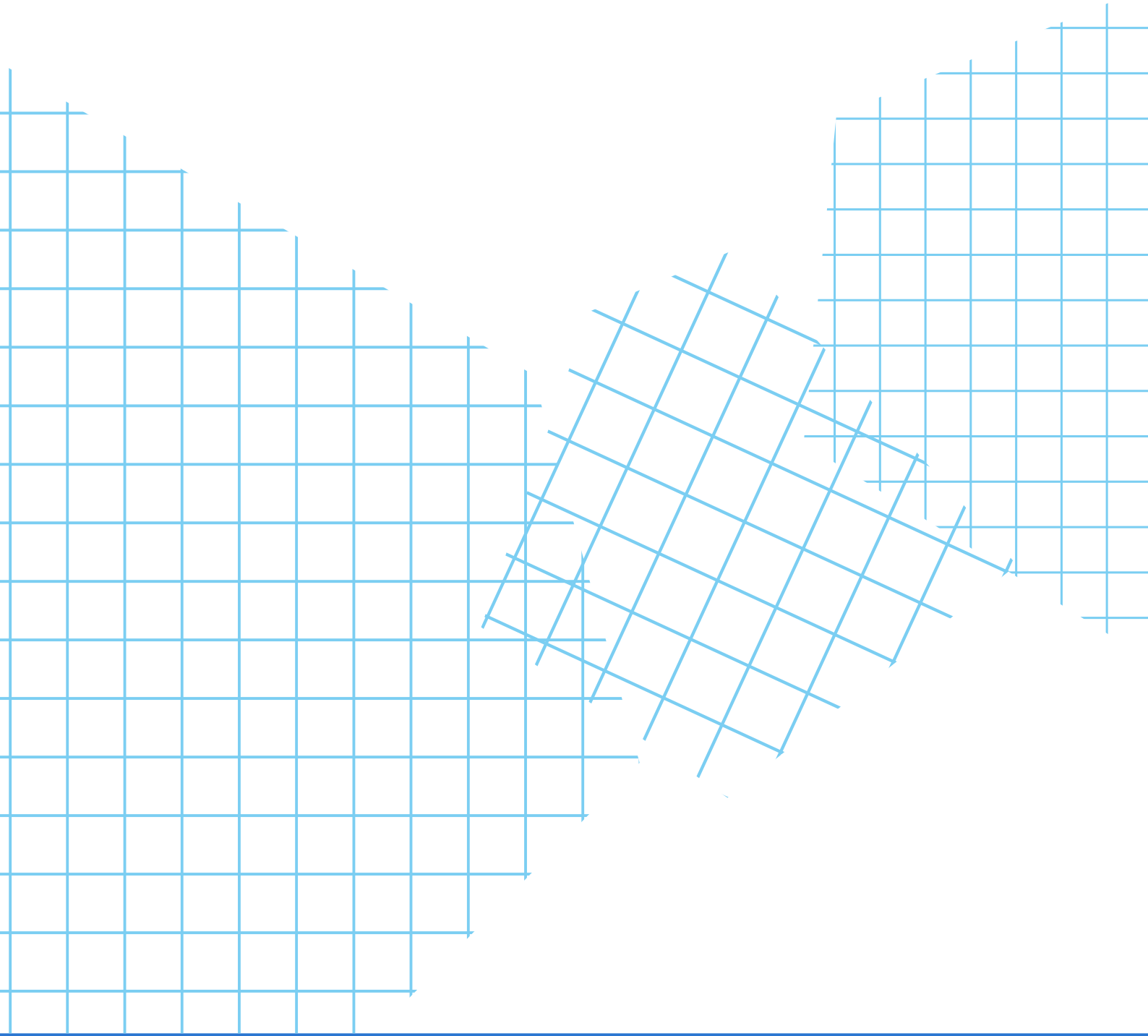
A comprehensive benchmark of web application vulnerability scanners 2024

Contents

Benchmark overview

- The analysis included the following web application vulnerability scanners:
  - Acunetix
  - Burp Scanner
  - Pentest-Tools.com Website Vulnerability Scanner
  - Qualys
  - Rapid7 InsightAppSec
  - ZAP (Zed Attack Proxy)
- The tests were performed against Broken Crystals, for its use of a wide range of modern technologies and the vulnerabilities it exposes, and DVWA (Damn Vulnerable Web Application), because it still reflects a significant percentage of the types of web apps found across the internet
- For those wishing to independently confirm the findings, it is essential to acknowledge that all scanners were updated with the latest detections as of February 2024.
- All tools were configured to use their most comprehensive crawling strategy and try all the available vulnerability detections.
- Where available, the REST API swagger files that defined the API were specified, as well as the GraphQL endpoint to be scanned.
- Where possible, each scanner was configured to run and try to validate if the authentication was successful.
- All scanning activities unfolded throughout February 2024.

- To guarantee a uniform and impartial evaluation, the analysis relied on three performance indicators:
  - **true positive count:** how many of the vulnerabilities the scanner reported actually existed
  - **false positive count:** how many of the vulnerabilities the scanner reported were not actually there
  - **false negative count:** how many existing vulnerabilities were not reported by the scanner.



