SecureIQlab™

Bridging the enterprise cloud security gap

Cloud Web Application Firewall (WAF) CyberRisk Validation Methodology

Methodology Version:	Draft V1.0
Last Revision:	June 02, 2021
Language:	English

www.secureiqlab.com

Q2 2021

AMTSO Standard Compliance Statement

This Test has been designed to comply with the Anti-Malware Testing Standards Organization, Inc. ("AMTSO") Testing Protocol Standard for the Testing of Anti-Malware Solutions, Version [1.3] (the "Standard"). This Test Plan has been prepared using the AMTSO Test Plan Template and Usage Directions, Version [2.4]. SecureIQLab is solely responsible for the content of this Test Plan.

TABLE OF CONTENTS

1. Introduction

1.1 The Need for Web Application Firewall (WAF)

Attackers have moved up the stack. They are no longer simply attacking the web server and its underlying operating systems; they are attacking the web applications running on the web server that are front-ending critical corporate data. Such applications are often incredibly complex and difficult to secure effectively, and simple coding errors can render them wide open to remote exploits.

To regain the upper hand against current attacks, enterprises must in turn evolve their network defenses to provide a different kind of protection. Web application firewalls (WAF) exist to prevent web servers and their applications from being exploited.

The Web Application Firewall remains the most frequently used security control to protect web applications (84%). The WAF market size is expected to grow to USD 5.48 Billion by 2022, at a Compound Annual Growth Rate (CAGR) of 18.3% during the forecast period.

Cloud-based WAF on the rise:

- **By 2022, 75% of public facing applications will be protected by Cloud-based WAF**
- **95.1% of the enterprise based WAF controls are deployed in the cloud today**
- **69.2% of the enterprises managing their own cloud based WAF controls and 25% being managed by their Cloud providers. Only 6.2% of the WAF based cloud deployments are managed by a 3rd party/MSSP**
- **With Cloud Data Centre traffic to represent 95% of the total Data Centre (DC) traffic by 2021, it stands to reason that enterprises are and will be moving to a private, public or a hybrid-based model**
- **With the DC traffic primarily constituting HTTPS (75.9%) and HTTP (64.5%) based traffic, WAF's are expected to play a critical role in protecting applications**

1.2 Cloud WAF Benefits:

Cloud WAF technology allows for the creation of customize security and benefits organizations in the following ways:

Less complex to manage than on premise WAF solutions

Ease of integration with existing security solution

Scalable and elastic

Fast deployment and easy to set-up

Protect web applications against external and internal attacks

Able to monitor and control access to web applications

Allows all transactions except that contain threat/attack (Negative Security model)

Able to collect access logs for compliance/auditing and analytics

1.3 Proposed Cloud WAF Deployment Models:

- **Reverse Proxy**
- **Software as a Service (SaaS)**
- **IaaS deployment as a software appliance or virtual machine**

- **Offered as pay-as-you-grow service**

1.4 Statement of Intent:

The purpose of this inaugural Cloud Web App Firewall (WAF) test is to educate security practitioners, business managers and Enterprise by providing empirically validated data based upon industry guidelines such as OWASP while securing cloud applications. The results of the test can be used to make purchasing decisions, understanding product-know how's as well as to improve any shortcomings the product may have during the course of the testing. SecureIQLab believes that the outcome of the test will make better, safer products.

1.5 Testing Goals Include:

- **Publication of knowledgeable outcomes.**
- **Adherence to Industry Compliance that drives that WAF market.**
- **Accurate results available in Public Forum**
- **Highlight Key Technology differentiators.**

1.6 Cloud WAF Features to be Evaluated

Performance, Availability and Reliability:

- **Helps maximize throughput and ensures application High Availability (HA).**
- **Caching copies of regularly requested content**
- **Automatic content compression**
- **Load balancing web requests**
- **WAF is delivered through Application load balancer as well as through amazon Cloud Front**
- **PCI DSS Compliance**
- **Compliance mgmt. module**

Security Features

The following are the list of Cloud Web Application Firewall security features that will be validated:

