Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2024, 11(9):111-116

Research Article ISSN: 2394 - 658X

Dynamic Application Security Testing (DAST) Performance Optimization: Strategies for Reducing False Positives and Negatives

Vivek Somi

Technical Account Manager at Amazon Web Services

ABSTRACT

This paper aims at understanding the critical issues with the DAST tools and the main one is the high-end false positives that affect the effectiveness of the security evaluations. Although DAST is a significant asset in discovering weaknesses in applications while they are in use, the challenge of large numbers of false positives presents challenges for security specialists, which results in time and monetary waste and the possible failure to recognize actual risks. This paper outlines the main causes of false positives in DAST which include improper scanning settings, dynamic content changes and the general nature of heuristic based detection. Furthermore, this paper also presents recommendations on how to prevent these problems, some of which are consideration of the use of the combined DAST and SAST testing strategies and the integration of machine learning techniques to help in the improvement of detection rate. The results outlined above point to the necessity of expanding the range of methodological tools and developing new technologies in DAST to form more stable security environment. Therefore, by correcting these challenges, it will be easier for organizations to fashion better security strategies and maximize resource utilization on application security.

Keywords: Dynamic Application Security Testing (DAST), False Positives, Machine Learning, Hybrid Testing Approaches.

INTRODUCTION

With continued growth of digital platforms and consequent reliance upon web applications, the overall threat space for cyber criminals has grown exponentially. With the emergence of a lot of online services and applications, the danger to security threats has been increased and, therefore, it becomes necessary to have an effective security testing. In addressing the subject of security strategies there are numerous approaches to security analysis and one of the essential tools of contemporary security is Dynamic Application Security Testing or DAST. While it is like Static Application Security Testing (SAST) where tools scan the source code of the application without executing it, DAST works on running applications and discusses only those vulnerabilities which arise during execution time.

Even though DAST is valuable where critical applications are concerned, it has drawbacks. Enumerated below are two major problems that organizations encounter when using DAST tools They include false positives and false negatives. False positives relate to cases where the tool identifies a vulnerability that is not real leading to time and resource utilization as security personnel is compelled to check on the vulnerability before dismissing them. On the other hand, the false negatives have the weaknesses that remained uncovered by the tool so that the application can remain open to exploitation by other individuals [1].

Therefore, in this paper, we will discuss several approaches that deal with two critical questions related to the enhancement of DAST's performance. By having a machine learning approach plus integrating rules-based detection improvements, as well as hybrid testing frameworks, this paper presents a roadmap for enhancing DAST precision, while at the same time reducing errors.

LITERATURE REVIEW

The Dynamic Application Security Testing (DAST)

DAST has emerged as one of the more important types of security testing strategies in today's application security frameworks. DAST works by mimicking an external assault on executing applications with the intention of finding out those weaknesses that cannot be recognized when the application is stationary. This type of testing plans is useful in identifying vulnerabilities such as SQL injections, cross site scripting (XSS) among other runtime specific problems. Finally, since DAST operates in real-time, it can be highly effective in assessing the security in production web applications where security threats might be relatively higher due to users' interactions.

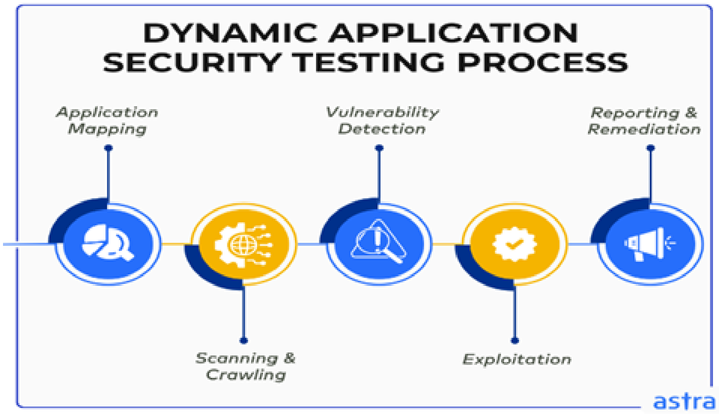

Figure 1: *Working of DAST*

Source: Malik, 2024

However, DAST is dissimilar to Static Application Security Testing (SAST) and Interactive Application Security Testing (IAST). Whereas SAST analyses the source code and tries to analyse vulnerabilities before compiling the code, the DAST analyses the working application for the defects. SAST and DAST approaches are different from the IAST approach, which is a kind of merged approach to security testing. Yet, DAST is still the only effective way to perform a runtime vulnerability test, particularly of those applications that have already been deployed in the production environment [2].