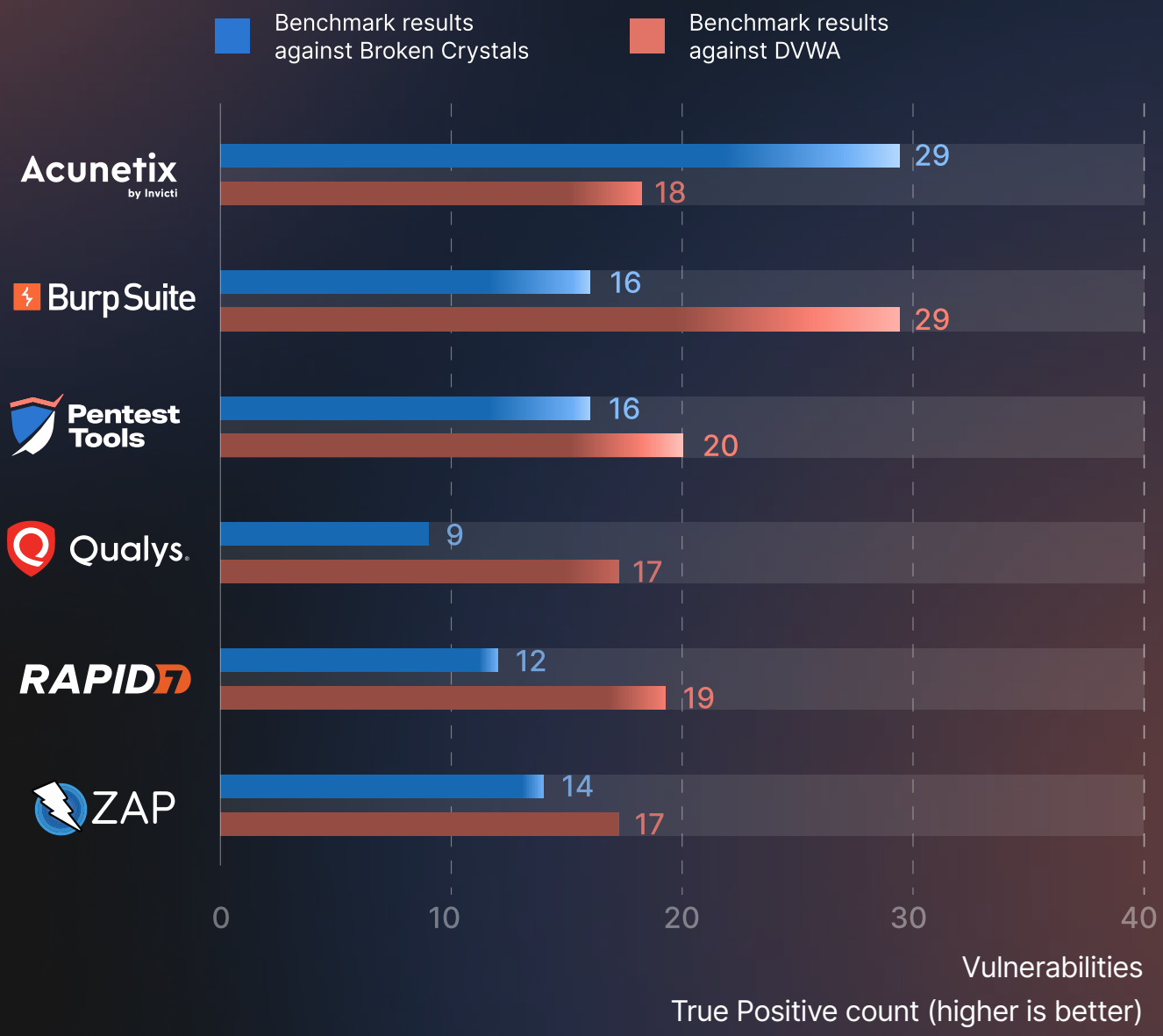
Key findings from the benchmark

The benchmark demonstrates a comparable level of vulnerability detection, albeit with minor discrepancies, among prominent commercial scanners and the main open-source contender, ZAP. Exceptions to this uniformity include Burp Suite in its scans against the Damn Vulnerable Web Application (DVWA) and Acunetix in the context of Broken Crystals. These observations are particularly pertinent given the claims by commercial vulnerability scanning solutions of extensive coverage across the majority of known vulnerabilities. More detailed insights into these exceptions are provided below.

In the examination of False Positives associated with Broken Crystals scans, all scanners reported some false positives, but the number identified by each was not substantial. This indicates a generally effective level of accuracy across the various scanning tools tested.

In terms of False Positives within the Damn Vulnerable Web Application (DVWA), the open-source tool ZAP exhibits considerable inaccuracies in its findings. Similarly, some commercial solutions, such as Qualys and Rapid7 InsightAppSec, also demonstrate a tendency to generate false positives. Conversely, the Pentest-Tools.com Website Vulnerability Scanner has been noted for maintaining a notably lower rate of false positives, thereby enhancing its reliability in security assessments.

Vulnerability detection across both targets



The true positive rate was calculated as

$$= \frac{\text{number of detected true positive vulnerabilities}}{\text{total number of true positive vulnerabilities reported by all scanners}} * 100$$

The false positive rate was calculated as

$$= \frac{\text{number of detected vulnerabilities which were false positive}}{\text{total number of false positive vulnerabilities reported by all scanners}} * 100$$

The false negative rate was calculated as

$$= \frac{\text{number of undetected true positive vulnerabilities}}{\text{total number of true positive vulnerabilities reported by all scanners}} * 100$$

Why this benchmark exists

Web security specialists commonly grapple with the challenge of comparing products for lack of standard benchmarks or information from vendors.

First-hand information is scarce and evaluating tool performance in a relevant way to particular use cases would require building a web security testing lab and a consistent testing methodology.

This is why benchmarks for web application vulnerability scanners are extremely sporadic. For instance, one of the few extensive benchmarks in the industry dates from 2017, when Shay Chen evaluated 10 web application vulnerability scanners, publishing the results after a 2-year long effort.

Another challenge is the constant flux of web security vulnerabilities and the evolution of the technology ecosystem. Both require scanners to perpetually refine their detection capabilities, making it exceedingly difficult to create a static benchmark that retains relevance over time.

Moreover, since a benchmark must be both adaptable to the vast diversity of vulnerabilities and capable of spanning a wide range of scenarios, choosing a universal evaluation metric is complicated.

Additionally, logistical hurdles, such as setting up the tools and securing the necessary accounts, present another layer of challenges.

And, lastly, the most formidable obstacle lies in the inherent discrepancy between controlled benchmarks and their real-world applicability. Despite meticulous design and rigorous execution, the outcomes from a real-world deployment are likely to differ from those obtained in a controlled setting.

Despite these complexities, the current demands of offensive security specialists underscores the critical need for such a benchmark.

Evaluated web application vulnerability scanners

Web application vulnerability scanners are complex tools designed to find vulnerabilities in web applications at runtime, from a black-box perspective.

These security tools sit in two primary categories based on their **licensing type**: **open-source** and **commercial**.

Open-source web application vulnerability scanners,

such as ZAP (Zed Attack Proxy), are freely available and can be modified and distributed under their respective licenses.

These scanners are highly attractive to organizations with robust technical teams, as they allow for customization and extension to meet specific operational requirements. Notable advantages of these tools include extensive scanning capabilities, community-driven detection updates for recent vulnerabilities, and the flexibility to integrate with other security solutions.

However, they generally demand greater efforts for setup and ongoing maintenance compared to commercial products.