- **Protection against attacks that can be mapped to OWASP Top 10.**
- **Protection against Multi-layered application-based attacks.**
- **Geolocation Attack protection from Layer 7 DDOS, SQL injection, Cross-site scripting and Zero-day web application attack.**
- **Protection against HTTP(s) attacks.**
- **Advanced Attacks**
    - **This will include protection against Bots that usually don't get detected by traditional security controls. These attacks use open-source tool kits, simulate users and have the ability to remain undetected. These bots have been used in account take over, content scraping, fraudulent transactions and payments. DNS tunneling has also been used in exfil activity by these bots.**

1.7 Cloud WAF Vendor Participation Selection Criteria

We select vendors based on three following criteria's:

1. **Market Leaders – Either in terms of revenue generated, customer numbers globally, or strong channel play**
2. **Analyst and Enterprise challengers – Small-mid-large enterprise security professional surveys, Direct 1:1 Inquiries and engagement with enterprises, organizations, MSP's, MSSP's and Gartner MQ, buyers guide, Forrester Wave, and IDC reports**
3. **New market entrants and interested participating vendors with breakthrough technology offerings**

There are no known conflicts of interest that exist now.

1.8 Scope:

The scope of this iteration of the test will be limited to Cloud WAF that are available in the AWS environment. Any physical WAF is out of the Scope of this methodology. Here is the list of considered vendors at the time of this publication:

Product Vendor	Product Name	Process Used
Akamai	Kona	*Test to be evaluated utilizing Blackbox Security and Greybox Security Tasks*
AWS	AWS WAF	
Barracuda	Barracuda WAF	
Check Point	CloudGaurd	
Citrix	Citrix WAF	
CloudFlare	Cloud WAF	
CromiWAF	CromiWAF	
Fastly	Next Generation WAF	
Fortinet	FortiWeb	
F1Security	F1-WebCastle	
F5	Advanced WAF	
Google	Cloud Armor	
Imperva	Imperva WAF	
Indusface	AppTrana WAF	
NSFOCUS Information Technology	NSFOCUS WAF	
Oracle	Oracle WAF	
Prophaze	Prophaze WAF	
Radware	AppWall	
SiteLock	TrueShield Premium	
StackPath	StackPath WAF	
Sucuri	Sucuri WAF	
Verizon	Verizon WAF	
VMware	NSX	

1.9 Funding Agreement:

Vendors are offered a non-committal partnership based on testing results with potential publication to provide insight and alignment with their specific security offerings. If there is a fee agreement in exchange for services rendered, vendors are given the option to progress forward with published results following AMTSO standards.

1.10 Opt-Out Policy

All vendors are provided the option to opt-out of the publishing of test results as reflected in the public test report, whether they meet satisfactory guidelines or conflict with specific adherence needs.

2. General Evaluation Approach

The aim of this section is to verify that the Cloud Web Application Firewall (WAF) referred here as the product under test (PUT) is capable of detecting, preventing, and logging attack attempts accurately, while remaining resistant to false positives.

The PUT can be configured either by “training” the PUT — walking through the applications, e-commerce and other sites as relevant (automatically, or manually) — or by creating rulesets and a security policy manually. Appropriate deployment model will be chosen and WAF will be deployed to protect against attacks that are targeting the assets beings protected. Examples of such assets will include typical e-commerce based applications.

Note: Since this is a Cloud WAF validation, there are no physical or appliance On-prem based configurations

2.1 Cloud WAF Security Effectiveness Validation

SecureIQLab will evaluate the security effectiveness of the Cloud WAF using the following approaches,

- **Blackbox Security Testing**
- **Greybox Security Testing**

Each of the categories above will consist of the following validation tasks:

Information Gathering and PUT Reconnaissance

Information gathering and reconnaissance will be performed against the application to gather as much information as possible to be utilized when penetrating the target during the vulnerability assessment and exploitation phases. SecureIQLab will perform vulnerability analysis using automated tools such as Burpsuite and Nessus and perform manual analysis. The main objective of vulnerability analysis is to discover flaws in systems and applications which can be leveraged by an attacker. These flaws can range anywhere from host and service misconfiguration, or insecure application design. Vulnerability Analysis will be based on:

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

Exploitation

Once Information gathering and reconnaissance is over, we will begin exploitation as the next phase in this process. Penetration testing is critical in the evaluation of Cloud WAF technologies.

