NOTA TÉCNICA · RENDIMIENTO

¿Cuánto le cuesta a su firewall una blocklist de OneFirewall?

Lo hemos medido. Comprobar una dirección en una blocklist de 300.000 entradas tardó como máximo 53,0 ns (milmillonésimas de segundo), una vez por cada nueva conexión. Mantener la lista actualizada requiere unos pocos kilobits por segundo.

53,0 ns	0,00037%	3,8 kbit/s
búsqueda más lenta en una lista de 300.000 entradas	de un round trip de red de 14,3 ms dentro de Europa	descarga media de una lista de 10.000 entradas actualizada cada 5 minutos

A ESCALA



La barra mide 170 mm. A la misma escala, la búsqueda mediría 0,63 milésimas de milímetro de ancho, demasiado fina para imprimirse. Dicho de otro modo: si el round trip durara un día entero, la búsqueda duraría 0,32 s. Si fuera un paseo de un kilómetro, la búsqueda serían los últimos 3,7 mm.

LO QUE CUESTA

Por nueva conexión
como máximo 53,0 ns

Por millón de nuevas conexiones
53 ms de tiempo de procesador

Tráfico de descarga
3,8 kbit/s (10.000 entradas, cada 5 minutos)

Camino del tráfico
sin cambios; nada pasa por OneFirewall

LO QUE DETUVO EN PRODUCCIÓN

35.000 ataques al día
eliminados para una plataforma SaaS en 24 horas. Su latencia bajó un 28%. [11]

El 92,6% de las IP atacantes
en la oleada de explotación de PHP2Shell ya era conocido por OneFirewall. Esas IP enviaron el 99,96% de los 32.252 intentos. [12]

8.934 eventos críticos y altos
interceptados en 19 días en una aplicación web, donde el 12% del tráfico que permitía su firewall era malicioso. [13]

Las cifras de producción proceden de nuestros casos de estudio publicados, no de esta prueba.

> El intercambio está desequilibrado. El firewall dedica unas decenas de nanosegundos a cada nueva conexión y, a cambio, las conexiones de atacantes conocidos se descartan en su primer paquete.

1. Cómo lo medimos

El único trabajo por conexión que añade un feed de OneFirewall es una búsqueda: ¿está esta dirección de origen en la lista? El firewall ya hace una comprobación de política para cada nueva conexión, así que el feed solo alarga la lista de direcciones que comprueba. Esa búsqueda es lo que cronometramos.

Escribimos un pequeño benchmark en C y lo ejecutamos en un Apple M3 con 16 GB de RAM (macOS 26.7.1; Apple clang version 21.0.0 (clang-2100.1.1.101); -O2). Crea listas de direcciones IPv4 públicas aleatorias, de 1.000 a 1.000.000 de entradas, en las dos formas que más usan los firewalls. Una tabla hash encuentra una dirección en aproximadamente un paso. Una tabla de rangos ordenada se busca dividiendo por la mitad, necesita unos 20 pasos con un millón de entradas y es la más lenta de las dos.

Cada configuración ejecutó 16.777.216 búsquedas siete veces, y nos quedamos con la mediana. Cada búsqueda esperaba el resultado de la anterior, así que el procesador no podía solaparlas; los tiempos de abajo son el coste completo de una sola búsqueda, incluidos 0,3 ns de overhead del bucle. Repetimos la prueba completa tres veces y damos la más lenta de las tres.

Tiempo de una búsqueda (la más lenta de tres pruebas)

Entradas	Hash, dirección no está	Hash, dirección está	Rangos, no está	Rangos, está	Memoria (hash / rangos)
1.000	10,4 ns	4,7 ns	23,3 ns	10,2 ns	8 KB / 8 KB
10.000	5,8 ns	2,9 ns	20,4 ns	28,0 ns	131 KB / 80 KB
100.000	8,3 ns	4,0 ns	37,9 ns	47,5 ns	1,0 MB / 800 KB
300.000	6,0 ns	3,2 ns	39,0 ns	53,0 ns	4,2 MB / 2,4 MB
1.000.000	15,0 ns	5,3 ns	79,0 ns	63,1 ns	8,4 MB / 8,0 MB

“No está” es el tráfico ordinario, el caso más común; “está” es el tráfico de un atacante incluido en la lista. Los tiempos de la tabla hash dependen de lo llena que esté la tabla (29–49% aquí), no del número de entradas, y por eso 1.000 entradas salieron más lentas que 10.000. 1 ns es una milmillonésima de segundo; 1 μs son 1.000 ns; 1 ms son 1.000.000 ns.