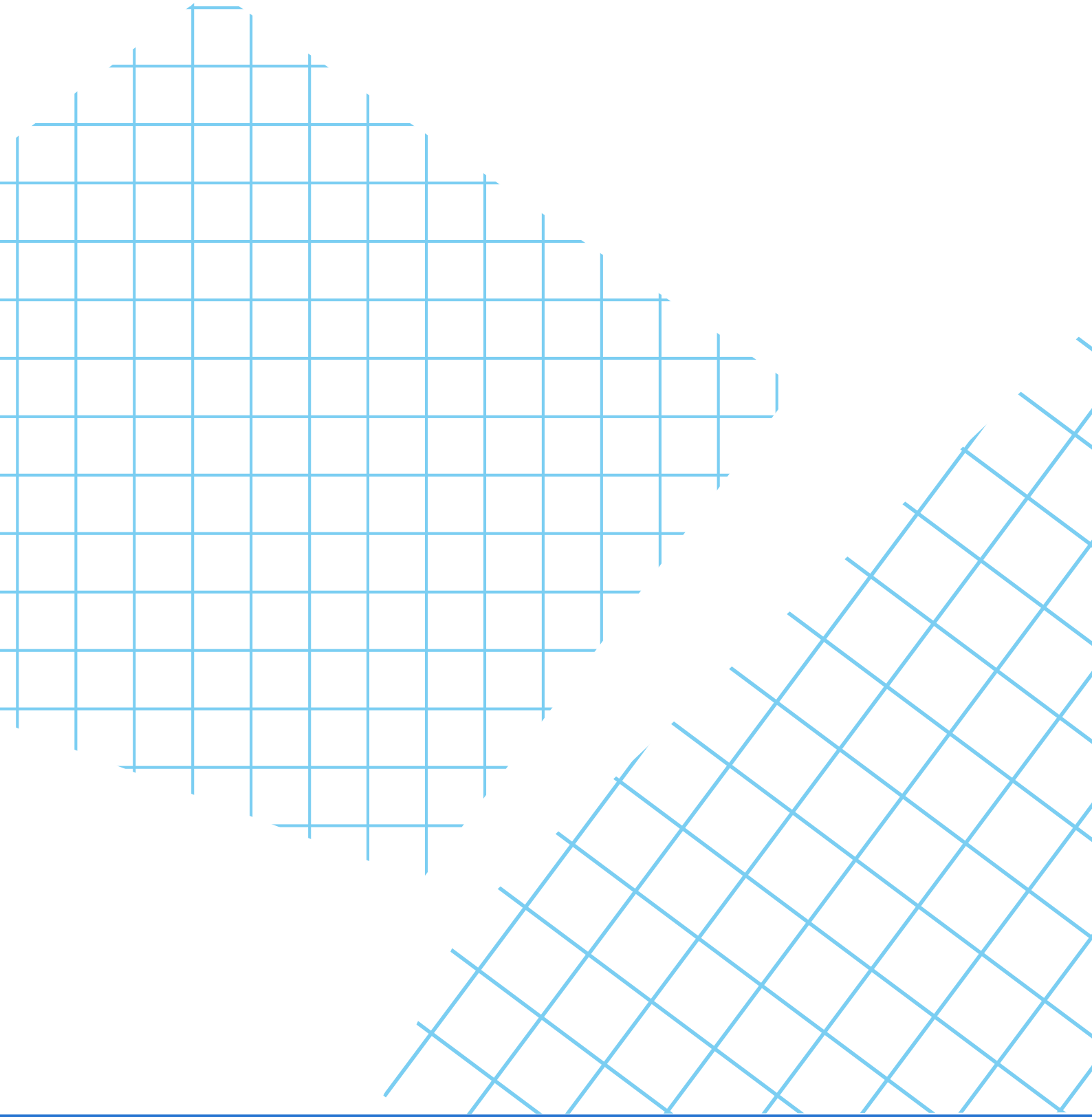
Commercial network vulnerability scanners,

such as the Acunetix, Burp Scanner, Qualys, Rapid7 InsightAppSec, and the Pentest-Tools.com Website Vulnerability Scanner, are proprietary tools that come with budget requirements.

These scanners are known for their ease of use, professional support, and continuous updates the vendor provides. They often feature a more user-friendly interface, advanced scanning options, and proprietary detection mechanisms.

Organizations typically choose commercial scanners because they are ready-to-use, reliable, regularly updated, offloading effort their teams need to use on more business-critical work.

Each type of scanner has its own set of advantages and considerations, and an AppSec engineer tasked with selecting a DAST (Dynamic Application Security Testing) tool will likely evaluate factors such as budget, technical expertise, specific organizational needs, and the complexity of the web applications to be scanned.



Selecting benchmark criteria and why they matter

When assessing a Dynamic Application Security Testing tool, an application security engineer cares about three things:

- **The breadth of application functionality the scanner can crawl**. A scanner can only look for vulnerabilities in an endpoint if it manages to crawl it.
- **The number and types of vulnerabilities the scanner is capable of finding**.
- **The level of trust in the accuracy of the vulnerabilities the scanner reports**. Any false positive reported by the scanner translates to cognitive overload.

For a benchmark to satisfy these web application scanning requirements, the deliberately vulnerable app used for testing must:

- Contain a wide variety of functionalities that cover most of what a real web application can do.
- Be built on tech stacks that accurately reflect the industry's current trends.
- Be transparent: knowing all the vulnerabilities that ought to be found.

This benchmark focuses on the scanners' vulnerability detection components and evaluates how the vulnerabilities each tool reported compare to the reality of the target's security posture.

Each scanner was evaluated based on its:

- **true positive rate**: how many of the vulnerabilities the scanner reported actually exist
- **false positive rate**: how many of the vulnerabilities the scanner reported are not actually there
- **false negative rate**: how many of the vulnerabilities that exist were not reported by the scanner.

To keep the evaluation impartial, the testbed needed not to be built by any of the companies behind the tools included in this benchmark. This ensured that no particular scanner had an advantage in the form of a testbed fine-tuned for their testing mechanisms.

With these requirements in mind, two targets were selected: Broken Crystals and DVWA (Damn Vulnerable Web Application).

Broken Crystals uses a wide range of technologies and it exposes a variety of vulnerabilities. What is more:

- it is built on a modern frontend framework, React, making it possible a challenge for crawlers which might result in fewer vulnerabilities found.
- it uses both a REST and a GraphQL API, with some vulnerabilities detectable only if the scanner knows how to work with these technologies.
- it contains a wide variety of vulnerabilities, from classic XSS, SQL injection, and the like, to modern vulnerabilities arising from flawed JWT and GraphQL implementations.

DVWA (Damn Vulnerable Web Application) was selected because it is an industry staple and because traditional applications (not single-page) continue to reflect a large part of the web. Additionally, given its notoriety, it was relevant to observe how close to 100% detection the scanners can get.

Benchmark methodology

Both Broken Crystals and DVWA were deployed to a VPS in the cloud. The setup consisted of a docker compose up or docker run, as both offer Dockerized versions.

Two vulnerable applications were scanned individually with each tool included in the benchmark. Once a scanner finished, the target web application was reset to its initial state to ensure the scanners didn't interfere with each other.

Important: If you are interested in verifying the results independently, please note that all scanners were updated to the latest plugins available as of February 2024.

Each scanner was manually configured to use their most comprehensive crawling strategy and to attempt to use all the vulnerability detections they have.

Where available, the REST API swagger files that defined the API were specified, as well as the GraphQL endpoint to be scanned.