Post Exploitation

The term "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 allows the tester, 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 will demonstrate the risk presented by exploitable systems and what post-exploitation may likely occur with web applications.

Defense Evasion Testing

Defense evasion is an important tool in an attacker's arsenal as old methods and techniques can be repurposed to evade protection against attacks which might otherwise get blocked by the Cloud WAF. SecureIQLab will focus defense evasion testing in the following areas.

1. **Preprocessor Attacks: Decide whether a request will be processed further. We will perform the pre-processor attack by identifying possible application inputs and end points.**
2. **Normalization: Standardize user input. We will perform the normalization task by tweaking the different end points for Example: compress Whitespace converts whitespace chars to spaces.**
3. **Validate Input with Payload: Check user input against policies. We will perform the fuzzing and will prepare the payload in order to bypass the security rules set by the Cloud WAF.**

2.2 Cloud WAF Test Life Cycle

The Cloud WAF test plan is within scope if the project remains within four weeks of the below timeline. This methodology is open to feedback/updates until it is finalized by June 14th.

SecureIQLab will execute the project in six phases.

Phase1: Reconnaissance
We will start the initial validation with basic and advance level reconnaissance

Phase 2: Attacking the pre-processor
As a part of the input validation, we will perform pre-processor attack by trying to skip input validation.

Phase 3: Attempting an impedance mismatch.
We will approach to make the WAF interpret a request differently than the backend and therefore not detect it.

Phase 4: Bypassing the rule set.
We will prepare a payload that will not be blocked and can bypass the WAFs rule set.

Phase 5: Identifying the vulnerabilities.
We will perform the security testing based on the guidelines around the OWASP Security Testing guideline.

Phase 6: Post Assessment Phase
We will review, assess and document the discovered vulnerabilities and the issues and will be tabulating the scorecard and prepare the final report.

The project's six phases are listed in graph format outlined below:

Sample Schedule Summary for Test Project

Index	Test Activity	Start Date Range	Dependencies
1	*Test Commencement*	*June 15th, 2021*	*Vendor Voluntary participation (or)*

			procurement of vendor Software
2	*Confirm Vendor Configuration Feedback*	*June 21s, 2021*	*All required vendors installed and testing commences without any problems*
3	*Milestone 1 – Preliminary Results*	*July 1st, 2021*	*Vendor confirmation and validation*
4	*Milestone 2 – Test Report First Edition – End of Testing Period*	*July 12th, 2021*	*Based on preliminary result disputes and resolution*
5	*Feedback and Dispute Resolution Time – Retests as Needed*	*July 16th, 2021*	*Based on report feedback and final dispute resolution.*
6	*Milestone 3 – Issue Final Report – End Date for Test*	*July 23rd , 2021*	*Based on retesting or testing period extended*

2.3 Risk and Risk Management:

No additional risks are known at this time.

2.4 Proposed Attack Types

The testing should provide a demonstration of effectiveness of the 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.

Attack types and Test Configuration: The SecureIQLab 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. We have 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 either 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.

2.5 Attack Relevance:

SecureIQLab will craft attacks that are relevant to today's cloud application hosted on cloud and cloud native applications. SecureIQLab 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 tools kits will also be utilized while performing this assessment.

2.6 GeoLimitations:

While Performing Web application attacks, SecureIQLab will ensure to the best of the ability based upon the resources available to perform attacks that are not geo-location centric. SecureIQLab will ensure that attacks are coming from broad geo-locations with different Internet Protocol as possible.

2.7 Distribution of Test Data:

Upon the completion of the six phases of this validation project, the resulting data will be organized into individual test reports and one comparative WAF CyberRisk graph. Based on vendor participation and approval, these results will then be publicly available to download at https://secureiqlab.com/publications/ and may also be available as resources from participating vendors.

