



**SecureIQlab**<sup>TM</sup>  
Bridging the enterprise cloud security gap

## **Cloud Web Application Firewall (WAF) CyberRisk Validation Report – AWS WAF - Web Application Firewall**

Product Version: AWS WAFV2

Published: Sep 30<sup>th</sup>, 2021

Language: English

## Report Contents:

<b>1. INTRODUCTION .....</b>	<b>2</b>
<b>2. TESTING PARAMETERS AND RESULTS .....</b>	<b>3</b>
1. SECURITY RESULT OVERVIEW .....	4
2. OWASP RATING .....	4
3. RESILIENCY RATING.....	5
4. BOTNET ATTACKS.....	6
5. LAYER 7 DoS ATTACKS.....	6
6. WEB APPLICATION PROTECTION.....	7
<b>3. OPERATIONAL EFFICIENCY .....</b>	<b>7</b>
7. DEPLOYMENT AND EASE OF INTEGRATION .....	8
8. MANAGEMENT COMPLEXITY .....	9
9. SCALABLE AND ELASTIC .....	9
10. LOGGING, MONITORING, AND AUDITING .....	10
11. ALLOWS GOOD TRANSACTIONS – FALSE POSITIVES .....	10
<b>4. APPENDIX .....</b>	<b>11</b>
1. CLOUD WAF TEST DEPLOYMENT .....	11
2. TEST EXECUTION .....	11
3. ATTACK TYPES .....	12
4. AWS WAF CONFIGURATION .....	12
5. AWS WAF RULES: .....	14
<b>5. CONTACT INFORMATION .....</b>	<b>16</b>
<b>6. COPYRIGHT AND DISCLAIMER .....</b>	<b>16</b>

## 1. INTRODUCTION

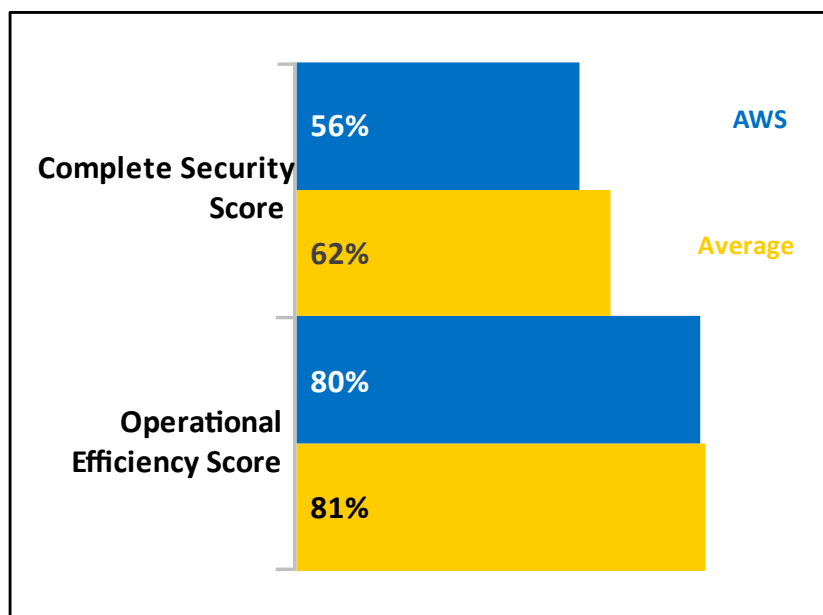


Figure 1. Overall Validation Results for AWS

The remote workforce has dissolved the network perimeter. This, along with cost savings and operational simplicity, is driving organizations to adopt cloud infrastructure. The explosive adoption of the cloud is not without challenges. There are risks associated with moving to the cloud. For example, web application-based vulnerabilities are among the top breach vectors<sup>1</sup>. Cloud-based web application firewalls (WAFs) are designed to mitigate this risk by protecting web applications without interrupting business operations in the cloud first world.

SecureIQLab has conducted a groundbreaking test of nine web application firewall (WAF) products to determine their security and operational efficiency. The test was conducted in accordance with the standards of the Anti-Malware Testing Standards Organization<sup>2</sup> (AMTSO). The test used version 1.0 of the SecureIQLab [Cloud WAF CyberRisk Validation Methodology](#) (AMTSO Test ID: AMTSO-LS1-TP039).

This report discusses the test results for the AWS WAF – Web Application Firewall version AWS WAFV2. To provide context, these individual results are presented alongside the collective averages for all nine of the tested products. This provides an at-a-glance comparative between the individual product under test and the collective results. One comparative report that highlights the performance of all nine vendors and individual reports for the remaining eight other tested WAF solutions are also available.

This CyberRisk Validation Report provides test results for the AWS WAF – Web Application Firewall version AWS WAFV2. Because thousands of attacks were simulated during the test, test results have necessarily been simplified and presented for review in a summary format for small and medium-sized businesses, enterprises, and managed service providers (MSPs). Figure 1 provides a summary of the product's overall validation results.

During the test, products were subjected to a battery of diverse attacks. Simple ecommerce applications and multiuser web applications were used as targets. Empirically validated data based upon industry guidelines and

<sup>1</sup> <https://www.verizon.com/business/resources/reports/dbir/2021/masters-guide/summary-of-findings/>

<sup>2</sup> <https://www.amtso.org/>

## Cloud Web Application Firewall CyberRisk Validation Report – AWS

regulations such as the OWASP Top 10<sup>3</sup> and PCI DSS<sup>4</sup> was obtained. It was obtained while securing targeted cloud applications on AWS with cloud WAFs.