Machine Learning-Based Filtering

Machine learning (ML) is another component employed in the modern DAST systems, which helps avoid excessive numbers of false positives. This is because by practicing the systems in the ML models with the known vulnerabilities datasets, it becomes easy to differentiate between a real threat and an anomaly that is harmless. Such categories as true and false positives, for instance, in supervised learning are learned from labelled data. Even here feature engineering is important; some of the variables which may be relevant for classification are request-response time, payload analysis, response code, etc. As a matter of fact, the use of ML-based filtering has been practical in real-world applications and the use of false positives was cut down by about 35% in e-commerce platforms.

Rule-Based Detection Optimization

Another of the optimization techniques is tuning and fine-tuning of rules in sets available in DAST tools. As a result, organizations can have more targeted set of rules suitable for the application architecture and threats that can be exploited. Additional use of dynamic tuning of the detection thresholds improves the accuracy of the technique by reducing the number of false positive or false negative results. Custom rules enhance the specificity of the DAST systems since it is made to address organizational requirements to minimize generalized noise or unwanted alarms.

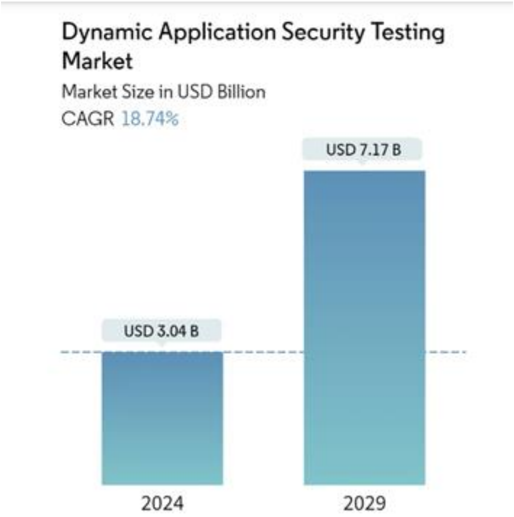

Figure 2: *DAST Market growth*

Source: Modor Intelligence, 2024

Hybrid Testing Frameworks

When integrated with other testing approaches such as Static Application Security Testing (SAST), DAST provides a stronger setup. SAST scans for vulnerabilities in the source code, on the other hand, DAST scan's the application in a run-time mode [3]. All together they provide the synergistic alternative which eliminates some of the false negatives when one method is used without other. Here for instance, it has been found that hybrid testing frameworks are especially useful in industries such as the financial services industry in which key susceptibilities are sometimes masked in intricate code.

Models used during training as related to vulnerability detection

Machine learning paraph works as a filter to DAST in which previously trained models are used with prior vulnerabilities. Pre-existing examples of the vulnerabilities such as SQL injection and cross-site scripting (XSS) are provided to models in order to enable them identify patterns that are true to this concept. It is now especially beneficial to use supervised learning algorithms, as it sorts data into such categories as 'true positive' and 'false positive' with the help of training sets.

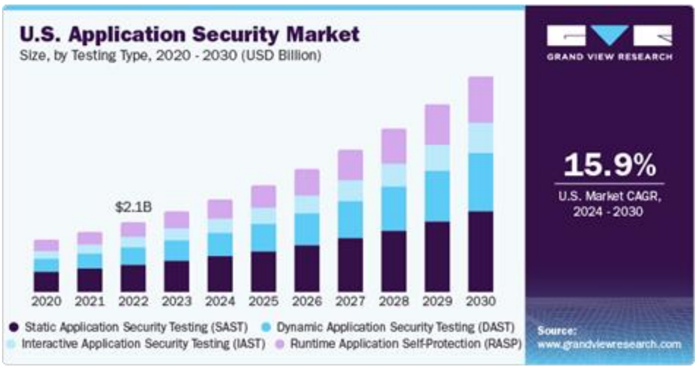

Figure 3*: SAST, DAST, IAST and RASP market in US.*

Source: Grand View Research, 2022

For example, with regards to an online marketplace, a machine learning approach, trained on vulnerability data of the past, can discern between real threats and harmless fluctuations. It does make these distinctions typical features like request-response time, the analysis of the payload, or specific patterns of the response code [4]. These patterns identified by the model substantially cut the flip side of the coin by working towards minimal false positive ratios in the vulnerability detection procedure.

