TEST METHODOLOGY

Cloud Network Firewall

December 1, 2021

v1.2

Table of Contents

1 Overview

1.1 Cloud Network Firewalls

For this methodology the DUT[1] is a cloud network firewall. A firewall is a mechanism used to protect a trusted network from an untrusted network while allowing authorized traffic to pass from one side to the other. Throughout their history, the goal has been to enforce an access control policy between networks and thus should be viewed as an implementation of policy. While the firewall market is one of the largest and most mature security technology segments, cloud network firewalls are relatively new. Firewalls have undergone several stages of development, from early packet filtering and circuit relay firewalls to application layer (proxy-based) and dynamic packet filtering firewalls. This latest evolution virtualizes this functionality to provide scalable and elastic policy enforcement in a cloud environment.

1.2 Topology

The following diagram shows the general test topology for the Cloud Network Firewall deployment in the AWS cloud infrastructure. Note that there may be be variations in the Central Management System (CMS) instance deployment location.

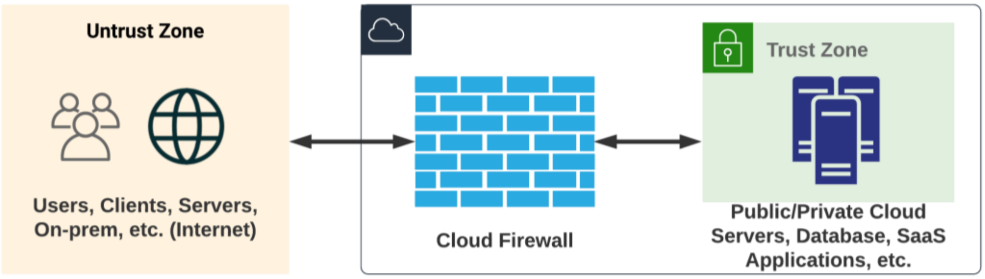
Untrust Zone
Users, Clients, Servers, On-prem, etc. (Internet)
Cloud Firewall
Trust Zone
Public/Private Cloud Servers, Database, SaaS Applications, etc.

Figure 1 – Cloud Network Firewall High-level Topology

1.3 What will be tested?

CyberRatings.org test reports are designed to address the challenges faced by security and IT professionals in selecting and managing security products. The scope of this particular methodology includes the following capabilities which are considered essential in a firewall:

- Basic Routing
- Access Control
- SSL/TLS Decryption
- Threat Prevention (exploits)
- Evasion
- Performance
- Stability and Reliability
- Management
- Reporting

[1] CyberRatings will continue to use this acronym (DUT - derived from Device Under Test). We recognize that other terms (e.g., appliance, platform, system, service) might be more applicable for certain technologies, but we will continue to use this legacy term for consistency across methodologies and reports. For this test, the DUT includes the instance in which the CNF is deployed.

Unless otherwise specified for a specific subtest, the DUT will be deployed using the default security policy or recommended settings available to the general public at the time of testing. No custom signatures are permitted. All vendors will be provided specific details about configuration requirements. For consistency across tests, DUTs will be deployed in controlled configurations to minimize the impact of confounding variables, (e.g., instance type, auto-scaling off, stream reassembly on, logging on, fragmentation on, etc.).

1.4 Inclusion Criteria

CyberRatings invites all leading cloud network firewall suppliers to include their products for testing at no cost. Products with significant market share as well as challengers with innovative technologies will be considered for inclusion. Decisions regarding inclusion of a vendor in a group test are based on an analysis of the market and an understanding of the criteria important to customers. Some of the elements considered are:

- Market presence
- Identification by industry analysts covering the specific technology area
- Consumer requests
- Innovative technology/offering or marketing claims that receive significant market attention (requires internal vetting for emerging vendors)

1.5 Product Guidance

CyberRatings issues product guidance based on evaluation criteria that is important to information security professionals. The evaluation criteria are as follows:

- **Security effectiveness** – How effectively does the cloud network firewall improve the security posture of the environment through policy enforcement?
- **Resistance to evasions** – How effectively does the product handle evasive techniques that would otherwise permit attackers to circumvent policy?
- **Stability and reliability** – How stable and reliable is the product in a production environment?
- **Performance** - How well does the product perform under varying network conditions?
- **Management** — What level of effort is required to install, configure/reconfigure, operate, and maintain the product?
- **Cost** – What are the licensing option models for the product and the associated operational costs over time?
- **Value** – How do the effectiveness and performance of the product rate within the context of cost?

2 Routing Functionality

This test includes some fundamental functionality that has a track record of working correctly in mature markets.

2.1 Network Segmentation

The cloud network firewall must support basic routing and network segmentation. The destination of traffic must be strictly enforced such that traffic does not "bleed" over into other networks.

2.1.1 Unrestricted Traffic Test

Test Objective: Does all traffic pass through the DUT when an "allow all" rule or policy is in place?

