Procesamiento de una única secuencia de gran sesión (flujo de elefantes) por servicios Firepower

Contenido

Introducción

Este documento describe por qué un único flujo no puede consumir todo el rendimiento nominal de un dispositivo Cisco Firepower.

Antecedentes

El resultado de cualquier sitio web de pruebas de velocidad de ancho de banda o el resultado de cualquier herramienta de medición de ancho de banda (por ejemplo, iperf) podría no exhibir la clasificación de rendimiento anunciada de los appliances Cisco Firepower. De manera similar, la transferencia de un archivo muy grande sobre cualquier protocolo de transporte no demuestra la clasificación de rendimiento anunciada de un dispositivo Firepower. Ocurre porque el servicio Firepower no utiliza un único flujo de red para determinar su rendimiento máximo.

Tráfico de proceso por Snort

La tecnología de detección subyacente del servicio Firepower es Snort. La implementación de Snort en el dispositivo Cisco Firepower es un proceso de subproceso único para procesar el tráfico. Un dispositivo se califica para una clasificación específica en función del rendimiento total de todos los flujos que pasan a través del dispositivo. Se espera que los dispositivos se implementen en una red corporativa, generalmente cerca del borde y que funcionen con miles de conexiones.

Firepower Services utiliza el balanceo de carga del tráfico a un número de procesos Snort diferentes con un proceso Snort que se ejecuta en cada CPU del dispositivo. Idealmente, la carga del sistema equilibra el tráfico de manera uniforme en todos los procesos de Snort. Snort debe poder proporcionar un análisis contextual adecuado para las inspecciones de firewall de última generación (NGFW), sistema de prevención de intrusiones (IPS) y protección frente a malware avanzado (AMP). Para asegurarse de que Snort es más efectivo, todo el tráfico de un flujo único se balancea de carga en una instancia de snort. Si todo el tráfico de un único flujo no se equilibraba a una única instancia de resnorte, el sistema podría eludirse y el tráfico se desbordaría de tal manera que una regla de Snort podría coincidir menos o las partes de un archivo no son contiguas para la inspección de AMP. Por lo tanto, el algoritmo de balanceo de carga se basa en la información de conexión que puede identificar de forma única una conexión determinada.

Algoritmo de 2-4 veces en ASA con Firepower Services y NGIPS virtual

En el dispositivo de seguridad adaptable (ASA) con la plataforma de servicio Firepower y el sistema de prevención de intrusiones de última generación (NGIPS) virtual, el tráfico se equilibra con la carga para realizar el Snort con el uso de un algoritmo de 2 tuplas. Los puntos de datos para este algoritmo son:

- IP de origen
- IP de destino

Algoritmo de 3 veces en la versión 5.3 o inferior del software en los appliances Firepower y FTD

En todas las versiones anteriores (5.3 o inferior), el tráfico se balancea de carga a Snort que utiliza un algoritmo de 3 tuplas. Los puntos de datos para este algoritmo son:

- IP de origen
- IP de destino
- Protocolo IP

Cualquier tráfico con el mismo origen, destino y protocolo IP se equilibra con la carga en la misma instancia de Snort.

Algoritmo de 5-tuplas en la versión de software 5.4, 6.0 y superior en los dispositivos FirePOWER y FTD

En la versión 5.4, 6.0 o posterior, el tráfico se equilibra de carga a Snort con un algoritmo de 5 tuplas. Los puntos de datos que se tienen en cuenta son:

- IP de origen
- Puerto de Origen
- IP de destino
- Puerto de Destino
- Protocolo IP

El propósito de agregar puertos al algoritmo es equilibrar el tráfico de manera más uniforme cuando hay pares de origen y destino específicos que representan grandes porciones del tráfico. Además de los puertos, los puertos de origen efímeros de alto orden deben ser diferentes por flujo y deben agregar entropía adicional de manera más uniforme que equilibre el tráfico con

diferentes instancias de snort.

Rendimiento total

El rendimiento total de un dispositivo se mide en función del rendimiento agregado de todas las instancias de sondeo que funcionan al máximo potencial. Las prácticas estándar del sector para medir el rendimiento son para varias conexiones HTTP con varios tamaños de objeto. Por ejemplo, la metodología de prueba de NGFW de NSS mide el rendimiento total del dispositivo con objetos de 44.000, 21.000, 10.000, 4.400 y 1.700. Esto se traduce en un rango de tamaños de paquete promedio de aproximadamente 1 k y bytes a 128 bytes debido a los otros paquetes involucrados en la conexión HTTP.