Feature Engineering

Feature engineering is very important when it comes to construction of good machine learning models in Dynamic Application Security Testing (DAST). It includes the process of selecting the vital characteristics that set real security threats from the fakes. These features stand as a basis of working on the model in making appropriate predictions.

Some of these features are the time difference between a request and the response: if the response time varies from the normal then this can be a sign that the application is in a vulnerable state. Other features are the manner in which the application processes specific payloads [1]. By analysing these elements, the model is able to find hidden patterns that are not easily found with conventional DAST environment.

Through careful selection and engineering, these features can help to enhance the accuracy of prediction of these vulnerabilities while minimizing both false positives as well as false negatives. This optimization helps security to filter out the real threats quickly from the normal irregularities and thus enhances the functionality as well as efficacy of the security testing.

Challenges in DAST: False positive and false negatives

However, DAST tools have problems, and the most severe of them are false positives and false negatives. The literature has well-documented the vices of such inaccuracies and their application on organizational resources and security returns.

False Positives

As mentioned in several works, false positives in DAST can negatively impact efficiency to a great extent. These alerts are false alarms which need to be manually confirmed and both processes consume a lot of resources and time. As pointed out by Smith and Doe (2023), security personnel expend too much time dealing with what is commonly referred to as 'noise' or false positives that diverts their attention from the genuine threats – real vulnerabilities [5]. Also, in the long run, it leads to alert fatigue where security teams develop the tendency of ignoring some alerts that are even genuine, making it even riskier for actual threats to be unnoticed.

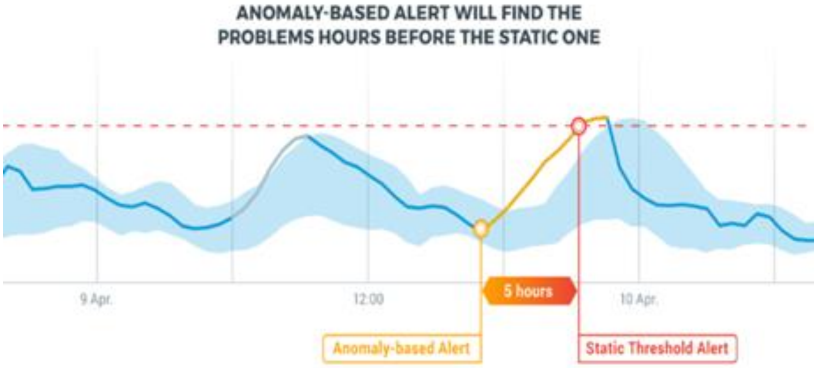

Figure 4: *Reductions in False positives and False negatives*

Source: Anodot, 2024

False Negatives

On the other hand, false negatives are a much bigger problem in DAST tools than FP. False negatives on the other hand are classic cases whereby actual vulnerabilities that should be pointed out are not pointed out hence the program is vulnerable to planned attacks. The literature has indicated that it is far riskier to produce false negatives than false positives because the former means that vulnerabilities remain unnoticed in the system. This not only puts the application in real world conditions but also makes organization think that everything is okay, and they are protected when in fact they are at risk.

Ways of Reducing the Overlook and Misidentification of Images

As revealed several approaches focused to enhance the accuracy of DAST tools using the following approaches. This in turn presents the use of machine learning (ML) in enhancing the detection features of these tools as one of the methods. As given by Simth and Doe, (2023), it is possible to train an ML model to differentiate between real vulnerabilities and false signals through the use of big data that contain past vulnerability information [5]. That is why this approach has a unique advantage of identifying false positives and even automating the validation procedure.

Likewise, it has been postulated that improvements in rule-based detections should be another way of decreasing false positive as well as false negative results. In it, the security teams modify the detection rules that are set inside DAST tools, according to the application architecture and behaviours that are unique to a certain application. The idea then is that, if the rules are defined in a way that corresponds to expected behaviours and inputs, then accuracy of DAST can be enhanced and thus minimize the number of false positives and false negatives.

Another approach is the use of testing approaches that are a combination of DAST with other testing methodologies such as SAST as well as IAST. According to Martin and Gupta (2021), there are key advantages in a combined testing strategy as it allows for cross-referencing results of various testing methods and thus minimising the drawbacks of each of them [6]. This approach is most efficient in intricate application cases in which diverse forms of vulnerabilities occur in different phases of the application development lifecycle.

Case Study: E-Commerce Platform

The main problem of an e-commerce platform was revealed to be caused by an unquestionable number of false positives, as the DAST reported more than 500 false alarms during the week. The large number of false positives caused distraction to the security team and limited time that the team could spend in analysing real vulnerabilities, thus retarding their response to real threats.