Test Approach: Testing would assess that traffic can pass through the DUT when there are no restrictions applied to the traffic. For this test, no other policies or rules are applied to restrict traffic.

2.1.2 Segmented Traffic Test

Test Objective: Does the DUT correctly direct traffic towards the intended network segment?

Test Approach: Testing would determine if all transmitted traffic reached the intended segment and no others.

3 Access Control

This includes testing basic routing and access control policies to ensure network traffic is properly transmitted and received across various network segments, and that policies that allow/restrict access to resources in the network are being properly enforced.

3.1 Policy Enforcement

Policies are rule sets and behaviors within a firewall to permit or deny access from one network resource to another based on identifying criteria such as source IP, destination IP, source port, destination port, and protocols. Policies are typically written to permit or deny network traffic across one or more of the following zones:

- Untrusted Zone – This zone is typically an external network and is considered to be unknown and not secure. An example of an untrusted network would be the Internet.
- Trusted Zone – This zone is typically an internal network, a network that is considered secure and protected. There are often multiple trusted networks in a cloud configuration.

This section of the test verifies that the DUT enforces security policies over a range of policy environments from simple to complex. The tests incrementally build on a baseline consisting of a simple configuration with no policy restrictions and no content inspection – to a complex multiple-zone configuration that supports many users, networks, policies, and applications. At each level of complexity, traffic will be tested ensuring specified policies are enforced.

3.1.1 Simple Policies

Test Objective: To what extent does the DUT correctly enforce simple policies (e.g., allow and deny specific and general traffic)?

Test Approach: Testing would determine whether all transmitted traffic that adhered to simple policies reached their intended destinations and whether all transmitted traffic that violated policies were blocked. Examples of policies include the following:

- An inbound policy allowing internal services to be accessed by untrusted users coming in from the Internet via core Internet services (HTTP, HTTPS, DNS, SMTP, IMAP, etc.)
- A deny-by-default action that blocks all traffic that has not been explicitly allowed
- A specific deny rule

3.1.2 Complex Multi-Zone Policies

Test Objective: Does the DUT correctly enforce complex policies with multiple zones?

Test Approach: Testing would determine whether all transmitted traffic that adhered to simple policies reached their intended destinations and whether all transmitted traffic that violated policies were blocked. Examples of policies include the following:

- An inbound policy allowing public shared services to be accessed by untrusted users coming in from the Internet via core Internet services (HTTP, HTTPS, DNS, SMTP, IMAP, etc.)
- Allow public shared services to access a second internal trusted zone for private services such as databases and APIs
- A specific deny rule

4 SSL/TLS Support

To address the growing threat of focused attacks using the most common web protocols and applications, CyberRatings tests the capabilities of cloud network firewalls to support a range of ciphersuites and provide visibility into the encrypted payloads to detect attacks concealed by encryption as well as attacks against the encryption protocols themselves.

4.1 Cipher Suite Support

To provide visibility into potential threats that are encrypted using SSL/TLS, the DUT is expected to support a wide range of commonly used cipher suites. Cipher suites are selected based on the published current frequency of use[2] and security status[3].

4.1.1 Current Cipher Suites

Test Objective: To what extent does the DUT support current cipher suites?

Test Approach: Testing will determine which cipher suites are supported. Tested cipher suites are selected based on frequency of use and security recommendations from reputable sources. The cipher suites available for this test include those listed in Table 1.

4.1.2 Insecure Cipher Suites

Test Objective: How does the DUT support the configuration of policies for insecure cipher suites?

Test Approach: Testing will determine how the DUT handles cipher suites known to be insecure including the following:

- Null cipher suites (no encryption of data provided)
- Anonymous cipher suites (no key authentication provided)

4.2 Decryption Validation

CyberRatings validates a cloud network firewall's ability to correctly decrypt and inspect SSL/TLS traffic prior to the associated performance testing. The DUT is expected to support all test cases with a single configuration.

Test Objective: To what extent can the DUT inspect encrypted traffic?

Test Approach: This test uses a set of exploits which have been successfully blocked by the DUT during testing. The exploit is then embedded in encrypted traffic to determine the capabilities of the DUT to enforce policy for encrypted streams.

4.3 Decryption Bypass Exceptions

The cloud network firewall is expected to support the configuration of policies that permit conditional bypass of decryption to preserve privacy, either for regulatory or other reasons (examples could be related to banking or medical data).

Test Objective: Does the DUT support exceptions to inspection policies?

Test Approach: The cloud network firewall will be tested to determine if it maintains decryption capabilities concurrently with inspection exception rules. (Turning off all decryption on a cloud network firewall would not be an acceptable method for meeting this requirement.)

4.4 TLS Session Reuse

To improve performance and reduce the overhead associated with conducting the full handshake for each session, the TLS protocol allows for abbreviated handshakes that reuse previously established sessions.