SecureQLab is an IT security testing lab that was founded in 2019 and works with enterprises, governments, and security vendors to bridge the applied intelligence gap that exists between market and technology research. SecureQLab provides services to operationalize security and the metrics to help organizations improve their return on security investments.

The Anti-Malware Testing Standards Organization (AMTSO) is an international non-profit association that focuses on addressing the global need for improvement in the objectivity, quality and relevance of anti-malware testing methodologies. SecureQLab is a member of AMTSO.

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. “Through community-led open-source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.”<sup>5</sup> It publishes the OWASP Top 10 Report. SecureQLab has no affiliation with OWASP.

AWS was founded in 1994 and is a key vendor in the WAF market. AWS product was selected for inclusion in this test because it meets the SecureQLab WAF validation methodology selection criteria.<sup>6</sup>

## 2. TESTING PARAMETERS AND RESULTS

Cloud-based web application firewalls (WAFs) should accurately detect, prevent, and log attack attempts while remaining resistant to false positives. The aim of this section is to demonstrate the efficacy of the AWS WAF in this area.

Tests were performed utilizing black-box and gray-box testing. Black-box testing assumes that the internal code structure of the product being tested is unknown to the tester. For this testing approach, testers are not required to know a system’s implementation details. Gray-box testing assumes that part of the product’s internal code structure is known to the tester.

Default configurations and rule sets were used for the majority of the products in this test. However, any “Detect Only” mode settings that were part of default configurations were modified to “Block” mode, with default rulesets used as applicable.

Any required tuning was performed according to standard vendor recommendations available on the AWS website and according to relevant documentation available on AWS Marketplace to align with what an organization would experience during use of the product.

Tuning was based on industry and marketplace expectations that these solutions will require minimal to no tuning during provisioning, deployment, and management phases, which translates to lower operational expenses and increased revenue for the targeted audience, i.e., SMBs, managed service providers (MSPs), and managed security service providers (MSSPs). Tuning a WAF can be complex. Enterprises are advised to exercise due diligence during this process to avoid impacting normal browsing of the web applications or normal web application transactions.

Browsing the WAF-protected applications was performed using standard user transactions that included form submissions, comment writing, ecommerce transactions, and other transactions. See Appendix Section 5 for a snapshot of the ruleset utilized during this test.

More detailed information about our testing methods is contained in version 1.0 of the SecureQLab [Cloud WAF](#)

<sup>3</sup> Open Web Application Security Project®

<sup>4</sup> Payment Card Industry Security Standards Council

<sup>5</sup> <https://owasp.org/>

<sup>6</sup> Analyst and Enterprise Challengers - Small-mid-large enterprise security professional surveys, direct 1:1 inquiries and engagement with enterprises, organizations, MSP's, MSSP's, Gartner MQ, buyers guide, and Forrester Wave.

## 1. SECURITY RESULT OVERVIEW

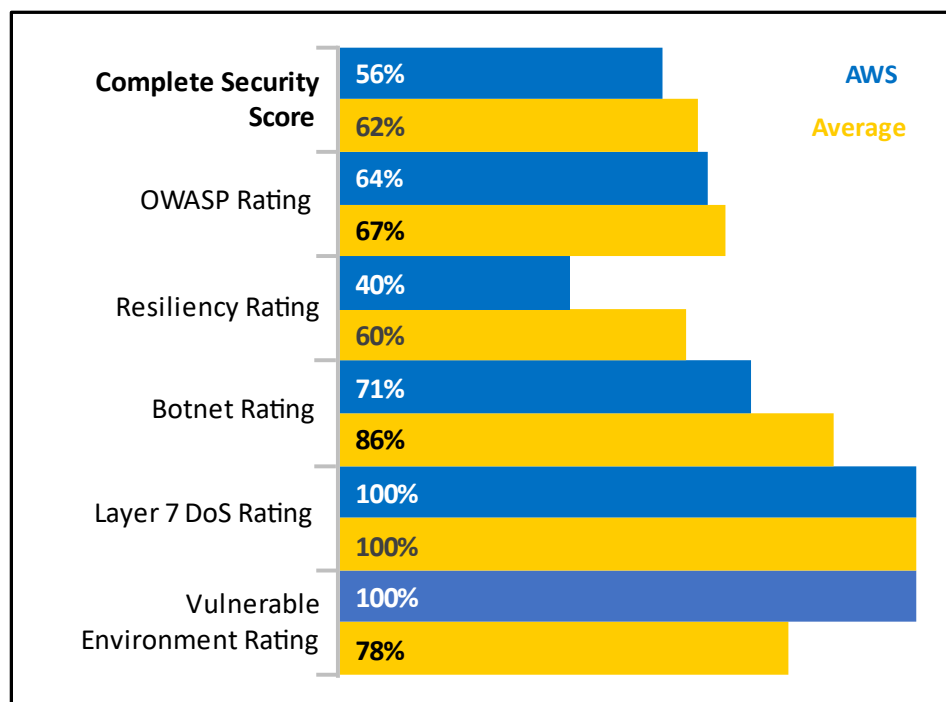


Figure 2. Security Validation Results for AWS

Figure 2 above provides an overview of the SecureIQLab findings during security validation of the AWS WAF. The *Complete Security Score* depicts the percentage of all attacks blocked by the WAF versus the total number of attacks tested. Equation 1 below depicts the *Complete Security Score* calculation, which is based on an unweighted percentage of all attacks blocked. The calculation method is unweighted to avoid the philosophical—and highly subjective—debate that invariably accompanies attack weighting. However, a necessary corollary to this is that threats that take more variations of simulated attacks to review will influence the Complete Security Score more than threats that can be evaluated with a lesser number of simulated attacks.

$$\text{Complete Security Score} = 100\% \times (\text{All Attacks Blocked}) / (\text{Total Attacks})$$

Equation 1. Calculation of Complete Security Score

Every cloud WAF evaluated in this test was subjected to twelve different categories of more than 100 real world-based operational scenarios targeting small-to-medium businesses and enterprises alike. A grand total of 22,465 attacks were used encompassing these scenarios and categories. The depth and scope of the testing performed by SecureIQLab is a first in the cybersecurity industry. SecureIQLab will continue to add attack libraries and other relevant operational metrics in future iterations of this test.