3. Control Procedures

- **Connection Validation:**
  - **Before any test is conducted, SecureIQLab ensures that Cloud WAF can be accessed by the administrator and,**
  - **passing the normal application traffic. This is to ensure that any dynamic content such as IP black list protection can be updated on regular basis by Cloud WAF.**
- **Logging:**
  - **SecureIQLab understands that logging is a critical and crucial component on running Cloud WAF. SecureIQLab expects that Cloud WAF that will be tested has good amount of administrative as well as Attack logging to ensure Security Analyst can troubleshoot and fix issues as required.**
- **Updates:**
  - **Protocol updates in the form of rules, signatures and reputations will be applied as it becomes generally available. SecureIQLab will make best effort to apply these updates to the products prior to the evaluation.**

4. Dependencies

Participant and Test Subject Vendors Required Actions:

Vendors who chose to participate must follow guidelines to initiate, continue and complete the testing process. These steps include:

- **1.Setup**
- **2.Tuning**
- **3.Testing**
- **4.Score card generation for the Cloud WAF products.**

5. Score, Dispute Process, and Evidence Setting Process

For every Web-attacks blocked by Cloud WAF, SecureIQLab will give the block credit to the Cloud WAF under test. No credit will be given for missed attacks and there is no negative scoring for Web attacks missed by Cloud WAF.

Industry norms and best practices will be followed If there are any disputes on the nature of attacks used during the testing window. SecureIQLab will make best efforts to resolve disputes regarding the score. Any changes to scoring resulting from disputes will be applied to all vendor results.

All Cloud WAF vendors who participate in this test will receive their score with relevant metadata. This will include a recipe to reproduce the attacks that are missed by cloud WAF vendors during the test. This data set will be shared individually with the Cloud WAF vendors and SecureIQLab will work closely with Cloud WAF vendors to go over the metrics as well as relevant metadata are warranted. Furthermore, SecureIQLab will not share Web-attacks that are missed during the testing window to third party unless warranted by law. SecureIQLab will provide vendors 1-2 weeks for the dispute resolution on the nature of attacks. Any security vulnerabilities that are uncovered during the testing windows related to the Cloud WAF under test will be shared based upon responsible disclosure policy and will give the Vendors up to 60 days to fix the vulnerability. Vulnerability details will be disclosed to the broader public when a fix is available, or it is in the interest of the general public.

6. Attestations

I understand and agree that I am submitting this Test Plan, and the following Attestations, on behalf of the entity listed below, and I represent and warrant that I have authority to bind such entity to these Attestations. All references to “I” or “me” or similar language refer to such entity. I represent and warrant that the following Attestations are true, to the best of my knowledge and belief, and each of the following commitments will be upheld to the best of my ability.
I will provide public notification on the AMTSO website covering my obligation for notification of a Public Test, regardless of whether a potential Participant is in actual receipt of such notification prior to the Commencement Date of a Test.

All products included in this Test will be analyzed fairly and equally.

I will disclose any anticipated or known imbalance or inequity in the Test design to all Participants in the Test.

Although I may charge for participation in a Test, I will not charge any additional fees for a vendor to be a test subject under the Standards.
I will disclose any material conflicts of interest or other information that could materially impact the reliability of the Test.

I will disclose how the Test was funded.
I hereby affirm, to the best of my knowledge and belief that this Test Plan complies with the AMTSO Testing Standards, as of the date hereof.

Signature: /s/ David Ellis

Name: David Ellis

Test Lab: SecureIQLab

AMTSO Test ID: [AMTSO-LS1-TP039]

7. Appendix:

7.1 Document Revisions:

Version	Section	Revision overview

7.2 Attack Types:

URL Parameter Manipulation

Altering URL data to gain potentially protected information or access protected areas of a website.

Form/Hidden Field Manipulation

Constructing POST requests to access protected information or protected areas of a website, or to manipulate "fixed" data directly (such as pricing information).

Cookie/Session Poisoning

Manipulation of cookie or session variables to access protected information/areas of a website.

Cross-Site Scripting (XSS)

The process of manipulating user input in such a way that, when rendered in the context of a webpage, it will be interpreted by the browser as code.

Directory traversal

Altering the URL to access areas of the web server that should not otherwise be accessible

SQL Injection

Manipulating user input in such a way that, when processed by the database server, it will be interpreted as code, potentially providing direct access to private data.