[2] Published international daily cipher suite usage can be found at https://crawler.ninja/files/ciphers.txt

[3] A list of cipher suites and associated attributes including security ratings can be found at https://ciphersuite.info/cs/

Test Objective: Does the DUT provide the option to reuse TLS sessions?

Test Approach: Testing will determine if session reuse is supported by the DUT.

Table 1 – Selected Cipher Suites[4]

Protocol	Cipher Suite Description	(Value)	Frequency Ranking	Security Classification
TLS 1.3	TLS_AES_256_GCM_SHA384	(0x13, 0x02)	1	Recommended
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0, 0x30)	2	Secure
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	(0xC0, 0x2F)	3	Secure
TLS 1.3	TLS_AES_128_GCM_SHA256	(0x13, 0x01)	4	Recommended
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	(0xC0, 0x28)	5	Weak
TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	(0xCC, 0xA8)	6	Secure
TLS 1.3	TLS_CHACHA20_POLY1305_SHA256	(0x13, 0x03)	7	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0, 0x2C)	8	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	(0xCC, 0xA9)	9	Recommended
TLS 1.2	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	(0x00, 0x9F)	10	Secure
TLS 1.2	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0, 0x2B)	11	Recommended
TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	(0x00, 0x39)	12	Weak
TLS 1.2	TLS_DHE_PSK_WITH_AES_256_GCM_SHA384	(0x00, 0xAB)	N/A	Recommended
TLS 1.2	TLS_ECDHE_PSK_WITH_AES_256_GCM_SHA384	(0xD0, 0x02)	N/A	Recommended
TLS 1.2	TLS_ECDHE_PSK_WITH_CHACHA20_POLY1305_SHA256	(0xCC, 0xAC)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	(0x00, 0xA2)	N/A	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256	(0xC0, 0x5C)	N/A	Recommended
TLS 1.2	TLS_DHE_PSK_WITH_CAMELLIA_128_GCM_SHA256	(0xC0, 0x90)	N/A	Recommended
TLS 1.2	TLS_DHE_PSK_WITH_CAMELLIA_256_GCM_SHA384	(0xC0, 0x91)	N/A	Recommended
TLS 1.2	TLS_DHE_PSK_WITH_AES_128_GCM_SHA256	(0x00, 0xAA)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_CAMELLIA_256_GCM_SHA384	(0xC0, 0x81)	N/A	Recommended
TLS 1.2	TLS_DHE_PSK_WITH_CHACHA20_POLY1305_SHA256	(0xCC, 0xAD)	N/A	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0, 0x86)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256	(0xC0, 0x56)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_CAMELLIA_128_GCM_SHA256	(0xC0, 0x80)	N/A	Recommended
TLS 1.2	TLS_ECDHE_PSK_WITH_AES_128_GCM_SHA256	(0xD0, 0x01)	N/A	Recommended
TLS 1.2	TLS_DHE_PSK_WITH_ARIA_128_GCM_SHA256	(0xC0, 0x6C)	N/A	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_ARIA_256_GCM_SHA384	(0xC0, 0x5D)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384	(0xC0, 0x57)	N/A	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0, 0x87)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	(0x00, 0xA3)	N/A	Recommended
TLS 1.2	TLS_DHE_PSK_WITH_ARIA_256_GCM_SHA384	(0xC0, 0x6D)	N/A	Recommended

[4] Cipher suite descriptions and associated value codes for testing are from https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

5 Threat Prevention

CyberRatings' threat prevention tests assess how accurately a cloud network firewall blocks and logs threats while remaining resistant to false positives (i.e., Type I errors). To accomplish this goal, the latest signature pack for the DUT will be acquired from the vendor's public support site, and the device will be deployed using vendor-recommended settings. Once deployed, the device's inspection capabilities will be restricted to publicly-available firmware and signature updates. All signature packs must be available to customers at the time of testing; no custom signatures will be permitted.

5.1 False Positives

False positives are any legitimate, non-malicious traffic that the DUT perceives as malicious and blocks or alerts on.

Test Objective: To what extent does the DUT block false positives (i.e., legitimate traffic that it is not supposed to block), and to what extent does eliminating false positives through policy changes impact protection?

Test Approach: Testing will determine which protections (e.g., signatures) trigger false positives. Signatures that trigger false positives must be disabled before the security testing begins. Consequently, the false positive test is repeated until no false positive signatures fire, and all legitimate traffic passes.

5.1.1 **Initial check – legitimate traffic, documents, and files**

This test transmits a varied sample of legitimate application traffic, documents, and files that should be identified and allowed or blocked based on policy rules.

5.2 Exploits

An exploit is an attack that takes advantage of a vulnerability in a protocol, product, operating system, or application. CyberRatings verifies that the DUT is capable of detecting and blocking exploits while remaining resistant to false positives by attempting to send exploits through the product under test; and verifying that the malicious traffic is blocked, and all appropriate logging and notifications are performed.