## 2. OWASP RATING<sup>7</sup>

The Open Web Application Security Project (OWASP) is a non-profit foundation dedicated to improving web application security<sup>8</sup>. The OWASP Top 10<sup>9</sup> Report is assembled by security experts from across the globe and

<sup>7</sup> Based on OWASP 2017 categories. Future test iterations are projected to use OWASP 2021 categories.

<sup>8</sup> SecureIQLab is not affiliated with OWASP.

<sup>9</sup> <https://owasp.org/www-project-top-ten/>

## Cloud Web Application Firewall CyberRisk Validation Report – AWS

describes the most critical web application vulnerabilities.

The AWS WAF was tested against five of these vulnerabilities. SecureQLab selected these vulnerabilities based on vulnerability prevalence and operational requirements.<sup>10</sup>

For detailed explanations of each of these attacks, please reference the OWASP Top 10. Table 1 below provides the results from these tests.


	OWASP Vulnerability Critical Rating				Test Results for AWS			
	Exploitability	Prevalence	Detectability	Technical	# Attacks	# Blocked Attacks	% Blocked	Test Average
<b>Injection Vulnerabilities</b>	3	2	3	3	-	-	26%	46%
SQL Injection					2225	666	30%	59%
Command Injection					2659	1069	40%	28%
CRLF Injection					78	2	3%	36%
HTML Injection					5	0	0%	18%
Host Header Injection					8	0	0%	56%
Server-Side Template Injection					140	11	8%	29%
LDAP Injection					45	45	100%	100%
Xpath Injection					16	4	25%	44%
<b>XML External Entities (XXE) Vulnerabilities</b>	2	2	3	3	-	-	91%	64%
Unrestricted File upload					4	4	100%	75%
XML External Entity Attacks					105	86	82%	52%
<b>Cross-Site Scripting XSS Vulnerabilities</b>	3	3	3	2	10209	9820	96%	86%
<b>Broken Access Control Vulnerabilities</b>	2	2	2	3	-	-	5%	43%
Path Traversal					6951	760	11%	42%
Brute Force					1	0	0%	44%
<b>Sufficient Logging &amp; Monitoring</b>	2	3	1	2	NA	NA	100%	89%
<b>OWASP Rating</b>							<b>64%</b>	<b>67%</b>

Table 1. OWASP Vulnerability Validation

Category averages are determined by equally weighting the test case averages within each category. As an example, Equation 2 below provides the formula for calculating the average for the *Broken Access Control Vulnerabilities* category.

$$\text{Broken Access Control Vulnerabilities} = [\text{Directory Traversal} + \text{Brute Force}] / 2$$

Equation 2. Formula for Calculating Average for Broken Access Control Vulnerabilities OWASP Category

In addition to security efficacy, the product's logging and monitoring capabilities were reviewed. More detailed analysis of these capabilities may be found in Section 3 under *Logging, Monitoring, and Auditing*.

### 3. RESILIENCY RATING

Security products must demonstrate resiliency. Failure to do so will have significant consequences. The prevailing definition of operational resilience is provided by the Department of Defense (DoD), and states it is: "The ability of systems to resist, absorb, and recover from or adapt to an adverse occurrence during operation that may cause harm, destruction, or loss of ability to perform mission-related functions."<sup>11</sup>

To test its operational resilience, the AWS WAF was tested against several attack masking techniques to determine whether it could successfully block attacks that would otherwise go unseen. A higher resiliency rating indicates a product is more capable of withstanding and absorbing different variations of attacks while a lower resiliency rating indicates a product is less likely to detect different variations of attacks.

Five test cases were employed to test the resiliency of the AWS WAF – Web Application Firewall version AWS WAFV2. Table 2 below provides the test cases and the product's results. The *Resiliency Rating* is calculated by

<sup>10</sup> Testing a product against all 10 categories may yield different overall results.

<sup>11</sup> [https://www.esd.whs.mil/portals/54/documents/dd/issuances/dodi/850001\\_2014.pdf#page=57](https://www.esd.whs.mil/portals/54/documents/dd/issuances/dodi/850001_2014.pdf#page=57)

averaging the results of the five test cases.


 <b>Resiliency</b>	Test Results for AWS	
	Blocked/Bypassed	Test Average
Web Shell Attack	100%	67%
Custom Web Shell Attack	0%	33%
Out-of-Band Data Exfiltration	100%	67%
SQL Injection WAF Ruleset Evasion	0%	67%
Command Injection WAF Ruleset Evasion	0%	67%
<b>Resiliency Rating</b>	<b>40%</b>	<b>60%</b>

Table 2. Resiliency Validation Results

#### 4. BOTNET ATTACKS

A botnet is a network of compromised computers that is used by a remote administrator to carry out automated attacks. AWS WAF – Web Application Firewall version AWS WAFV2 was tested against seven types of bot attack. These attacks were initiated from Asian and North-American locations to determine whether the geolocation of an attack source impacts the product's security effectiveness. Results show that geolocation does not impact the product's security effectiveness (see Table 3 below). The *Botnet Rating* is calculated by averaging the seven contributing scores.