2. Mantener la lista actualizada

El dispositivo descarga la lista en texto plano, una dirección por línea. Generamos listas de direcciones públicas aleatorias y las medimos: 14,2 bytes por entrada. La columna gzip es solo orientativa: las direcciones aleatorias se comprimen peor que las reales, y el uso de compresión depende del servidor y del dispositivo. La tabla de tráfico usa el tamaño sin comprimir.

Tamaño de la lista

Entradas	Texto plano	gzip	Bytes por entrada
10.000	142 KB	52 KB	14,2
100.000	1,4 MB	467 KB	14,2
300.000	4,3 MB	1,3 MB	14,2

Tráfico de descarga al día, con la velocidad media entre paréntesis

Entradas	Cada 5 min	Cada 30 min	Cada hora	Una vez al día
10.000	41,0 MB (3,8 kbit/s)	6,8 MB (0,63 kbit/s)	3,4 MB (0,32 kbit/s)	142 KB (0,01 kbit/s)
100.000	410 MB (38 kbit/s)	68,3 MB (6,3 kbit/s)	34,2 MB (3,2 kbit/s)	1,4 MB (0,13 kbit/s)
300.000	1,2 GB (114 kbit/s)	205 MB (19 kbit/s)	102 MB (9,5 kbit/s)	4,3 MB (0,40 kbit/s)

El cliente elige el intervalo de actualización, entre 60 segundos y 24 horas. Cambia lo reciente que es la lista y nada más.

3. Por qué la búsqueda se paga una vez por conexión

Los firewalls que siguen sesiones comprueban la política en el primer paquete de una conexión y gestionan el resto desde la tabla de sesiones. En FortiGate, las sesiones aceptadas se delegan al procesador NP y los paquetes siguientes ya no vuelven a pasar por la política [3]. Las cifras anteriores se añaden, por tanto, una vez por conexión y no una vez por paquete.

Búsqueda más lenta como parte de un round trip de red

Round trip (Verizon, octubre de 2026) [10]	300.000 entradas (53,0 ns)	1.000.000 de entradas (79,0 ns)
Dentro de Europa, 14,3 ms	0,00037%	0,00055%
Dentro de Norteamérica, 33,4 ms	0,00016%	0,00024%
De Londres a Nueva York, 70,4 ms	0,000075%	0,00011%

Como referencia, Fortinet especifica para un paso completo por el firewall de un FortiGate 100F 4,97 μs con paquetes UDP de 64 bytes [4], unas 94 veces nuestra búsqueda más lenta.

4. Límites de cada plataforma

Cada plataforma tiene su propio tamaño máximo de lista y sus opciones de actualización. OneFirewall limita cada feed al máximo de la plataforma, así que una lista nunca supera lo que el fabricante previó para el dispositivo.

Plataforma	Opciones de actualización	Tamaño máximo de la lista	Otros límites
FortiGate	De 1 a 43.200 min, 5 por defecto [1]	300.000 IP por dispositivo, FortiOS 7.4.4 y posteriores [2]	Las entradas por encima del límite no se cargan [2]
Palo Alto Networks	5 min (por defecto), cada hora, día, semana, mes [7]	50.000 IP en total; 150.000 en PA-3200, PA-5200, PA-7000 [8]	Hasta 30 listas [8]
AWS WAF (CloudFront, ALB)	Cualquiera; la fija la programación de la sincronización	10.000 por IP set [5]	Una regla de IP set cuesta 1 WCU sea cual sea su tamaño [5]; una llamada de actualización por segundo por cuenta y Region [6]
Check Point	Un intervalo para todos los feeds, en segundos; 30 min por defecto en Quantum Spark R82 [9]	No publicado	