The CyberRatings exploit repository contains exploits that demonstrate a wide range of protocols and applications. Exploit sets for individual tests are selected based on CVSS score (how widely used is an application + what can an attacker do?), use case and relevance to customers. This has implications for the age of exploits since some applications in industrial environments are deployed and then left untouched for years while other applications within office environments are refreshed every 5-7 years. In addition, a container in a cloud implementation would not be browsing the internet, so attacks against web browsers would not be included in that use case, but they would be included when testing an edge firewall protecting employees in a corporate office.[5]

5.2.1 **No Background Network Load**

Test Objective: How effective is the DUT at blocking exploits when protection IS NOT resource-constrained?

Test Approach: Testing will determine if exploit traffic is blocked by the DUT, and if the event is recorded in the log. This test is conducted with no background network load.

5.2.2 **With Background Network Load**

Test Objective: How effective is the DUT at blocking exploits when protection IS resource-constrained?

Test Approach: Testing will determine if exploit traffic is blocked by the DUT, and if the event is recorded in the log. All tests are performed with varying levels and mixes of background network load.

[5] Vendors will be provided with a baseline set of malicious traffic prior to testing. These baseline samples will be used to verify basic protection capabilities and will not be part of the actual test.

6 Evasions

Attackers use evasion techniques to disguise and modify attacks at the point of delivery to avoid detection by security products. Passing evasion tests is very important since just one successful evasion technique can enable an attacker to exploit systems undetected. Previously blocked exploits will be rerun using evasion techniques.

6.1 Spoofing

6.1.1 IP Address Spoofing

IP address spoofing attempts to bypass policies by forging the IP header to indicate a different source address from where the packet was actually transmitted.

Test Objective: How effective is the DUT in detecting IP address spoofing?

Test Approach: Exploits that were previously blocked by the DUT will be resent with forged IP headers.

6.1.2 TCP Split Handshake Spoofing

The TCP split handshake spoofing blends features of both the three-way handshake and the simultaneous-open connection. The result allows an attacker to bypass the cloud network firewall by having the attacker instruct the target to “initiate” the session back with the attacker.

Test Objective: How effective is the DUT in detecting TCP split handshake spoofing?

Test Approach: Exploits that were previously blocked by the DUT will be retested using this evasive technique.

6.2 IP Packet Fragmentation

This test determines the effectiveness of the DUT’s fragment reassembly mechanism and the impact of resource constraints on this functionality.

Test Objective: How effective is the DUT in reassembling fragmented IP packets with/without resource constraints?

Test Approach: Evasions including manipulation of the following traffic attributes will be tested:

- Fragment size (8 to 32 bytes)
- Fragment order (ordered, out-of-order, reverse order, overlapping, favoring, etc.)
- Packet manipulation (Interleaving, duplicate packets, constant and random payloads, delays, etc.)
- Payload Variation (constant, random)
- Any combination of the above attributes with/without resource constraints

6.3 TCP Stream Segmentation

This test determines the effectiveness of the DUT’s TCP segment reassembly mechanism and the impact of resource constraints on this functionality.

Test Objective: How effective is the DUT at TCP segment reassembly with/without resource constraints?

Test Approach: Evasions including manipulation of the following traffic attributes will be tested:

- Segments size (1 - 2048 bytes)
- Segment order (ordered, out-of-order, reverse order, overlapping, favoring, etc.)
- Packet manipulation (Interleaving, duplicate packets, constant and random payloads, delays, etc.)
- Control flags (Invalid, null, etc.)
- Sequencing (resync requests, random initial sequence number, or out-of-window sequence numbers)
- Segment manipulation (faked retransmits, wrapping sequence numbers, segments containing random data, etc.)
- Any combination of the above methods with/without resource constraints

6.4 Layered Evasions

These tests determine the effectiveness of the DUT when subjected to combinations of evasion techniques.

Test Objective: How effective is the DUT in detecting exploits hidden behind multiple evasion techniques?

Test Approach: This test attempts to bypass the cloud network firewall by performing any legitimate combination of the previous evasion techniques.

7 Performance

Cloud security architects are tasked with designing environments that scale. Making an informed decision regarding the performance of a cloud network firewall in an environment requires understanding the impact of the DUT on traffic passing through it under various load conditions. The tests in this section measure the performance of a device using traffic scenarios that allow the inference of real-world performance. Individual implementations will vary based on use case; however, these quantitative metrics inform whether a particular device is appropriate for a given environment.

Each test is performed concurrently without the DUT to provide a baseline control. The test is then repeated with the DUT in the exact same configuration as used for exploits and evasions. Results are reported both as measured, relative to the baseline, and in context with other measurable attributes and confounding variables.