 <b>Attacks</b>	Test Results for AWS	
	Blocked/Bypassed	Test Average
Credential Stuffing Attack	0%	56%
Tor-Based Layer 7 Attack	100%	100%
Web Content Scraping	100%	89%
Form Submission Abuse Attack	0%	56%
Website Crawler	100%	100%
Broken Link Checker	100%	100%
Bot User Agent Attack	100%	100%
<b>Botnet Rating</b>	<b>71%</b>	<b>86%</b>

Table 3. Botnet Attack Results

#### 5. LAYER 7 DOS ATTACKS

Layer 7 denial-of-service (DoS) attacks are more difficult to detect than other DoS layer attacks because they use a valid TCP connection. Test cases for such attacks are also more infrastructure friendly than distributed denial-of-service (DDoS) attacks and thus avoid the issues that DDoS attacks may trigger with cloud service providers. Below, Table 4 presents the results of testing the AWS WAF against five Layer 7 DoS attacks. The product's *Layer 7 DoS Rating* was determined by taking the average of its scores against the five attacks


 <b>Layer 7 DoS</b>	Test Results for AWS	
	Blocked/Bypassed	Test Average
HTTP Flood Attack	100%	100%
Asymmetric Attacks	100%	100%
Repeated Single Attacks	100%	100%
Application-Exploit Attacks DoS	100%	100%
Slowloris DDoS Attack	100%	100%
<b>Layer 7 DoS Rating</b>	<b>100%</b>	<b>100%</b>

Table 4. Layer 7 DoS Validation



## 6. WEB APPLICATION PROTECTION

A cloud-based WAF should protect vulnerable web applications. While the proliferation of web application frameworks has made deployment and maintenance of web applications simpler, it has also made it challenging to update these frameworks without affecting the functionality of the web applications. Businesses, and MSSPs by extension, can ill afford downtime and may delay updates to avoid breaking applications. Consequently, WAFs can remain vulnerable to various published vulnerabilities and exploits, which makes it easier for cybercriminals and script kiddies to compromise applications.

AWS WAF – Web Application Firewall version AWS WAFV2 was tested against non-zero-day exploits and successfully blocked all attacks. See Table 5 below for the full results. The *Vulnerable Web Environment Rating* is calculated by taking the average of the three vulnerability scores.


 <b>Vulnerable Web Environment</b>	Test Results for AWS	
	Blocked/Bypassed	Test Average
Vulnerable Wordpress Installation	100%	78%
Vulnerable Joomla Installation	100%	78%
Vulnerable Drupal Installation	100%	78%
<b>Vul. Web Env. Rating</b>	<b>100%</b>	<b>78%</b>

Table 5. Vulnerable Web Environment Results

## 3. OPERATIONAL EFFICIENCY

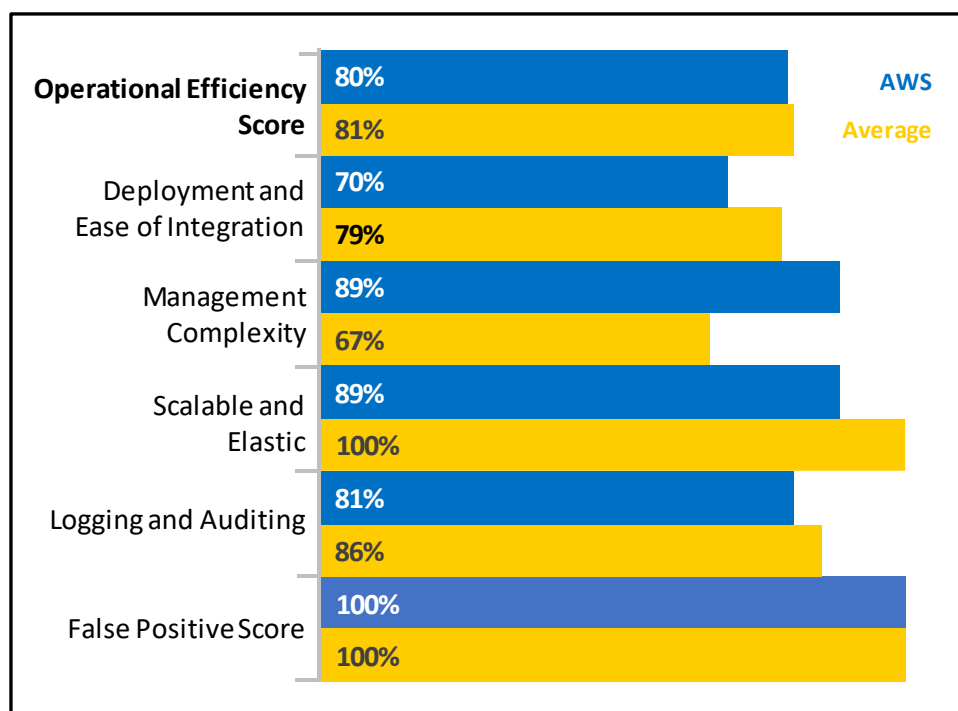


Figure 3. Validation of Operational Efficiency for AWS

Cloud-based WAF technology allows for the creation of customized security, which benefits organizations in the following ways:

- Ease of deployment and integration
- Less complex to manage



## Cloud Web Application Firewall CyberRisk Validation Report – AWS

- Scalable and elastic
- Monitoring, logging, and control capabilities
- Allows business-related transactions

AWS WAF – Web Application Firewall version AWS WAFV2 was validated in each of these areas of operational efficiency. Figure 3, above, displays the product’s high-level results.

