authentication, CWE-290 Authentication Bypass by Spoofing, CWE-200
A07 OWASP API Top-10
API2 OWASP API Security Top-10

Be On The Lookout for These Too

What to address first? Triaging vulnerabilities for mitigation can be based on a variety of criteria, including:

Ranking based on frequency and severity, much like how MITRE assesses CWEs[3]

Frequency how many vulnerabilities are found in the vendor's products?

Severity how bad are the vulnerabilities in a particular vendor's products?

Top-5 based on ranking (frequency x CVSS)

Vendor	count	CVSS avg
Robustel	9	9.1
Ruijie Networks	6	9.0
Argo Project	6	8.2
Wordpress	7	6.8
Strapi	4	7.2

Top-5 based on frequency (count)

Vendor	count	CVSS avg
Robustel	9	9.1
Ruijie Networks	6	9.0
Argo Project	6	8.2
Wordpress	7	6.8
Gitlab	5	4.4

Top-10 based on severity (CVSS average)

Vendor	count	CVSS avg
Illumina	1	10.0
Pypi	2	9.8
ApolloGraphQL	1	9.8
Bonita Software	1	9.8
Couchbase	1	9.8
Form.io	1	9.8
git-pull-or-clone	1	9.8
IObit	1	9.8
MediaWiki	1	9.8
Powertek	1	9.8
Roncoo	1	9.8
Zyxel	1	9.8

Of course, ultimately it comes down to what is in your environment, how exposed it is, and how easy it is to exploit.

[3] See https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25_supplemental.html#methodDetails

Assessing Your API Security

API-specific vulnerabilities reported in Q2 grew by 268% to 184 (or about 2 per day) – which suggests an ever-increasing risk in your API portfolio.

The number of Critical and High risk API vulnerabilities have increased dramatically – which also indicates that extra vigilance is needed.

Injections (OWASP A03 / API8) are now the highest risk for APIs, ahead of BOLA by all metrics (number of discovered issues, exploitability and severity) – which points to the need for more pre-release testing.

33% of the reported API vulnerabilities are almost immediately exploited, with POCs published within a median of 2-½ weeks – since these are probably underreported, this illustrates the need for run-time protection.

Expanding your vulnerability management program to cover APIs will require visibility across your entire API portfolio, assessing and triaging vulnerabilities as they arise, and ensuring mitigations are implemented – both in the code and at run-time. Refer to the **API Security Tutorial** for more information.

Want to learn more about API vulnerabilities and exploits?

Join the LinkedIn 🔥 API security community 🔥 group at https://www.linkedin.com/groups/12624726/

Subscribe to our newsletter at lab.wallarm.com

Download the Q1-2022 API Vulnerability Report at https://www.wallarm.com/resources/api-vulnerabilities-discovered-and-exploited-in-q1-2022

Register for the upcoming Q2-2022 API Vulnerability Report webinar at http://lab.wallarm.com/2022-q2-vulnerability-report-webinar/

wallarm

(415) 940-7077
188 King St. Unit 508, San Francisco, CA 94107
www.wallarm.com