7.1 Raw Packet Processing Performance and Latency

This test uses UDP packets to determine the capacity of the DUT to transfer raw data and determines the associated latency. This test does not attempt to simulate any form of real-world network condition. No TCP sessions are created during this test. Each DUT must have a signature that detects the test packets to ensure that they are being passed through the detection engine.

Test Objective: How much raw data can be transferred through the DUT per second and what are the associated latency and drop packet counts?

Test Approach: A constant stream of UDP packets — with variable source and destination IP addresses transmitting from a fixed source port to a fixed destination port — is transmitted bi-directionally through the DUT. Each packet contains dummy data and is targeted at a valid port on a valid IP address on the target subnet. Testing will determine the maximum rate the DUT can process UDP packets of various sizes, the associated latency, and the number of dropped packets. The packet sizes and maximum frames targets include those shown in Table 2 and example descriptions of the associated conditions.

Table 2 - UDP Test Traffic Characteristics

L2 Ethernet Frame Size	L1 Ethernet Packet Overhead			L1 Ethernet Packet Size	L2 Ethernet Frames / Second
	Preamble	Start Frame	Interpacket Gap		
Bytes	Bytes	Bytes	Bytes	Bytes	
64	7	1	12	84	1,488,095
128	7	1	12	148	844,595
256	7	1	12	276	452,899
512	7	1	12	532	234,962
1,024	7	1	12	1,044	119,732
1,280	7	1	12	1,300	96,154
1,518	7	1	12	1,538	81,274

7.2 Theoretical Maximum Capacity

The use of traffic generation tools allows CyberRatings engineers to create traffic at multi-Gigabit speeds as a background load for the tests.

The goal is to stress the DUT and determine how it handles high volumes of TCP connections per second, HTTP transactions per second, and concurrent open connections. All packets contain valid payload and address data, and these tests provide an excellent measurement of maximum connection rates.

7.2.1 Theoretical Maximum Concurrent TCP Connections

This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible concurrent connections.

Test Objective: How many simultaneous users/sessions does the DUT support?

Test Approach: HTTP 1.1 persistent connections are opened across the DUT. Each connection is opened normally and then held open for the duration of the test as additional connections are added. Load is increased until no more connections can be established.

7.2.2 Maximum HTTP Transactions per Second

This test is designed to determine the maximum HTTP transaction rate of the DUT with a one- byte response size. The object size defines the number of bytes contained in the body, excluding any bytes associated with the HTTP header. A one-byte response size is designed to provide a theoretical maximum HTTP transactions per second rate.

Test Objective: What is the maximum HTTP transaction rate of the DUT with a one- byte HTTP response size?

Test Approach: The client will open a TCP connection, send 10 HTTP requests, receive 10 one-byte HTTP responses, and then close the TCP connection. (This ensures that TCP connections remain open until all 10 HTTP transactions are complete, thus eliminating the maximum connection per second rate as a bottleneck as one TCP/HTTP 1.1 connection = 10 HTTP transactions). The number of opened TCP connections are increased until the device is accepting more connections that it can process.

7.3 HTTP Capacity

The purpose of this test is to stress the detection engine to see how the device copes with HTTP network loads of varying average packet size and varying connections per second. By creating session-based traffic with varying session lengths, the device is forced to track valid TCP sessions, ensuring a higher workload than simple packet-based background traffic. The tested CPS values for this section are shown in Table 3.

Table 3: HTTP Test Traffic Characteristics

Connections per Second (per Gigabit)	HTML Response Size (bytes)	Total Response Size (bytes)
1,000	115,570	129,738
2,000	57,388	64,824
4,000	28,048	32,136
8,000	13,512	15,920
16,000	6,353	7,916
32,000	2,667	3,903

7.3.1 HTTP Capacity (without transaction delays)

The purpose of these tests is to determine the performance curve for HTTP connections.

Test Objective: How many HTTP connections can the DUT process and how does the size of what is being transferred impact performance?

Test Approach: Each transaction consists of a single HTTP GET request with no transaction delays (i.e., the web server responds immediately to all requests). All packets contain valid payloads. The tested CPS values are shown in Table 3.

7.3.2 HTTP Capacity (with transaction delays)

The purpose of these tests is to determine the performance curve for HTTP connections when introducing delays that more closely simulate actual activity.

Test Objective: How many HTTP connections can the DUT process and how does the introduction of transaction delays impact performance?

Test Approach: Each transaction consists of a single HTTP GET request with delays (i.e., the web server responds immediately to all requests, but the client waits for 10 seconds before closing the connection). All packets contain valid payloads. The tested CPS values are shown in Table 3.

7.4 HTTPS Capacity

The purpose of these tests is to stress the detection engine to see how the device copes with HTTPS network loads of varying average packet size and varying connections per second. By creating session-based traffic with varying session lengths, the device is forced to track valid TCP sessions, ensuring a higher workload than simple packet-based background traffic. The following cipher suites (Table 4) were selected from Table 1 for performance testing.