5. Lo que esta prueba no dice

La ejecutamos en un procesador de uso general, no en hardware de firewall, y no medimos ningún modelo de firewall. Los firewalls usan sus propias estructuras de datos y muchos usan chips dedicados, así que sus valores absolutos serán distintos de los nuestros. Cronometramos solo la búsqueda, no el recorrido completo de un paquete por el firewall. Nuestras direcciones eran aleatorias, mientras que las listas y el tráfico reales tienen

patrones que esta prueba no modela. Los round trips son mediciones publicadas por Verizon, no nuestras. OneDevice en Serie y WAP están en el camino del tráfico por diseño y quedan fuera del alcance de esta nota.

El benchmark tiene unas 250 líneas entre C y Python. Se ejecuta con un solo comando y guarda todos los resultados en un archivo JSON; lo compartimos con gusto bajo petición.

Fuentes

1. Fortinet, FortiOS 7.4.0 Administration Guide: Configuring a threat feed (refresh rate 1-43,200 min, default 5) · https://docs.fortinet.com/document/fortigate/7.4.0/administration-guide/379433/configuring-a-threat-feed
2. Fortinet, FortiOS 7.4.4 Administration Guide: Threat feeds (device-wide entry and file-size limits) · https://docs.fortinet.com/document/fortigate/7.4.4/administration-guide/9463/threat-feeds
3. Fortinet, FortiOS 7.4.4 Hardware Acceleration Guide (NP fast path: policy match on the first packet, session offloaded) · https://fortinetweb.s3.amazonaws.com/docs.fortinet.com/v2/attachments/6613da86-0277-11ef-8c42-fa163e15d75b/fortios-hardware-acceleration-744.pdf
4. Fortinet, FortiGate 100F Series data sheet (firewall latency 4.97 µs, 64-byte UDP; firewall throughput 20 Gbps)
5. AWS WAF Developer Guide: IP set match rule statement (up to 10,000 addresses, 1 WCU) · https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-ipset-match.html
6. AWS WAF Developer Guide: AWS WAF quotas (5,000 WCU per web ACL, 1 Create/Put/Update call per second) · https://docs.aws.amazon.com/waf/latest/developerguide/limits.html
7. Palo Alto Networks, PAN-OS 11.1: Objects > External Dynamic Lists (refresh: five minutes by default, hourly, daily, weekly, monthly) · https://docs.paloaltonetworks.com/ngfw/help/11-1/objects/objects-external-dynamic-lists
8. Palo Alto Networks, PAN-OS 11.1 Admin Guide: External Dynamic List (IP capacity per model, 30 EDLs) · https://docs.paloaltonetworks.com/pan-os/11-1/pan-os-admin/policy/use-an-external-dynamic-list-in-policy/external-dynamic-list
9. Check Point, Quantum Spark R82.00.X Admin Guide: IoC Feeds (fetched every 30 minutes by default); sk132193 Custom Intelligence Feeds · https://sc1.checkpoint.com/documents/Appliances/Quantum_Spark_R82.00.X/AdminGuides_Locally_Managed/EN/Content/Topics/IoC_Feeds.htm
10. Verizon, Monthly IP Latency Data (October 2026: Europe 14.263 ms, North America 33.354 ms, Trans-Atlantic 70.440 ms) · https://www.verizon.com/business/terms/latency/
11. OneFirewall Alliance, SaaS platform case study · https://onefirewall.com/es/case-study
12. OneFirewall Alliance, PHP2Shell threat briefing · https://onefirewall.com/es/php2shell-threat-briefing
13. OneFirewall Alliance, Web application case study · https://onefirewall.com/es/case-study-web-application

---

Elaborado por OneFirewall Engineering, Londres · support@onefirewall.com · https://onefirewall.com/es/performance-impact