Puede estimar la calificación de rendimiento de una instancia individual de Snort. Tome el rendimiento nominal del dispositivo y divida eso por el número de instancias de Snort que se ejecutan. Por ejemplo, si un dispositivo se califica a 10 Gbps para IPS con un tamaño medio de paquete de 1.000 bytes y ese dispositivo tiene 20 instancias de Snort, el rendimiento máximo aproximado para una sola instancia sería de 500 Mbps por Snort. Diferentes tipos de tráfico, protocolos de red, tamaños de los paquetes, junto con las diferencias en la política de seguridad general pueden afectar el rendimiento observado del dispositivo.

Resultado de la prueba de herramientas de terceros

Cuando se realiza la prueba con cualquier sitio web de pruebas de velocidad o cualquier herramienta de medición del ancho de banda, como, iperf, se genera un flujo TCP de flujo único grande. Este tipo de flujo TCP grande se denomina flujo elefante. Un flujo de elefante es una única sesión, una conexión de red que se ejecuta durante relativamente tiempo y que consume una gran o desproporcionada cantidad de ancho de banda. Este tipo de flujo se asigna a una instancia de Snort, por lo que el resultado de la prueba muestra el rendimiento de una única instancia de snort, no la calificación de rendimiento agregado del dispositivo.

Síntomas observados

CPU alta observada

Otro efecto visible de los Flujos de Elefantes puede ser la cpu alta de la instancia de snort. Esto se puede ver a través de "show asp inspect-dp snort", o con la herramienta "top" del shell.

```
> show asp inspect-dp snort

SNORT Inspect Instance Status Info

Id  Pid   Cpu-Usage  Conns   Segs/Pkts  Status tot (usr | sys)
--  ----  ---------  -----   --------   ----------

0  48500  30% ( 28%|  1%)  12.4 K      0        READY
1  48474  24% ( 22%|  1%)  12.4 K      0        READY
2  48475  34% ( 33%|  1%)  12.5 K      1        READY
3  48476  29% ( 28%|  0%)  12.4 K      0        READY
4  48477  32% ( 30%|  1%)  12.5 K      0        READY
5  48478  31% ( 29%|  1%)  12.3 K      0        READY
```

```
6  48479  29% ( 27%|  1%)  12.3 K      0        READY
7  48480  23% ( 23%|  0%)  12.2 K      0        READY
8  48501  27% ( 26%|  0%)  12.6 K      1        READY
9  48497  28% ( 27%|  0%)  12.6 K      0        READY
10 48482  28% ( 27%|  1%)  12.3 K      0        READY
11 48481  31% ( 30%|  1%)  12.5 K      0        READY
12 48483  36% ( 36%|  1%)  12.6 K      0        READY
13 48484  30% ( 29%|  1%)  12.4 K      0        READY
14 48485  33% ( 31%|  1%)  12.6 K      0        READY
15 48486  38% ( 37%|  0%)  12.4 K      0        READY
16 48487  31% ( 30%|  1%)  12.4 K      1        READY
17 48488  37% ( 35%|  1%)  12.7 K      0        READY
18 48489  34% ( 33%|  1%)  12.6 K      0        READY
19 48490  27% ( 26%|  1%)  12.7 K      0        READY
20 48491  24% ( 23%|  0%)  12.6 K      0        READY
21 48492  24% ( 23%|  0%)  12.6 K      0        READY
22 48493  28% ( 27%|  1%)  12.4 K      1        READY
23 48494  27% ( 27%|  0%)  12.2 K      0        READY
24 48495  29% ( 28%|  0%)  12.5 K      0        READY
25 48496  30% ( 30%|  0%)  12.4 K      0        READY
26 48498  29% ( 27%|  1%)  12.6 K      0        READY
27 48517  24% ( 23%|  1%)  12.6 K      0        READY
28 48499  22% ( 21%|  0%)  12.3 K      1        READY
29 48518  31% ( 29%|  1%)  12.4 K      2        READY
30 48502  33% ( 32%|  0%)  12.5 K      0        READY

31 48514 80% ( 80%| 0%) 12.7 K 0 READY <<< CPU 31 is much busier than the rest, and will stay
busy for while with elephant flow.

32 48503  49% ( 48%|  0%)  12.4 K      0        READY
33 48507  27% ( 25%|  1%)  12.5 K      0        READY
34 48513  27% ( 25%|  1%)  12.5 K      0        READY
35 48508  32% ( 31%|  1%)  12.4 K      0        READY
36 48512  31% ( 29%|  1%)  12.4 K      0        READY
```

```
$ top

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND

69470 root       1 -19 9088m 1.0g  96m R   80  0.4 135:33.51 snort     <<<< one snort very busy,
rest below 50%

69468 root       1 -19 9089m 1.0g  99m R   49  0.4 116:08.69 snort
69467 root       1 -19 9078m 1.0g  97m S   47  0.4 118:30.02 snort
69492 root       1 -19 9118m 1.1g  97m R   47  0.4 116:40.15 snort
69469 root       1 -19 9083m 1.0g  96m S   39  0.4 117:13.27 snort
69459 root       1 -19 9228m 1.2g  97m R   37  0.5 107:13.00 snort
69473 root       1 -19 9087m 1.0g  96m R   37  0.4 108:48.32 snort
69475 root       1 -19 9076m 1.0g  96m R   37  0.4 109:01.31 snort
69488 root       1 -19 9089m 1.0g  97m R   37  0.4 105:41.73 snort
69474 root       1 -19 9123m 1.1g  96m S   35  0.4 107:29.65 snort
69462 root       1 -19 9065m 1.0g  99m R   34  0.4 103:09.42 snort
69484 root       1 -19 9050m 1.0g  96m S   34  0.4 104:15.79 snort
69457 root       1 -19 9067m 1.0g  96m S   32  0.4 104:12.92 snort
69460 root       1 -19 9085m 1.0g  97m R   32  0.4 104:16.34 snort
```