Table 4: Cipher Suites for Performance Tests

Protocol	Cipher Suite Description	(Value)	Frequency Ranking	Security Classification
TLS 1.3	TLS_AES_256_GCM_SHA384	(0x13, 0x02)	1	Recommended
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0, 0x30)	2	Secure
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	(0xC0, 0x2F)	3	Secure
TLS 1.3	TLS_AES_128_GCM_SHA256	(0x13, 0x01)	4	Recommended

7.4.1 HTTPS Capacity TLS_AES_256_GCM_SHA384 (0x13, 0x02)

The purpose of these tests is to determine the performance curve for HTTPS connections.

Test Objective: How many HTTPS connections can the DUT process and how does the size of what is being transferred impact performance?

Test Approach: Each transaction consists of a single HTTPS GET request with no transaction delays (i.e., the web server responds immediately to all requests). All traffic contains valid payloads. The tested CPS values are shown in Table 5.

Table 5: Cipher Suite Test Traffic Characteristics (0x13, 0x02)

TLS_AES_256_GCM_SHA384 (0x13, 0x02)		
Connections per Second (per Gigabit)	HTML Response Size (bytes)	Total Response Size (bytes)
1,000	113,430	127,666
2,000	54,917	62,455
4,000	25,700	29,710
8,000	11,170	13,483
16,000	3,870	5,358
32,000	150	1,227

7.4.2 HTTPS Capacity (0xC0, 0x30)

The purpose of these tests is to determine the performance curve for HTTPS connections.

Test Objective: How many HTTPS connections can the DUT process and how does the size of what is being transferred impact performance?

Test Approach: Each transaction consists of a single HTTPS GET request with no transaction delays (i.e., the web server responds immediately to all requests). All traffic contains valid payloads. The tested CPS values are shown in Table 6.

Table 6: Cipher Suite Test Traffic Characteristics (0xC0, 0x30)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xC0, 0x30)		
Connections per Second (per Gigabit)	HTML Response Size (bytes)	Total Response Size (bytes)
1,000	115,000	129,360
2,000	56,257	63,945
4,000	26,970	31,047
8,000	12,394	14,808
16,000	5,047	6,738
32,000	1,365	2,605

7.4.3 HTTPS Capacity (0xC0, 0x2F)

The purpose of these tests is to determine the performance curve for HTTPS connections.

Test Objective: How many HTTPS connections can the DUT process and how does the size of what is being transferred impact performance?

Test Approach: Each transaction consists of a single HTTPS GET request with no transaction delays (i.e., the web server responds immediately to all requests). All traffic contains valid payloads. The tested CPS values are shown in Table 7.

Table 7: Cipher Test Traffic Characteristics (0xC0, 0x2F)

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xC0, 0x2F)		
Connections per Second (per Gigabit)	**HTML Response Size (bytes)**	**Total Response Size (bytes)**
1,000	115,000	129,358
2,000	56,257	63,863
4,000	26,981	31,204
8,000	12,337	14,756
16,000	5,047	6,651
32,000	1,380	2,694

7.4.4 HTTPS Capacity with no delays (0x13, 0x01)

The purpose of these tests is to determine the performance curve for HTTPS connections.

Test Objective: How many HTTPS connections can the DUT process and how does the size of what is being transferred impact performance?

Test Approach: Each transaction consists of a single HTTP(S) GET request with no transaction delays (i.e., the web server responds immediately to all requests). All traffic contains valid payloads. The tested CPS values are shown in Table 8.

Table 8: Traffic Characteristics (0x13, 0x01)

TLS_AES_128_GCM_SHA256 (0x13, 0x01)		
Connections per Second (per Gigabit)	**HTML Response Size (bytes)**	**Total Response Size (bytes)**
1,000	113,430	127,781
2,000	54,917	62,381
4,000	25,700	29,781
8,000	11,170	13,413
16,000	3,870	5,365
32,000	150	1,239

8 Stability and Reliability

8.1 Protocol Fuzzing & Mutation

Testing will determine how the DUT responds (e.g., crashes, reboots, etc.) due to traffic generated from various protocol randomizers and mutation tools. The product is expected to remain operational and capable of detecting and blocking exploits throughout the test.

Test Objective: How stable is the DUT when exposed to mutated traffic?

Test Approach: This test stresses the protocol stacks of the cloud network firewall by exposing it to traffic from various protocol randomizers and mutation tools which may include ISIC, ESIC, TCPSIC, UDPSIC, ICMPSIC.

8.2 Blocking Under Extended Attack

This test provides an indication of the ability of the DUT to remain operational and stable (i.e., block violations and raise associated alerts) throughout a period of extended attack.

8.2.1 Blocking with Minimal Load

Test Objective: To what extent does the the DUT maintain policy enforcement reliability (i.e., block violations and raise associated alerts) under extended attack scenarios with no load?