Category scores were calculated by aggregating earned points and then dividing this number by the total possible score to find a percentage. Points (integers 0 – 3) are earned for each feature within a category. Results highlighted in green are worth three points; results highlighted in yellow are worth two points; results highlighted in orange are worth one point; and results highlighted in red are worth zero points.

The *Operational Efficiency Score* was calculated by adding together the total points for each category, then dividing this number by the maximum potential points (84) and multiplying that number by 100%. Below, Equation 3 states the *Operational Efficiency Score* calculation.

$$\text{Operational Efficiency Score} = \frac{\left( \begin{array}{l} \text{Deployment and Ease} \\ \text{of Integration Points} \end{array} + \begin{array}{l} \text{Management} \\ \text{Complexity Points} \end{array} + \begin{array}{l} \text{Scalable and} \\ \text{Elastic Points} \end{array} + \begin{array}{l} \text{Logging and} \\ \text{Auditing Points} \end{array} + \begin{array}{l} \text{False Positive} \\ \text{Points} \end{array} \right)}{84 \text{ Points}} \times 100\%$$

Equation 3. Operational Efficiency Score Calculation

Test average results are determined by either calculating the mean results or taking the mode from the vendor group results where relevant. Mean results are taken when the results are quantitative, e.g., *Time to Deploy*, *# of Steps for Setting up WAF service* or *# Audit Trail Fields*. The mode is used in the group average results when the results are qualitative in nature, e.g., *Complexity of Tuning WAF*, *Auto-Scaling Capability* or *Log Configuration Complexity*.

### 7. DEPLOYMENT AND EASE OF INTEGRATION

Cloud WAFs typically take less than an hour to a few days to set up and seldom require installation of software or hardware.

Cloud WAFs should integrate with other security tools, Security Information and Event Management (SIEM) systems, and data repositories. This typically allows both tasks and alerts to be automated. Typical integrations include DevOps tools like Slack and Jira and can include automated additions to policies as threats are detected.

SecureQLab validated the ease of deployment and integration for AWS’s SaaS WAF, AWS WAF – Web Application Firewall version AWS WAFV2. Overall, it was a simple deployment. AWS WAF can be integrated with ELB, an API gateway, and a CDN. A strong understanding of AWS resources is a prerequisite for some non-default settings. Findings for deployment and ease of integration are in Table 6 below.


 <b>Deployment and Ease of Integration</b>	Validation Results	
	AWS Results	Test Average
Time to Deploy	15-20 minutes	29 minutes
# of Steps for Setting Up WAF Service	7 steps	3 steps
# of Steps for Requesting a Public Certificate	7 steps	2 steps
# of Steps for Requesting a Private Certificate	7 steps	3 steps
# of Steps to Add an SSL Certificate to an Existing Site	4 steps	3 steps
Application Load Balancing and Monitoring	Yes	Yes
Integration with Multi-Cloud WAF	NA	NA
Plug and Play Integration with On-prem Firewall	NA	NA
Plug and Play Integration for SIEM/S3 Bucket	Yes	Yes
Plug and Play Integration for API Gateway	NA	Yes
Required Contacting Support During Deployment	No	No

Table 6. Deployment and Ease of Integration Findings

The *Deployment and Ease of Integration Score* is the percentage of the AWS 23 points earned to the possible 33 points, or 70%. The group score average for *Deployment and Ease of Integration* was 79%.

## 8. MANAGEMENT COMPLEXITY

Cloud WAFs market the promise that they are more easily managed than on-premises solutions. They are less complex and more easily managed because they receive automated updates, automatically deal with spikes in traffic (as discussed below) and work out of the box.

Cloud WAFs are less complex because the cloud WAF provider manages the security details and automatically allocates the resources needed. The cloud WAF provider typically has already tuned the security software. Users typically have a single management console to monitor.

Users can typically create additional policies that allow identity-based access and network segmentation. Because the cloud WAF provider automatically applies policy modifications wherever the cloud WAF is deployed, policy modification is generally simple and rapid. Results from SecureQLab's experience while managing AWS's WAF are in Table 7, below.


 <b>Management Complexity</b>	Validation Results	
	AWS Results	Test Average
Complexity of Tuning WAF	Low	Low
Complexity of Setting Security Policy	High	Low
Complexity for Setting Up WAF	Low	Low
Managing WAF Updates	Automatic	Automatic
Complexity of Managing Web ACL	Low	NA
Internal WAF Migration Complexity	Low	NA

Table 7. Management Complexity Experience

The *Management and Complexity Score* is the percentage of earned points for AWS, 16, to the total complexity points possible, 18 points or 89%. The group score average for *Management Complexity* was 67%.

## 9. SCALABLE AND ELASTIC

Cloud WAFs should help the customer avoid sizing issues planning by providing for automated flexible scaling. Scaling typically occurs in response to changing traffic load patterns. Providers typically allow customers to choose options that balance optimizing performance and optimizing costs. Table 8 highlights SecureQLab's findings in this area. The *Scalable and Elastic Score* is the percentage of the earned 8 points to the total possible 9 points or 89%. The group score average for *Scalable and Elastic* was 100%.