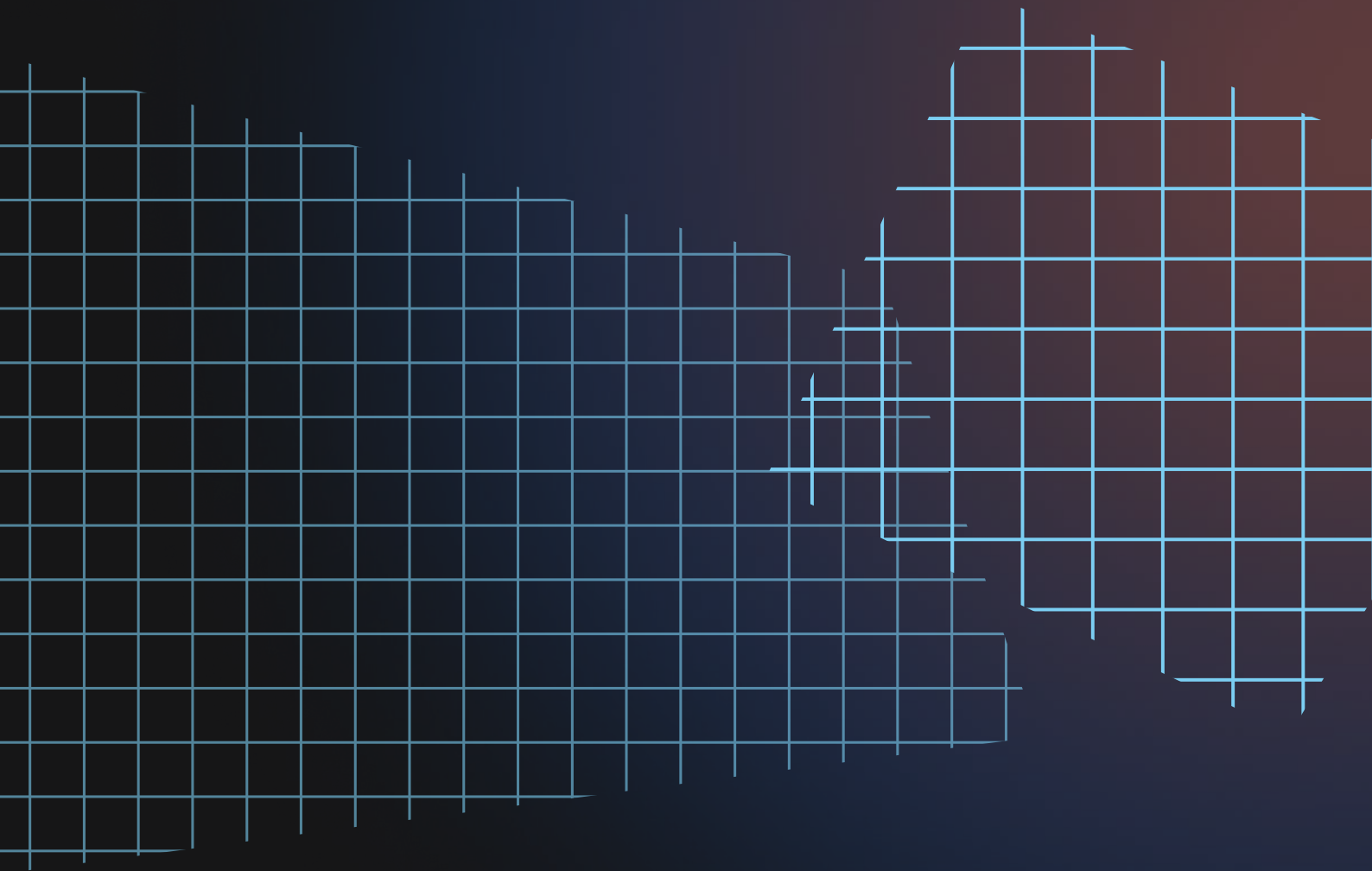
Some vulnerabilities were present in endpoints protected by authentication, so, where possible, each scanner was configured to run the scan as an authenticated user.

While using the default scan settings would have made it easier for independent reviewers to validate the findings in this benchmark, doing so would have excluded many vulnerability detection modules, making the results inaccurate.

Benchmark results

As stated in the overview, the analysis of the main web application vulnerability scanners on the market reveals a generally consistent level of vulnerability detection across both commercial and the notable open-source tool - ZAP - with minor variations. Below you can find more information on specific exceptions, such as Burp Suite's performance on the Damn Vulnerable Web Application (DVWA) and Acunetix on Broken Crystals.

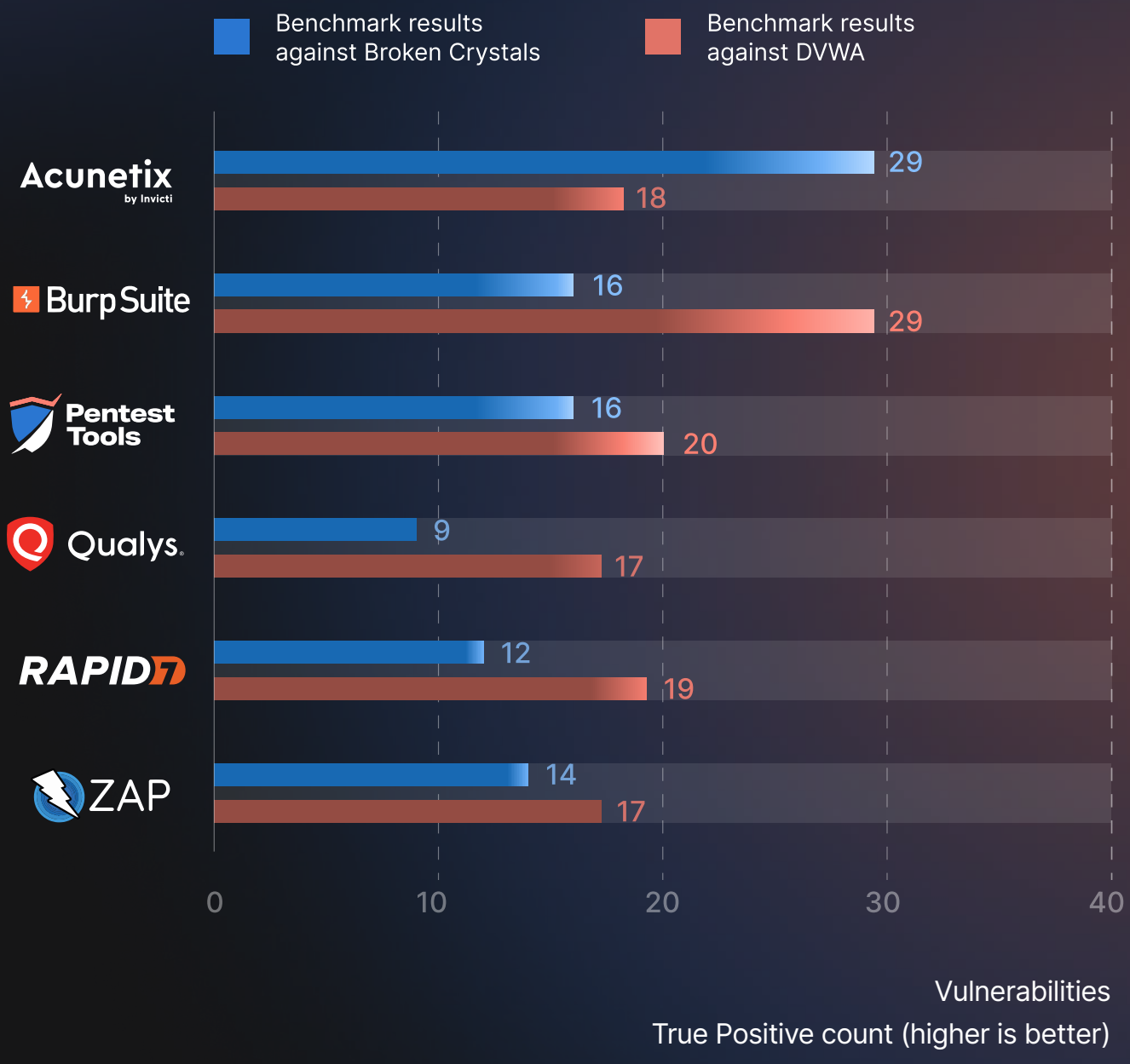
Regarding false positives, none of the scanners produced a significant number, suggesting a broadly effective accuracy. However, within DVWA scans, ZAP showed a surprising volume of inaccuracies, and some commercial tools, like Qualys and Rapid7 InsightAppSec, did too, but in much smaller numbers. In contrast, the Pentest-Tools.com Website Vulnerability Scanner was observed to maintain a much lower rate of false positives.