Padding Oracle attacks

Altering a block-cypher cryptographic hash in such a way as to decrypt encrypted information.

Cross-Site Request Forgery (CSRF)

The process of executing a request on behalf of a user without their knowledge, using a trusted session between a vulnerable website and the user's browser.

Unmodified Exploit Validation

A number of common exploits are executed across the PUT to ensure that they are detected in their unmodified state. These will be chosen from a suite of older/common basic exploits for which SecureIQLab is certain that all vendors will have signatures/rules.

URL Obfuscation and Normalization

Random URL encoding techniques are employed to transform simple URLs, which are often used in pattern-matching signatures, to apparently meaningless strings of escape sequences and expanded path characters using one or any combination of techniques such as:

- **Escape encoding using various character sets**
- **Microsoft %u encoding**
- **Path character transformations and expansions**
- **Null-byte string termination**
- **HTML entities**
- **Base64**

- **Path references**
- **Padding**
- **Delimiters**

These techniques are combined in various ways for each URL tested, ranging from minimal transformation, to extreme (every character transformed). All transformed URLs are verified to ensure they still function as expected after transformation.

Sample OWASP based security effectiveness section for WAF:

OWASP Top 10
OWASP Category – Injection
SQL Injection
injection Search box - GET
Injection Malicious Character
Injection in URL - GET
Injection Search box - POST
injection Login Form - POST
Injection User Agent
Injection Stored Blog
Injection Blind Boolean-Based
SQLMap
Attack SQLab1
Attack SQLab2
XML Injection
SSI Injection
XPATH Injection
Code Injection
Command Injection
OWASP Category – Weak Authentication and Session Management
Privilege Escalation Admin param in URL
Privilege Escalation Admin param in Burp param
Session Fixation back button after logging out
Session Timeout
OWASP Category – Cross-Site Scripting
Cross-Site Scripting
Reflected GET
Malformed img tag 1
Malformed img tag 2
IMG on ERROR and javascript alert encode
Extraneous open brackets

Escaping escapes
SVG object tag
Body Tag
iFrame
Reflected POST
Malformed img tag 1
Malformed img tag 2
IMG on ERROR and javascript alert encode
Extraneous open brackets
Escaping escapes
SVG object tag
Body Tag
URL Encoding
Base64 Encoding
Reflected User Agent(Intercept on)
Malformed img tag 1
Malformed img tag 2
IMG on ERROR and javascript alert encode
Extraneous open brackets
Escaping escapes
SVG object tag
Body Tag
Stored User Agent
Malformed img tag 1
Malformed img tag 2
IMG on ERROR and javascript alert encode
Extraneous open brackets
Escaping escapes
SVG object tag
Body Tag
HTML Injection
Injected blog
Injected GET
Injected POST
iFrame injection
Normal iFrame
Encoded iFrame URL
Reflected URL

Standard URL
Encoded URL
OWASP Category – Insecure Direct Object Reference
Insecure Direct Object Reference
Change Password
Change Ticket price
Local and Remote File Inclusion
OWASP Category – Security Misconfiguration
Fingerprint Web Server
Fingerprint Web Application Framework:
HTTP Methods
OWASP Category – Sensitive Data Exposure
Insufficient TLS
Heartbleed
OWASP Category – Missing Function Level Access Control
Directory Traversal/File Include
File traversal
Directory traversal
OWASP Category – Cross-Site Request Forgery
CSRF Change Password
CSRF Transfer amount
OWASP Category – Using Components with Known Vulnerabilities
Denial of Service
XML DoS
Nginx DoS
Shellshock
PHP CGI Remote Code Execution
/admin/?-s
/admin/?-cat
OWASP Category – Unvalidated Redirects and Forwards
Client-Side URL Redirect
Redirect and forward 1
Redirect and forward 2

8. COPYRIGHT AND DISCLAIMER

This publication is Copyright © 2021 by SecureIQLab®. Any use of the results, etc., in whole or in part, is ONLY permitted after the explicit written agreement of the management board of SecureIQLab prior to any publication. SecureIQLab 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 SecureIQLab. 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 SecureIQLab and the testing methodologies, please visit our website.

SecureIQLab (May 2021)