 <b>Scalable and Elastic</b>	Test Results	
	AWS Results	Test Average
Auto-Scaling Capability	Yes	Yes
Manual Scaling Capability	Yes	Yes
Load Balancing and Failover	Not Default	Yes

Table 8. Scalable and Elastic Validation

## 10. LOGGING, MONITORING, AND AUDITING

Sufficient logging capabilities are required for incident response, auditing, and many compliance and regulatory purposes. Cloud WAFs need to provide enough visibility into web traffic and sufficient control capabilities for security teams to spot issues and resolve them. Additionally, Cloud WAFs need to have a means to integrate logged data with other storage devices for redundancy. Below, Table 9 covers our logging, compliance and auditing findings.

Our researchers found that setting up logs was a simple process of clicking through 29 steps. Additionally, AWS uses the JSON format for logs. Lastly, there are at least six default dashboards to choose from and creating custom rules only takes four steps.


 <b>Logging and Auditing</b>	Validation Results	
	AWS Results	Test Average
Log Configuration Complexity	Low	Low
Third Party Log Storage Facility	Yes	Yes
Web Request Inspection	Yes	Yes
Multi-Factor Authentication	Yes	Yes
Application Monitoring	Needs add-on	NA
Infrastructure Monitoring	Needs add-on	Yes
# Audit Trail Fields	29 fields	7 fields

Table 9. Log and Audit details

The *Logging and Auditing Score* is the percentage of the 17 points earned by AWS to the total 21 total possible points for this section or 81%. The group score average for *Logging and Auditing* was 86%.

## 11. ALLOWS GOOD TRANSACTIONS – FALSE POSITIVES

WAFs need to allow for business-related transactions while blocking malicious activity. The false positive rate is important because false positives prevent the operation of the business. Policies need to be adjusted to minimize false positives.

False Positives increase noise for already stretched thin security teams and contribute to alert fatigue. Properly tuned security devices will not improperly detect benign traffic as malicious. Four different false positive test cases were used to validate that the WAF under test would not block simulated consumer purchases. These test cases simulated users that would browse the web application normally while being protected by the cloud WAF. Given the importance of WAFs not interfering with ecommerce, all four test cases are required to pass through the product under test to receive a passing score. The results for the false positive testing are found below in Table 10. The *False Positive Score* is the percentage of the 3 points earned by AWS to the total possible 3 points. The higher the *False Positive Score*, the lesser the operational overhead in tuning the WAF.


 <b>False Positives</b>	Test Results	
	AWS False Positive Results	Group False Positive Results
False Positive Tests	100%	100%
<b>False Positive Score</b>	<b>100%</b>	<b>100%</b>

Table 10. False Positive Testing Results

## 4. APPENDIX

### 1. CLOUD WAF TEST DEPLOYMENT

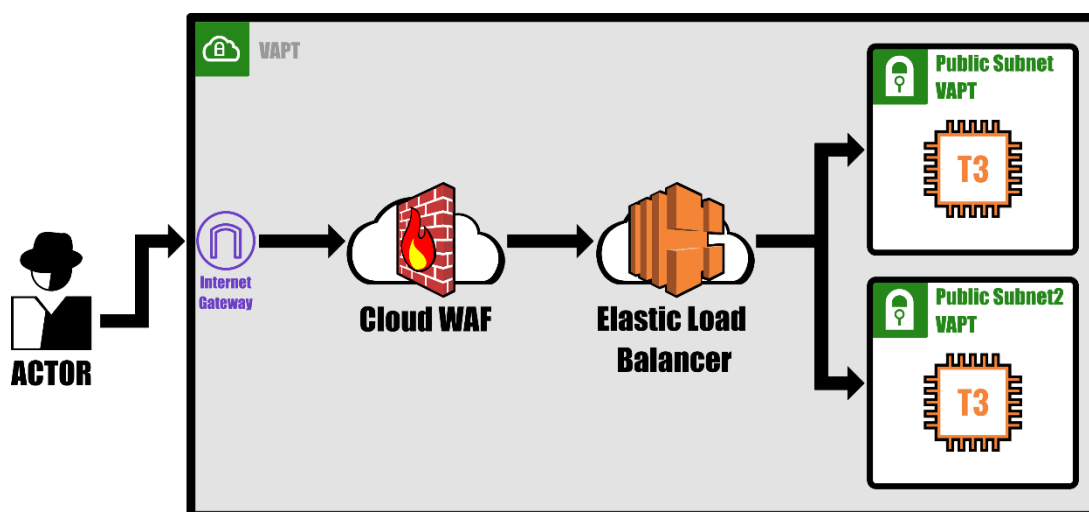


Figure 4. WAF deployment diagram

The cloud WAF was deployed with default policy with an elastic load balancer to protect the web-applications on AWS, see Figure 4. All web-application transactions were inspected by the cloud WAF. In doing so, the cloud WAF was expected to provide protections against threats that were originated by the malicious actors while allowing normal actors to access the web application resources.

During deployment, our engineers noted the time it took to deploy with out of the box controls and the complexity of the deployment. Also noted was whether our engineering team was required to contact the WAF vendor's support team to successfully complete the WAF deployment. See Table 6 for deployment findings.

### 2. TEST EXECUTION

SecureQLab performed security validation using crafted attacks that are relevant to today's cloud application hosted on cloud and cloud native applications. SecureQLab carefully curated such attacks via research generated by our own red team as well as the attacks that are prevalent in the wild. Open-source tool kits were also utilized while performing this assessment.

Before the testing was conducted, SecureQLab validated that the cloud WAF solution was in an operational state by verifying the following:

#### Connection Validation:

1. Before any test is conducted, SecureQLab ensures that the Cloud WAF can be accessed by the administrator and is passing normal application traffic. This is to ensure that any dynamic content such as IP blacklist protection can be updated on regular basis by the cloud WAF.