This comprehensive evaluation highlights areas of strength and opportunities for improvement in current web vulnerability scanning technologies.

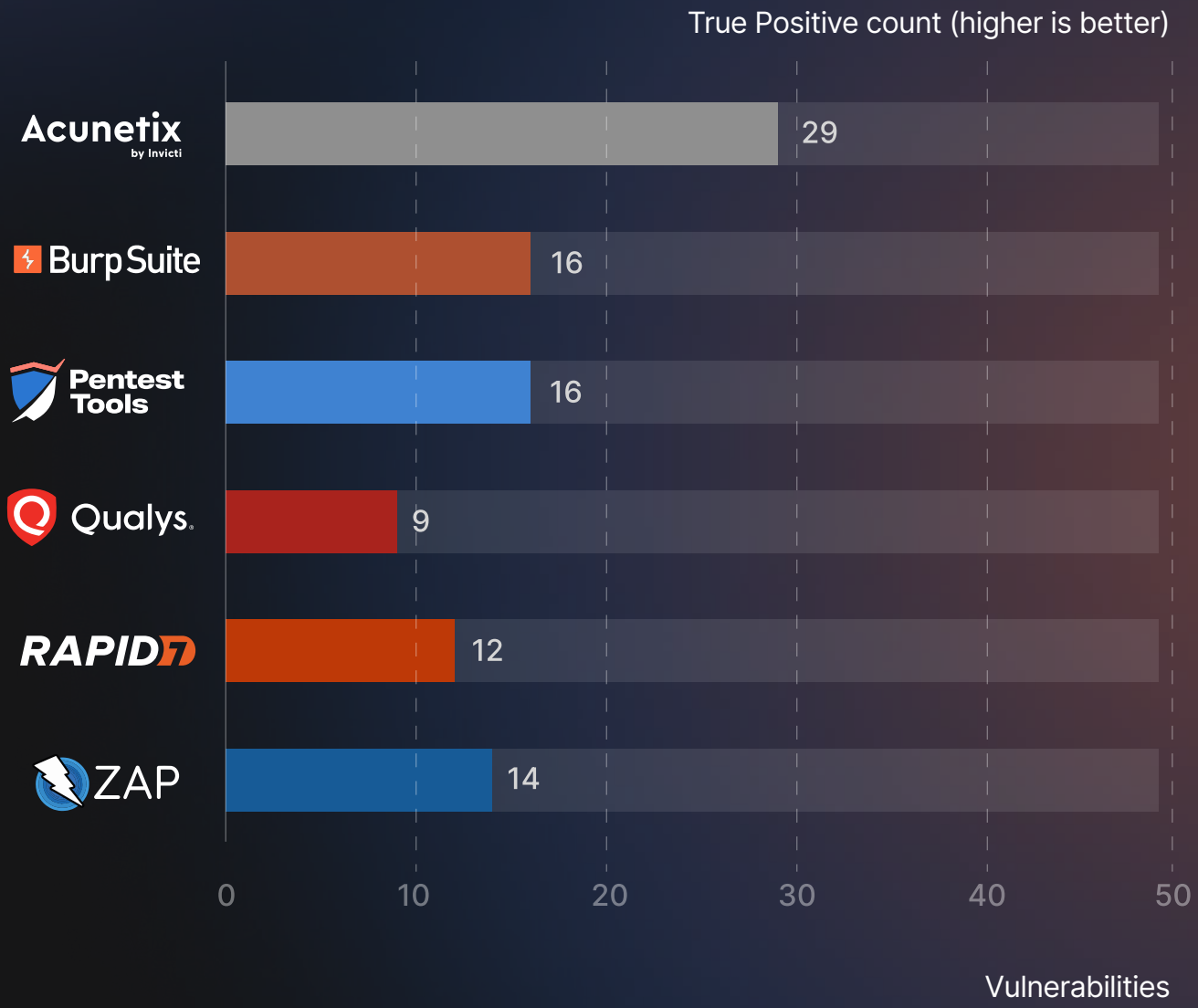
All the data behind the results in this benchmark are in this publicly available Google Sheet: Public Comparison - Web aplication vulnerability scanners benchmark data - 2024