Test Approach: A continuous stream of security policy violations mixed with legitimate traffic is transmitted through the product for an extended period of time with no additional background traffic. This is not intended as a stress test for traffic load (covered in the performance section); it is a reliability test for consistency of blocking.

8.2.2 Blocking Under Load

This test provides an indication of the ability of the DUT to remain operational and stable (i.e., block violations and raise associated alerts) throughout a period of extended attack with load.

Test Objective: To what extent does the the DUT maintain policy enforcement reliability (i.e., block violations and raise associated alerts) under extended attack scenarios with load?

Test Approach: This is intended as a stress test. This test adds legitimate traffic to the Blocking with Minimal Load test.

8.3 Behavior of the State Engine Under Load

This test determines whether the device is capable of preserving state across a large number of open connections over an extended period of time. At various points throughout the test (including after the maximum has been reached), it is confirmed that the device is still capable of inspecting and blocking traffic that is in violation of the currently applied security policy while confirming that legitimate traffic is not blocked (perhaps as a result of exhaustion of the resources allocated to state tables).

8.3.1 Attack Detection/Blocking – Normal Load

Test Objective: To what extent is the DUT able to detect and block policy violations as the number of concurrent connections is increased under normal load?

Test Approach This test determines whether the device is able enforce policy as the number of concurrent open connections increases.

8.3.2 State Preservation – Normal Load

Test Objective: To what extent is the DUT able to preserve state as the number of concurrent connections is increased under normal load?

Test Approach: A legitimate HTTP session is opened, and the first packet of a two-packet exploit is transmitted. As the number of open connections approaches the maximum, the initial HTTP session is completed with the second half of the exploit, and the session is closed. If the cloud network firewall is still maintaining state of the original session, the exploit will be recorded and blocked. If the state tables have been exhausted and the connection was removed from the state table AND fails open (to a bypass condition), the exploit string will not be reconstructed properly and will not be detected as both halves of the exploit are required to trigger an alert.

8.3.3 Pass Legitimate Traffic – Normal Load

Test Objective: To what extent is the DUT able to pass legitimate traffic while blocking exploits as number of open connections is increased?

Test Approach: See 8.3.2 Test Approach

8.3.4 State Preservation – Maximum Exceeded

Test Objective: To what extent is the DUT able to preserve state as the number of concurrent connections exceeds the maximum determined in section 8.2.1?

Test Approach: See 8.3.2 Test Approach

8.3.5 Drop Traffic – Maximum Exceeded

Test Objective: Does the DUT continue to drop all traffic as the number of concurrent connections exceeds the maximum determined in section 8.2.1?

Test Approach: See 8.3.2 Test Approach

9 Management Capabilities

It is important that a cloud network firewall provide comprehensive management control to accomplish the expected functionality. Further best practices in user experience (UX) should be fundamental to the associated interface.

The DUT will be assessed to determine the answers to the following questions:

9.1 Authentication

9.1.1 Role-Based Access Control (RBAC)

Test Objective: Does the DUT support RBAC?

9.1.2 Authentication

Test Objective: Are third-party authentication systems such as LDAP and Active Directory supported?

9.2 Policy

9.2.1 Policy Definition

Test Objective: Does the DUT allow multiple security policies?

9.2.2 View Policy

Test Objective: When an alert is selected, does the DUT provide the ability to access directly (single-click) and view the policy and rule that triggered the event?

9.2.3 Policy Association

Test Objective: Once policies have been defined, does the DUT provide the ability to apply them to specific users or groups?

9.2.4 Policy Inheritance

Test Objective: Does the DUT allow (by default) the creation of groups and sub-groups such that sub-groups can inherit configuration and policy definitions from parent groups?

9.2.5 Policy Version and Checksums

Test Objective: Does the DUT have functionality minimizing risk of tampering by third parties (e.g., recording the version and the hash of a policy, etc.)?

9.2.6 Bulk Operations

Test Objective: Does the DUT support bulk operations for particular groups/classes of signatures (e.g., search functionality, inclusion/exclusion, enable/disable, switches from block mode to log mode, etc.)?

9.3 Change Control

Test Objective: Change Control functionality and capabilities will be assessed to determine the depth of support for each of the following:

- Change Control Logging
- Roll-Back
- Revision History

10 Reporting Capabilities

Logging, alerting, and reporting are critical functions that inform your security posture and facilitate incident response actions. Reporting capabilities will be assessed to determine the ability of the DUT to support these requirements.

10.1 Logs

All cloud network firewall offerings will be tested to determine if they retain log and event data to support the incident response process. The use of standardized logging and reporting formats, which facilitate the fast and accurate consumption of presented data, is imperative to enable administrators to validate conviction accuracy. The cloud network firewall offering should allow easy generation and exportation of reports, logs, and alerts into industry-standard formats in support of incident response. (Aspects like log time normalization, log file maintenance options, and forensic traffic-capture will also be factored in the assessment.)