#### Logging:

2. SecureQLab understands that logging is a critical and crucial component while running a cloud WAF. SecureQLab verifies that the cloud WAF being tested will have sufficient administrative as well as attack logging to ensure Security Analyst can troubleshoot and fix issues as required.

Updates:

3. Protocol updates in the form of rules, signatures and reputations will be applied as they become generally available. SecureQLab will make best effort to apply these updates to the products prior to the evaluation.

The above processes were repeated wherever applicable throughout the test. Once the deployment of AWSs WAF solution and baseline testing were completed, the security validation testing began.

The first phase of attack was to gather information and perform reconnaissance against the application. The was done to gather as much information as possible to be utilized when penetrating the target during the vulnerability assessment and exploitation phases. SecureQLab performed vulnerability analysis using automated tools such as Burpsuite and Nessus in addition to performing manual analysis. The main objective of vulnerability analysis is to discover flaws in the systems and applications which can be leveraged by an attacker. These flaws ranged anywhere from host and service misconfiguration to insecure application design. Vulnerability Analysis was based on:

1. Active Scan: Active scan involves direct interaction with the component being tested for security vulnerabilities.
2. Passive Scan: Passive scan involves meta-data analysis and traffic monitoring.

Once information gathering and reconnaissance was completed, we began exploitation as the next phase in this process. Penetration testing was critical in the evaluation of cloud WAF technologies.

Once exploited, “post-exploitation” was undertaken. Post-exploitation refers to the actions taken after the initial compromise of a system or device. It often describes the methodical approach of using privilege escalation or pivoting techniques—which allowed SecureQLab, in this case, to establish a new source of attack from the new vantage point in the system—to gain additional access to systems or network resources. We demonstrate the risk presented by exploitable systems and what post-exploitation may likely occur with web applications.

Additionally, defense evasion is an important tool in an attacker’s arsenal. This allows old methods and techniques to be repurposed to evade protection against attacks which might otherwise get blocked by the Cloud WAF. More details on these techniques are covered in the Resiliency section.

The testing demonstrates the effectiveness of the product under test (PUT) to protect vulnerable assets from targeted threats and exploitation. This asset/target and threat-based approach forms the basis from which PUT security effectiveness is measured.

### 3. ATTACK TYPES

The SecureQLab threat and attack suite contains attacks (including mutations of the same underlying attacks) and proprietary exploits harvested through our test harness or crafted by our threat research team. SecureQLab has a number of complex web applications which have also been constructed to include known vulnerabilities and coding errors. Groups of exploits are carefully selected from this library to test based on the intended attack. Each exploit has been validated to impact the target vulnerable host(s) by compromising the asset, which can range from being the web server, the web application or sites.

The level of compromise can vary between instigating a denial-of-service (DoS) condition, providing administrator/root access to the host server, allowing malicious users to amend system parameters or application data before submission, browse and/or retrieve files stored on the host server, escalating user privileges, and so on.

### 4. AWS WAF CONFIGURATION

AWS WAF – Web Application Firewall version AWS WAFV2 was deployed and configured according to the

instructions found in the AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide<sup>12</sup>.

### SET UP WEB WAF

#### Step 1: Create Web ACL

To create a Web ACL:

- i. From the AWS home page, choose **Create web ACL**
- ii. For **Name**, enter the name that you want to use to identify this web ACL
- iii. **Note** Name cannot be changed after creating the web ACL
- iv. (Optional) For **Description - optional**, enter a description for the web ACL.
- v. For **CloudWatch metric name**, default name is present, or we can change it.
- vi. Resource type is set to the Regional Resources (Application Load Balancer, API Gateway, AWS AppSync)
- vii. **Note** Name cannot be changed after creating the web ACL
- viii. For Resource Type, select the resource type including **Amazon API Gateway** or **Application Load Balancer** or **AWS AppSync**.

Next click on the resource and click on **Add**.

#### Step 2: Add an AWS-Managed Rule Group:

Managed rule groups are created and maintained by AWS and AWS Marketplace sellers.

- i. On the **Add rules and rule groups** page, choose **Add rules**, and then choose **Add managed rule groups**.
- ii. On the **Add managed rule groups** page, expand the listing for the AWS managed rule groups. (You'll also see listings offered for AWS Marketplace sellers. You can subscribe to their offerings and then use them in the same way as for AWS Managed Rules rule groups.)
- iii. For the rule group that you want to add, do the following:
  - a) In the Action column, turn on the **Add to web ACL** toggle.
  - b) Select **Edit** and, in the rule, group's **Rule's** listing, turn on the **Set all rule actions to count** toggle. This sets the action for all rules in the rule group to count only. This allows you to see how all the rules in the rule group behave with your web requests before you put any of them to use.
  - c) Choose **Save rule**.

In the Add managed rule groups page, choose Add rules. This returns you to the Add rules and rule groups page.

#### Step 3: Add String Match Rule:

A string match rule statement identifies strings that you want AWS WAF to search for in a request, such as a specified value in a header or in a query.

- i. On the **Add rules and rule groups** page, choose **Add rules, add my own rules and rule groups, Rule builder**, then **rule visual editor**.