Vulnerability detection across both targets

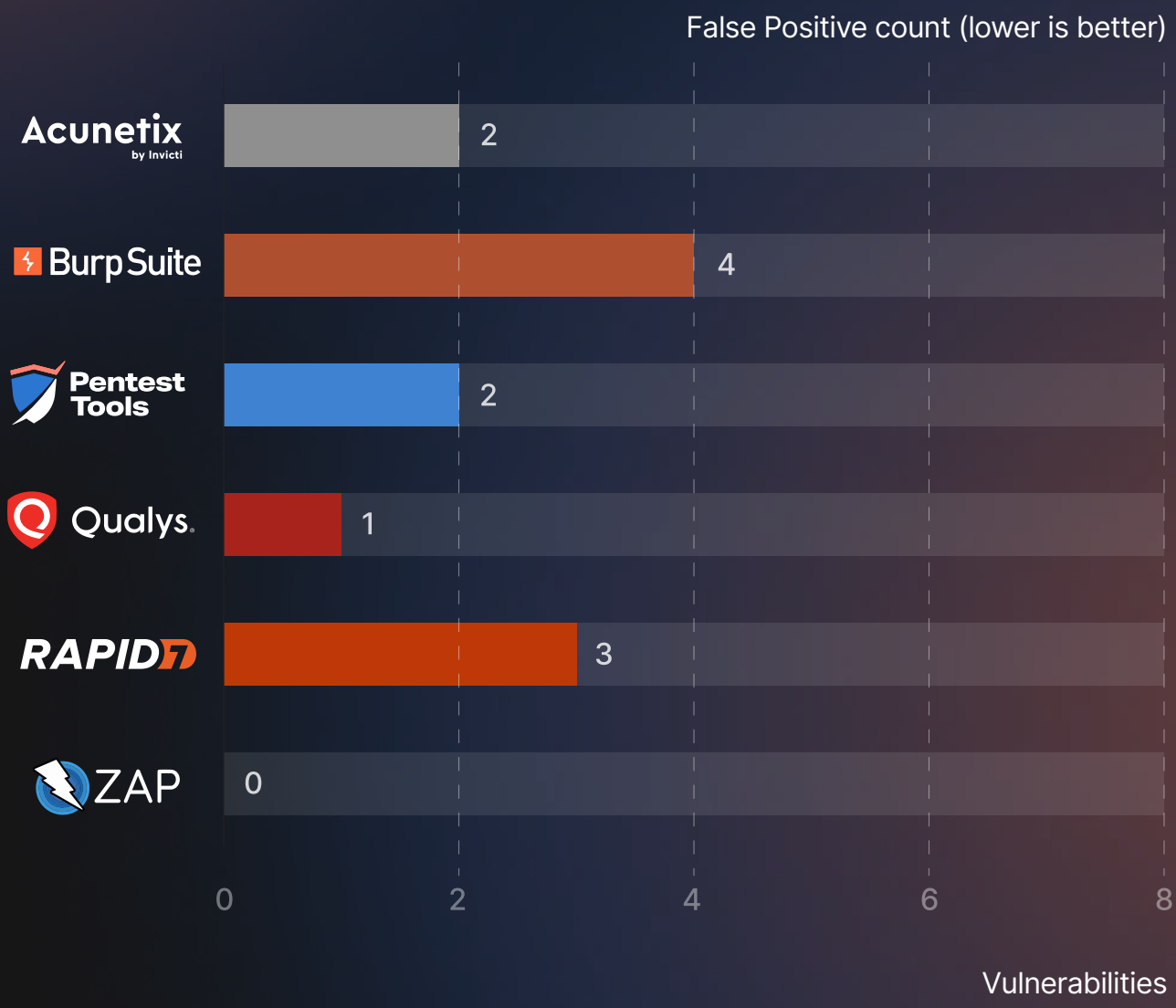


Benchmark results against Broken Crystals

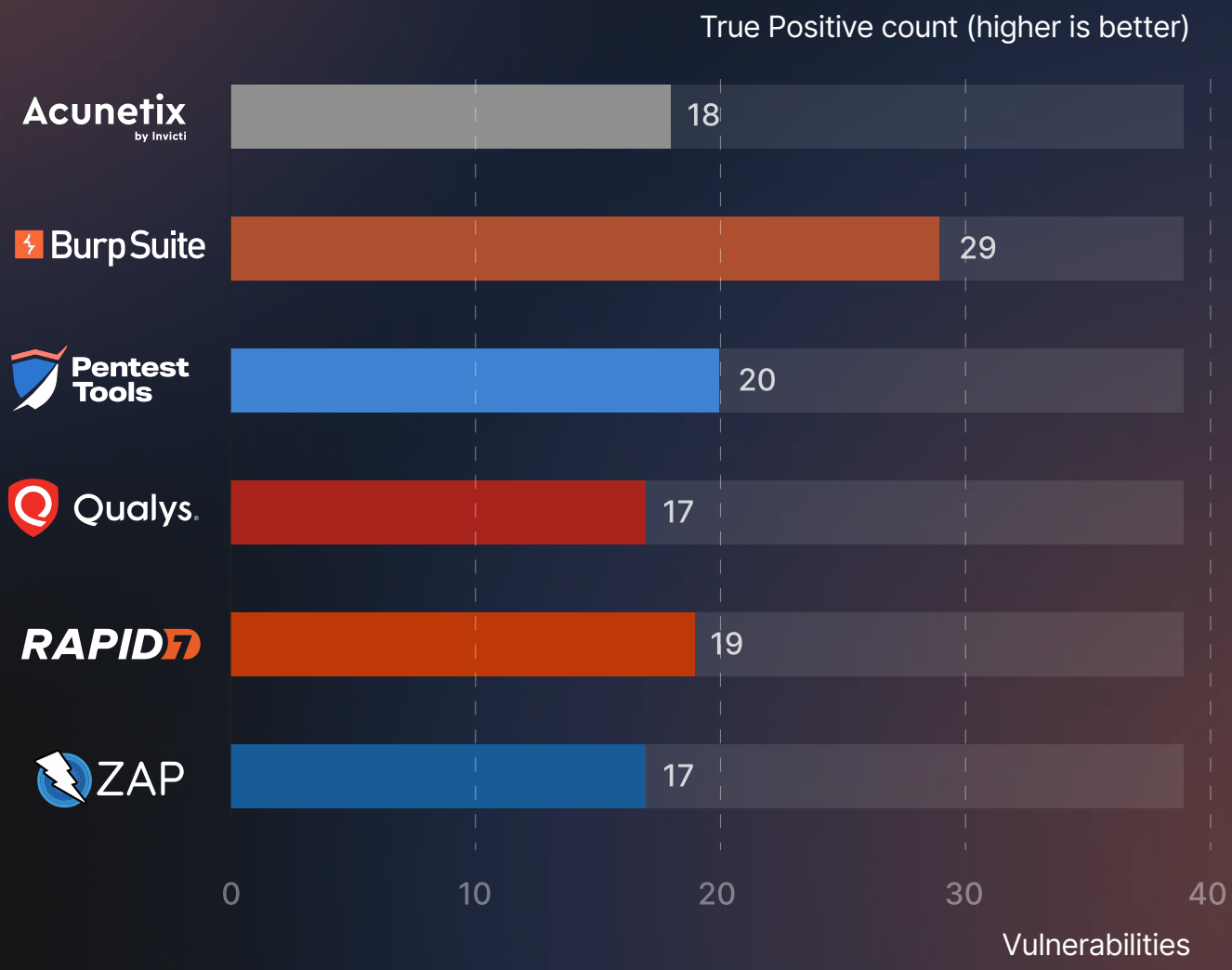
In the Broken Crystals assessment, Acunetix secured the leading position by a significant margin. The Pentest-Tools.com Website Vulnerability Scanner achieved second place, in a tie with Burp Suite. Notably, ZAP exceeded the performance of both Qualys and Rapid7 InsightAppSec, earning a commendable fourth place.



Regarding the detection of false positives, the performance was uniformly moderate across all scanners; each identified some false positives, though none distinguished themselves negatively. The minor variations observed were insufficient to decisively differentiate any of the scanners, indicating an overall even match among them.