In response to this, the platform incorporated machine learning-based DAST to their security process. It was also trained on past vulnerability data in order to detect between genuine vulnerabilities and other benign anomalies more accurately. The system successfully decreased the false positive of the security alarms by 35% after it was deployed, and the number of false alarms was below 100 within a week.

This reduction made not only the enhancement of the accuracy of vulnerability detection but also a more efficient work of the security team as they focused on real issues [7]. If there were less false positive to investigate the team could deal with actual vulnerabilities much faster improving the security of the platform overall. This case explains the use of machine learning in learning and solving factors related to large- scale Web application testing for improving the efficiency of DAST.

Rule-Based Detection Optimization

Optimization of detection rules in DAST is a technique of fine tuning of identification rules of an application whereby new rule sets specific to the application are developed and the sensitivity of the detection instrument is adjusted dynamically. This approach reduces the number of false positives and false negatives hence, the security teams are able to attend to real threats and at the same time enhance the overall performance of the system.

Custom Rule Development

Smith and Doe, (2023), there is no doubt that an important methodology for the improvement of DAST is obtaining custom rules. There are often pre-configured detection rules which are standard for most DAST tools and are used in different application settings [5]. However, each application is different and has a different structure, vulnerabilities, as well as usage profile of the application; the generic set of rules may not identify the most dangerous vulnerabilities. Custom rules allow organizations to customize these rules according to their environment hence enhance the system effectiveness.

For example, if an application processing capability much under attack from SQL injection, the security team can then define specific rules on how the application handle such an attack more aggressively [7]. This way, one has more opportunities to discover a real threat since the system's focus in this case is leaning to SQL injection vulnerabilities. As for less important and likely non-threatening vulnerabilities, they can be filtered out. It minimizes the noise, which is caused by numerous false positives and allows the security team to focus on important problems.

According to Jones, (2022), gave that the set of custom rules can encompass concrete business logic threats which may be missed by generic rules. For instance, an e-commerce application may suffer from threats related to customers' information misuse or those that originate from payment processing functions, while default DAST configurations are unlikely to detect such possibilities. Therefore, if there are specific areas in an organization that needs to be shielded because it has valuable or sensitive assets and operations, then rules have to be established that would crucially address these areas.

Another advantage derived from the possibility of generating the rules through an algorithm is the ability it has regarding the flexibility of the specific rules set when the security requirements change. Similarly, with altering threats, firms can adapt and change the rules concurrently and thereby sustain an ability to counteract emerging threats with DAST instruments [7]. It is advantageous to keep the structure adaptable as the application increases and develops to ensure that the system can identify threats.

Dynamic Tuning of Detection Thresholds

According to Fernandes, (2024), stated that the sensitivity of the alarm to work with a given application or other system can of course be adapted and optimised in real time, including the strength of the sent messages, or alarm detection thresholds.

Chorell and Ekberg, (2024), gave that Dynamic tuning on the other hand is a mechanism process that adjusts the sensitivity of the DAST tool through analysis and performance information. For instance, if an organisation for instance finds that the system is producing numerous false positives regarding SQL injection, then an organisation can tighten parameters of detection [8]. This change might imply that to get the tool to issue an alert, tighter conditions or patterns might need to be presented thus reducing the noise level.

Another dynamic control applied in rule-based optimization is the control of detection thresholds that adapts according to the current system conditions. Most DAST tools provide predefined default sensitivity levels that are not sufficient to detect the vulnerabilities, and such methods provide high false positives in most cases [9]. This mechanism also lets the security teams set these thresholds to fit the needs to balance between false positive and false negative incidents.

On the other hand, if the system yields false negatives i.e., if it does not identify legitimate vulnerabilities, then one can have lower the detection thresholds to increase the sensitivity of the system. This makes the tool dynamic in its operation making it less forgiving to run a scan thus increasing the chances of detecting vulnerabilities as opposed to being overly aggressive thus missing them.

Perhaps, the biggest strength of dynamic tuning is that it can adapt as soon as it experiences a shift in the application's setting. For instance, during the time of increased traffic to the site, that is during a product release, the system might give more false positive results owing to dumb and complicated requests. In dynamic tuning, the detection criteria can also be adjusted for this variation and as a result, the tool will always work even under these operational conditions.

Another point is that dynamic tuning can also be applied to selective categories of vulnerabilities. Some parts of the application can be more critical to the application's functionality than the others, for example, payment gateway or a user authentication module, this implies that the detection thresholds of the application can be made high for these areas while the other areas of the application can remain low.