**Note** The console provides the **Rule visual editor** and a **Rule JSON editor**. The JSON editor makes it easy for you to copy configurations between web ACLs and is required for more complex rule sets, like those with multiple levels of nesting. This procedure uses the Rule visual editor

- ii. For **Name**, enter the name that you want to use to identify this rule.
- iii. For **Type** choose **Regular rule**.
- iv. For **If a request** choose matches the statement.
  - a. The other options use the logical statement types for rules, which allow you to combine or negate rule statement results.
- v. On **Statement**, for Inspect, open the dropdown, and choose the web request component that you want AWS WAF to look for your string in. For this example, choose **Header**.

<sup>12</sup> <https://docs.aws.amazon.com/waf/latest/developerguide/getting-started.html>

- a. When you choose **Header**, you also specify which header you want AWS WAF to inspect. Enter **User-Agent**. This value is not case sensitive.
- vi. For **Match type**, choose where the specified string must appear in the User-Agent header.
  - a. For this example, choose **Exactly matches string**. This indicates that AWS WAF inspects the user-agent header in each web request for a string that is identical to the string that you specify.
- vii. For **String to match**, specify a string that you want AWS WAF to search for. The maximum length of **String to match** is 200 characters. If you want to specify a base64-encoded value, you can specify up to 200 characters before encoding.
  - a. For this example, enter **MyAgent**. AWS WAF will inspect the User-Agent header in web requests for the value MyAgent.
- viii. Leave **Text Transformation** set to **None**.
- ix. For Action, select the action that you want the rule to take when it matches a web request either to allow, block or count the request that matches the statements
- x. Choose **Add rule**.
  - a. After of that wizard return to the add rules and rule groups page

### Step 4: Finish Web ACL Configuration

- i. On the Add rules and rule groups page, select the default web ACL action to **Allow** or **Block** for the request that do not match any rules
- ii. choose **Next**.
- iii. On the **Set rule priority** page, you can see the processing order for the rules and rule groups in the web ACL. AWS WAF processes them starting from the top. You can change the processing order by moving them up and down. To do this, select one in the list and choose Move up or move down.
- iv. Choose **Next**.
- v. On the **Configure metrics** page, for Amazon **CloudWatch metrics**, you can see the planned metrics for your rules and rule groups, and you can see the web request sampling options.
- vi. Choose **Next**
- vii. On the **Review and create web ACL** page, review your settings, then choose **Create web ACL**.

The wizard returns you to the Web ACL page, where your new web ACL is listed.

## 5. AWS WAF RULES:

### A. Rule Configuration

#### i. How to subscribe

Purchase Cyber Security Cloud Managed Rules in the AWS Marketplace.

- a) Sign into the AWS Management Console.
- b) In the navigation pane, choose **Marketplace**.
- c) From the Marketplace list, search for **the Cyber Security Cloud Managed Rules**.
- d) If you want to subscribe to the ruleset, choose **Continue**.

#### ii. How to deploy

Subscribe to Cyber Security Cloud Managed Rules, add the ruleset to your AWS WAF settings.

- a) Sign into the AWS Management Console.
- b) In the navigation pane, choose **AWS WAF**.
- c) Create **a new Web ACL**.
- d) In the Rules section, select **the Cyber Security Cloud ruleset** that you subscribed to.
- e) You're done.



## B. Implemented Rules

- Cyber Security Cloud Managed Rules for AWS WAF -HighSecurity OWASP Set-
- Cyber Security Cloud Managed Rules for AWS WAF -API Gateway/Serverless-

sqli-body-001	Use action defined in the rule
sqli-q5-001	Use action defined in the rule
oscommandi-body-001	Use action defined in the rule
oscommandi-q5-001	Use action defined in the rule
xss-body-001	Use action defined in the rule
xss-q5-001	Use action defined in the rule
nosqli-body-001	Use action defined in the rule
nosqli-q5-001	Use action defined in the rule
xxe-ssci-body-001	Use action defined in the rule
xxe-ssci-q5-001	Use action defined in the rule
sqli-body-002	Use action defined in the rule
sqli-q5-002	Use action defined in the rule
sqli-url-001	Use action defined in the rule
pathtraversal-body-001	Use action defined in the rule
pathtraversal-q5-001	Use action defined in the rule
pathtraversal-url-001	Use action defined in the rule
cookie-body-001	Use action defined in the rule
cookie-q5-001	Use action defined in the rule
ldapi-url-001	Use action defined in the rule
ssrf-multi-001	Use action defined in the rule
suspicious_access-url-001	Use action defined in the rule
bad_useragent-header-001	Use action defined in the rule

Figure 4. Snapshot of Ruleset

## 5. CONTACT INFORMATION

SecureQLab, LLC.

801 Barton Springs Road

9th Floor

Austin, TX 78704 USA

+1.512.575.3457

[www.secureqlab.com](http://www.secureqlab.com)

[info@secureqlab.com](mailto:info@secureqlab.com)

## 6. COPYRIGHT AND DISCLAIMER

This publication is Copyright © 2021 by SecureQLab®. Any use of the results, etc., in whole or in part, is ONLY permitted after the explicit written agreement of the management board of SecureQLab prior to any publication. SecureQLab cannot be held liable for any damage or loss, which might occur as result of, or in connection with, the use of the information provided in this paper. We take every possible care to ensure the correctness of the basic data, but a liability for the correctness of the research results cannot be taken by any representative of SecureQLab. We do not give any guarantee of the correctness, completeness, or suitability for a specific purpose of any of the information/content provided at any given time. No one else involved in creating, producing or delivering research results shall be liable for any indirect, special or consequential damage, or loss of profits, arising out of, or related to, the use or inability to use, the services provided by the website, research documents or any related data.

For more information about SecureQLab and the testing methodologies, please visit our website.

SecureQLab (September 2021)