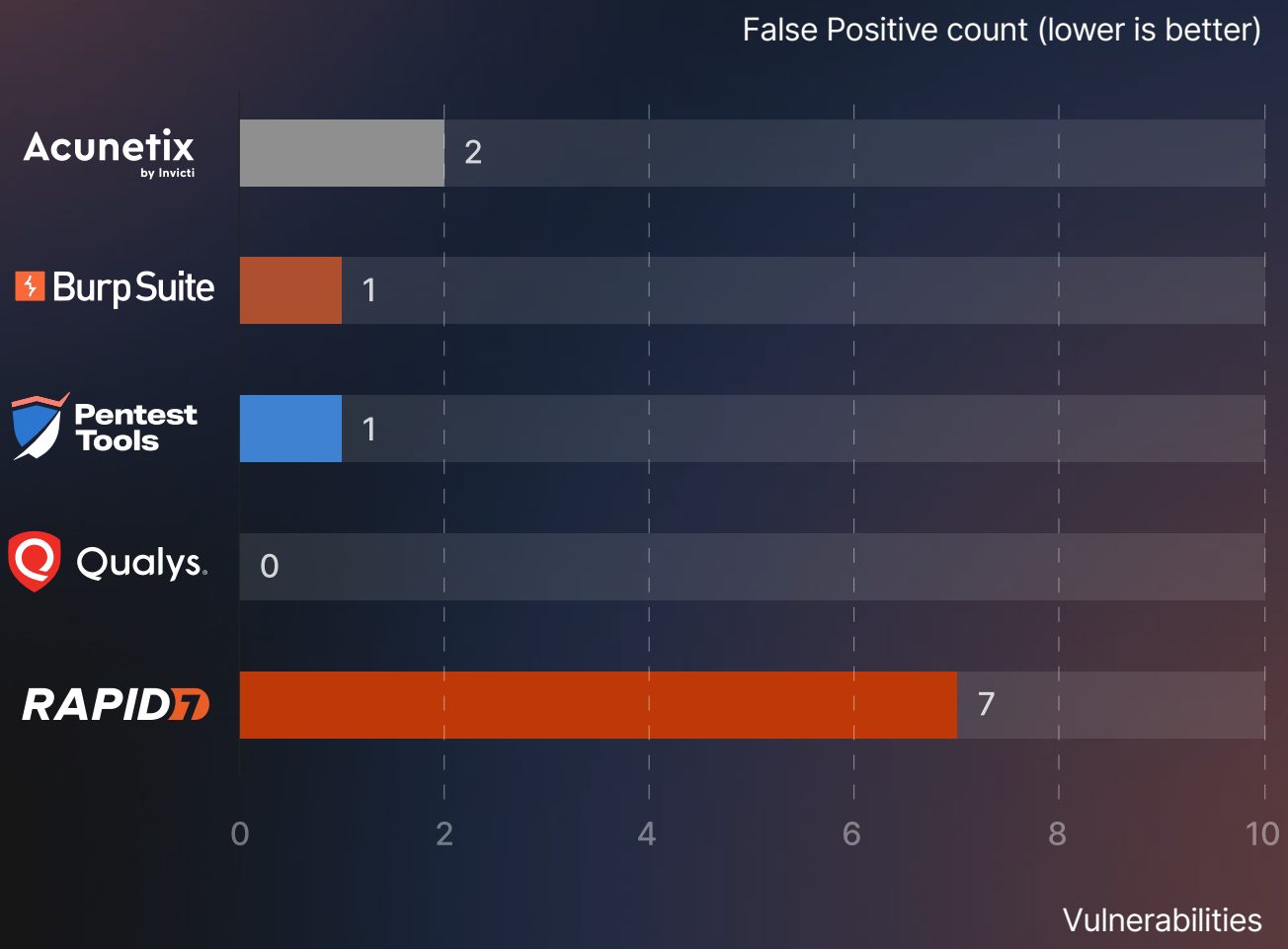


Benchmark results against DVWA



Burp Suite leads the field, identifying 29 out of 39 vulnerabilities, positioning it at the forefront. Following closely, the Pentest-Tools.com Website Vulnerability Scanner detected the second highest number of vulnerabilities. Rapid7 InsightAppSec and Acunetix are ranked third and fourth respectively, with Rapid7 identifying 19 vulnerabilities and Acunetix 18.

In the context of false positives, the situation presented significant differences in the Broken Crystals tests in some aspects.



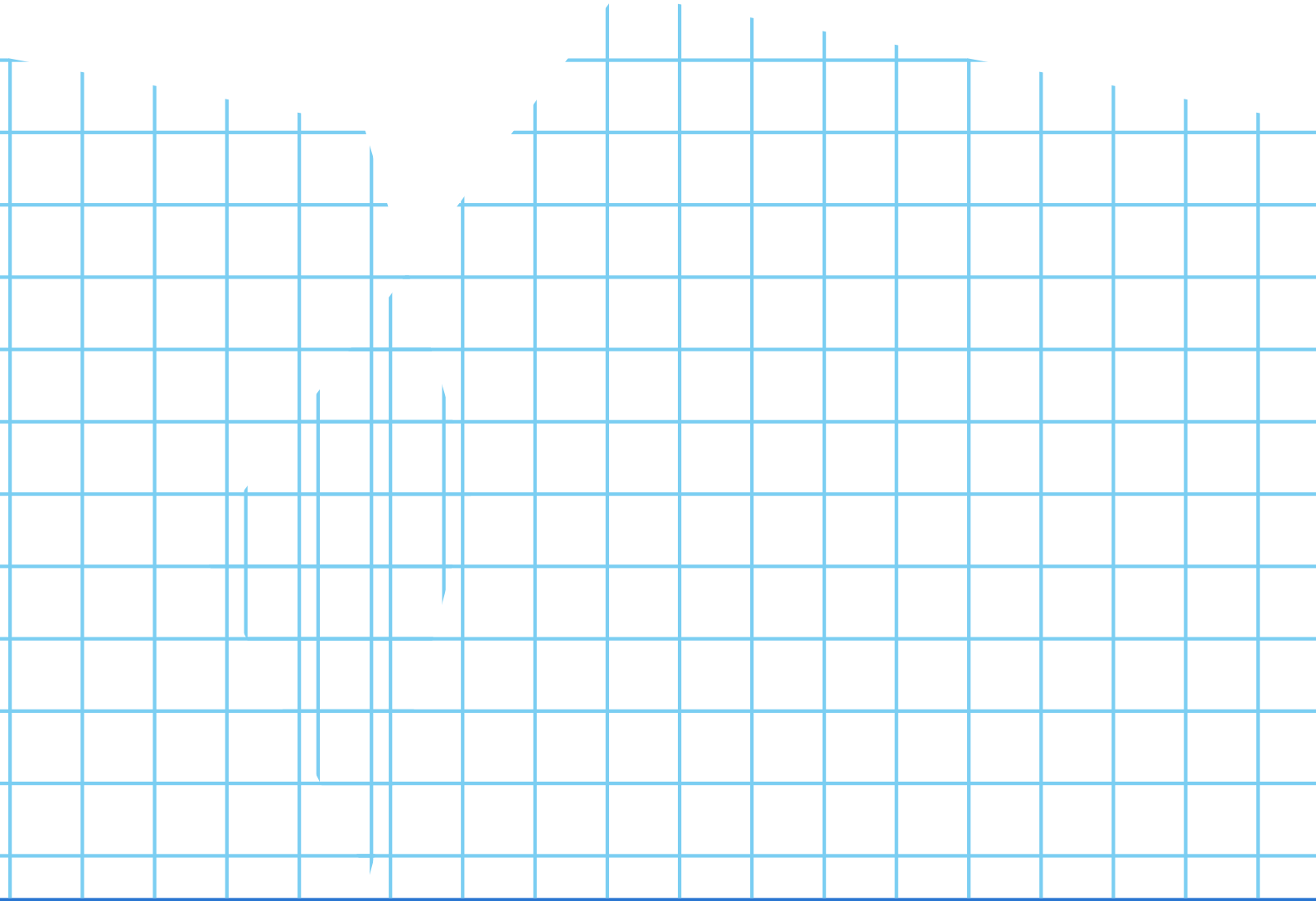
The astute observer will note the absence of ZAP from this comparison. This decision was made due to ZAP's reporting of false positives at a markedly higher rate than other scanners; specifically, it incorrectly identified 88 instances of SQL injections.