Test Objective: To what extent does the DUT collect and store information about events including the following?

- Malicious Traffic
- Administrator Login/Logout
- Successful Authentication
- Unsuccessful Authentication
- Policy Changes
- Policy Deployment

10.2 Alerts

The DUT will be assessed to determine the answers to the following questions:

10.2.1 Centralized Alerts

Test Objective: Are alerts are delivered to and handled centrally?

10.2.2 Performance Alerting

Test Objective: Does the DUT support performance-related alerts (e.g., disk quota is close to being exceeded, utilization exceeds a specific threshold, etc.)?

10.2.3 Alert Filtering

Test Objective: Does the DUT support filtering of alert contents and summaries for selected attributes (e.g., view all alerts for a selected source IP)?

10.2.4 View Alert Detail

Test Objective: Does the DUT support provide capabilities for in-depth information about alerts?

10.2.5 View Packet Contents

Test Objective: Does the DUT provide the capability to view the content of the the trigger packet or context data for the exploit associated with an alert?

10.2.6 Alert Suppression

Test Objective: Does the DUT provide the functionality to create exception filters based on alert data to eliminate further alerts that match the specified criteria (e.g., same alert ID from the same source IP)? (This does not disable detection, logging, or blocking but merely excludes alerts from the console display.)

10.2.7 Incident Workflow

Test Objective: Does the DUT provide the functionality to annotate and track incidents to resolution?

10.2.8 Correlation (automatic)

Test Objective: Does the DUT automatically infer connections between multiple alerts and group them together as incidents?

10.2.9 Correlation (manual)

Test Objective: Does the DUT allow the administrator to manually infer connections between multiple alerts and group them together as incidents?

10.3 Reports

Reporting functionality is critical to ascertaining the state of the system and investigating incidents. The DUT will be assessed to determine the answer to the following questions:

10.3.1 Custom Reports

Test Objective: Does the DUT include a report generator that can construct complex data filters and summarize alerts on selected criteria?

10.3.2 Saved Reports

Test Objective: To what extent does the DUT allow you to create and save custom report filter templates?

10.3.3 Report Automation

Test Objective: To what extent is automated report scheduling and delivery supported?

10.3.4 Centralized Reports

Test Objective: To what extent does the DUT support the summary reporting functionality critical to support investigation of incidents.

11 Total Cost of Ownership

Implementation of security products can be complex, with many factors contributing to acquisition, deployment, and upkeep costs.

Test Objective: How much will it cost to adopt, operate, and maintain this DUT?

Test Approach: The following will be be considered in the context of the cloud product life cycle to determine the Total Cost of Ownership (TCO).

- **Adoption Cost –** Fixed and variable costs to acquire a license or subscribe to a service.
- **Vendor Support** – Fees paid to the vendor to provide support throughout the product life cycle.
- **Implementation Time –** Time required to install and configure the DUT in a production environment.
- **Provisioning Cost –** Cost of the virtual infrastructure required to support the product in a production environment.
- **Upkeep** – Effort required to keep product current (e.g., apply periodic updates, vendor patches, etc.).
- **Operational Management -** Time commitment for day-to-day operations within a production environment, including support for monitoring logs, updating policies, and supporting incident investigations.

Contact Information

CyberRatings.org

2303 Ranch Road 620 South
Suite 160, #501
Austin, TX 78734

info@CyberRatings.org

www.CyberRatings.org

This and other related documents available at: **www.CyberRatings.org**. To receive a licensed copy or report misuse, please contact CyberRatings.

1. The information in this report is subject to change by us without notice, and we disclaim any obligation to update it.

2. The information in this report is believed by us to be accurate and reliable at the time of publication, but is not guaranteed. All use of and reliance on this report are at your sole risk. We are not liable or responsible for any damages, losses, or expenses of any nature whatsoever arising from any error or omission in this report.

3. NO WARRANTIES, EXPRESS OR IMPLIED ARE GIVEN BY US. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, ARE HEREBY DISCLAIMED AND EXCLUDED BY US. IN NO EVENT SHALL WE BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, PUNITIVE, EXEMPLARY, OR INDIRECT DAMAGES, OR FOR ANY LOSS OF PROFIT, REVENUE, DATA, COMPUTER PROGRAMS, OR OTHER ASSETS, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

4. This report does not constitute an endorsement, recommendation, or guarantee of any of the products (hardware or software) tested or the hardware and/or software used in testing the products. The testing does not guarantee that there are no errors or defects in the products or that the products will meet your expectations, requirements, needs, or specifications, or that they will operate without interruption.

5. This report does not imply any endorsement, sponsorship, affiliation, or verification by or with any organizations mentioned in this report.

6. All trademarks, service marks, and trade names used in this report are the trademarks, service marks, and trade names of their respective owners.