Con el algoritmo 5-Tuple descrito anteriormente, un flujo de larga duración siempre se enviará a la misma instancia de snort. Si hay amplias políticas de AVC, IPS, archivos, etc. activas en snort, la CPU puede verse alta (>80%) en una instancia de snort durante algún tiempo. La adición de la política SSL aumentará aún más el uso de la CPU, lo que repercutirá en la naturaleza costosa desde el punto de vista informático de la descifrado SSL.

El uso elevado de la CPU en pocas de las muchas CPU de resoplido no es causa de alarma crítica. Es el comportamiento del sistema NGFW al realizar una inspección profunda de paquetes en un flujo, y esto puede usar naturalmente grandes porciones de una CPU. Como pauta general, el NGFW no se encuentra en una situación crítica de escasez de CPU hasta que la mayoría de las CPU de tubo superan el 95% y permanecen por encima del 95% y se observan caídas de paquetes.

Las siguientes Remediaciones ayudarán a resolver la situación de la CPU debido a los flujos de elefantes.

Remediaciones

Desvío de aplicaciones inteligente (IAB)

La versión de software 6.0 introduce una nueva función llamada IAB. Cuando un dispositivo Firepower alcanza un umbral de rendimiento predefinido, la función IAB busca flujos que cumplan criterios específicos para eludir inteligentemente y aliviar la presión sobre los motores de detección.

> **Consejo:** Puede encontrar más información sobre la configuración del IAB aquí.

Identificación y confianza de flujos grandes

Los flujos grandes a menudo se relacionan con el uso elevado de tráfico de bajo valor de inspección, por ejemplo, copias de seguridad, replicación de bases de datos, etc. Muchas de estas aplicaciones no pueden beneficiarse de la inspección. Para evitar problemas con flujos grandes, puede identificar los flujos grandes y crear reglas de confianza de control de acceso para ellos. Estas reglas son capaces de identificar de forma única flujos grandes, permitir que estos flujos pasen sin inspeccionar y no se vean limitados por el comportamiento de una sola instancia de sonda.

> **Nota:** Para identificar flujos grandes para las reglas de confianza, comuníquese con el TAC de Cisco Firepower.

Información Relacionada

- Control de acceso mediante la omisión inteligente de aplicaciones
- Soporte Técnico y Documentación - Cisco Systems