Beyond the benchmark: understanding scanner performance in complex real-world scenarios

When interpreting performance data, **it is crucial to acknowledge that exemplary results in benchmark testing do not always translate directly to real-world scenarios.**

The diverse implementation of similar features across web applications frequently results in corner-case scenarios and complex flows.

These nuances, which may not be accounted for in the benchmark's design, can affect the performance of scanners when deployed in actual environments.



Web app vulnerability scanners benchmark FAQs

1. **What were the criteria for identifying vulnerabilities in the benchmark testbeds??**

   In the benchmark testbeds, vulnerabilities are defined based on established standards, including OWASP Top 10 and the Common Weakness Enumeration (CWE). In addition, issues identified by scanners that serve as effective defense-in-depth measures, such as anomalies in the Content Security Policy (CSP) header, were also included.

   While this approach introduces a degree of subjectivity, it aligns with the practical responsibilities of security engineers who evaluate scanner reports during real-world assessments.

2. **What was the methodology for determining true and false positives in the benchmark testbeds?**

   The validation of each vulnerability reported by the scanners was conducted manually to ascertain its accuracy as a true positive. We encourage stakeholders to contact us should there be any concerns or discrepancies identified in our measurement of results, as we aim for the highest level of precision in our evaluations.

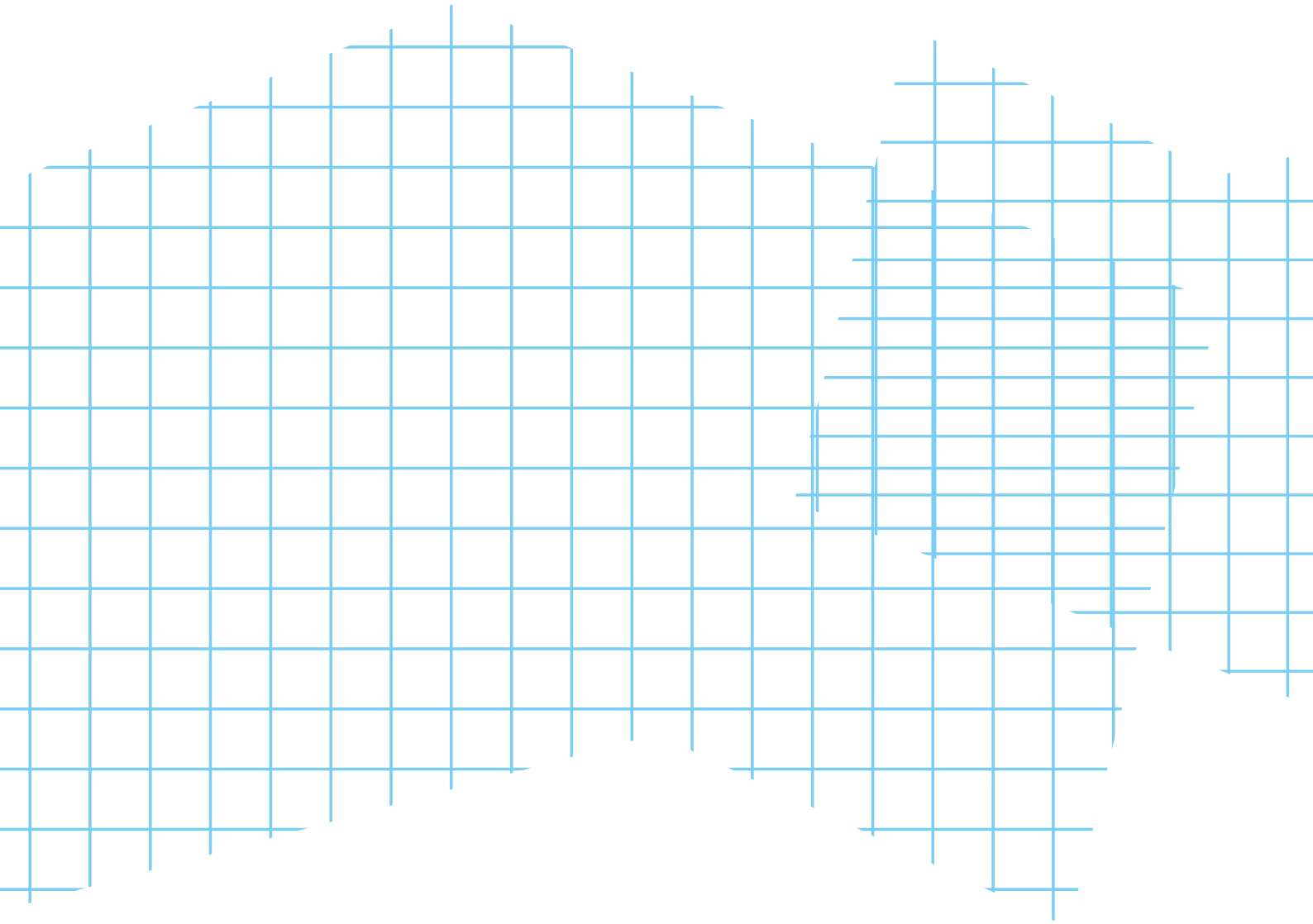
3. **Why does the benchmark not include other testbeds?**

   In choosing benchmark testbeds, the preference for open-source options was guided by the goal of making it easy for independent testers to replicate the setup

   A benchmark is only as valuable as its level of transparency and the lengths to which it allows for results verification within the cybersecurity community.

4. **The number of False Positives ZAP reported on DVWA seems suspicious. How was the scan configured?**

   The Active Scan policy was configured to a **Default Alert Threshold** of **Low** and **Default Attack Strength** to **Insane**. Additionally, injection was enabled in all the input vectors. At the time of writing, these were: URL Path, URL Query String (with the option to add parameters), POST Data, HTTP Headers, and Cookie Data.



Europe, Romania, Bucharest
48 Bvd. Iancu de Hunedoara

E: support@pentest-tools.com
pentest-tools.com

Join our community of ethical hackers!

Pentest Tools