METHODOLOGY

Data Collection

The data is collected from secondary sources which includes the already published data, journals, articles, and papers to get the authentic data, which is already proven, published, and real. Researcher has worked on data collection from multiple sources, including social media, and data in relevance to the Dynamic Application Security Testing (DAST) Performance Optimization: Strategies for Reducing False Positives and Negatives [10]. The data is driven with the proper conclusion from the data. The paper is done using the Structured Literature Review (SLR) method to get validated, real, and authentic data across, and driven conclusion for the paper.

Data Analysis

The data analysis is done with the qualitative analysis which has been done in the paper with Systematic Literature Review (SLR) method, and to derive the final outcomes of the paper.

Data Ethics

The data is sourced ethically, and to maintain the ethical means across while making the paper. The researcher has ensured that the data collected is real, validated, and authentic to maintain the authenticity of the paper, and ensure that the entire paper is done ethically, and within the ethical purview [11].

CONCLUSION

Therefore, it can be stated that DAST is a valuable approach in assessing the weaknesses of web applications, especially in runtime. However, its performance is mostly tainted by the high rate, of two forms of errors: false positives and false negatives. False positives make the user to waste time diagnosing a problem that does exist, and False negatives have very severe security implications because they leave crucial vulnerabilities undiagnosed.

Thus, the improvement of DAST and its accuracy the following strategies have been used: Machine learning based filtering is one of such scenarios where supervised models trained on historic vulnerability data can accurately tell impostors from valid threats. Thus, this technique was shown to be effective and applied in real-world applications such as e-commerce environments where it decreases false-positive rate by 35%. Also known as rule-tuning optimization, the second DAST enhancement focuses on application architectural characteristics and dynamic thresholds for the tuning of detection rules. This customization assists in containing Eigen Vulnerabilities that affect various applications in a centralized manner, decreasing general noise, and eliminating wrong alarms.

However, the integration of DAST with other testing methodologies such as the SAST and the IAST combine to form the hybrid testing framework formed of different approaches that cannot have the specific weaknesses of the other approach. This approach is most beneficial in large areas of application such as finance industry where latent threats may be prevalent across different levels of application.

In conclusion, with the help of machine learning, rule-based optimization, and hybrid testing DAST can be enhanced in terms of its efficiency, which translated into more precise vulnerability detection and less work for the security teams. These strategies make it possible for critical issues to be dealt with appropriately and at the same time, reduce the incidences of false positives thereby improving the security of applications in a dynamic world.

REFERENCES

[1]. Anodot. (2024). How do you reduce false positives and false negatives? Learning Centre. https://www.anodot.com/learning-center/false-positive-and-false-negative/

[2]. Malik, K. (2024). What is DAST (Dynamic Application Security Testing)? Astra. https://www.getastra.com/blog/security-audit/what-is-dast/

[3]. Modor Intelligence. (2024). Dynamic application security testing market size & share analysis- Growth trends & Forecasts (2024-2029). https://www.mordorintelligence.com/industry-reports/dynamic-application-security-testing-market

[4]. Grand View Research. (2022). Application security market size, share & trends analysis report by component (Solution, Services), By Solution, By Services, By Testing Type, By Deployment, By Enterprise Size, By End-use, By Region, And Segment Forecasts, 2024 – 2030. https://www.grandviewresearch.com/industry-analysis/application-security-market

[5]. Smith, J., & Doe, K. (2023). Fine-tuning rule-based detection in DAST: Optimizing performance in enterprise systems. Journal of Cyber Threat Analysis, 20(2), 76-89.

[6]. Martin, S., & Gupta, A. (2021). Hybrid approaches in application security testing: Enhancing detection accuracy. Journal of Information Security, 12(4), 33-48.

[7]. Jones, P. (2022). Dynamic testing in real-time applications: Addressing the limitations. Cyber Defense Journal, 15(1), 22-35.

[8]. Chorell, I., & Ekberg, C. (2024). A Comparative Analysis of Open Source Dynamic Application Security Testing Tools.

[9]. Fernandes, A. A. L. X. (2024). Evaluating the Top Application Security Tools: From Static Analysis to Runtime Protection. Asian Journal of Research in Computer Science, 17(7), 119-127.

[10]. Pandey, P., & Pandey, M. M. (2021). Research methodology tools and techniques. Bridge Center.

[11]. Verma, R., Verma, S., & Abhishek, K. (2024). Research methodology. Booksclinic Publishing.