Elaborazione di una sessione di grandi dimensioni a flusso singolo (flusso elefante) da parte dei servizi Firepower

Sommario

Introduzione

In questo documento viene illustrato il motivo per cui un singolo flusso non può utilizzare l'intero throughput nominale di un'appliance Cisco Firepower.

Premesse

Il risultato di un sito Web per il test della velocità della larghezza di banda o l'output di uno strumento per la misurazione della larghezza di banda (ad esempio, iperf) potrebbe non mostrare la velocità di trasmissione pubblicizzata degli accessori Cisco Firepower. Analogamente, il trasferimento di un file di grandi dimensioni su qualsiasi protocollo di trasporto non dimostra la velocità effettiva dichiarata di un'appliance Firepower. Si verifica perché il servizio Firepower non utilizza un singolo flusso di rete per determinare la velocità effettiva massima.

Elabora traffico per snort

La tecnologia di rilevamento sottostante del servizio Firepower è Snort. L'implementazione di Snort sull'appliance Cisco Firepower è un processo a thread singolo che ha lo scopo di elaborare il traffico. Un accessorio è classificato per una specifica classificazione in base al throughput totale di tutti i flussi che attraversano l'accessorio. Si prevede che le apparecchiature siano installate su una rete aziendale, generalmente vicino al confine e funzionanti con migliaia di connessioni.

I servizi Firepower utilizzano il bilanciamento del carico del traffico diretto a diversi processi Snort

con un processo Snort in esecuzione su ogni CPU dell'accessorio. In teoria, il carico del sistema bilancia il traffico in modo uniforme tra tutti i processi Snort. Snort deve essere in grado di fornire un'analisi contestuale appropriata per l'ispezione di Next-Generation Firewall (NGFW), Intrusion Prevention System (IPS) e Advanced Malware Protection (AMP). Per garantire la massima efficienza di Snort, tutto il traffico proveniente da un singolo flusso viene bilanciato in base al carico in una singola istanza di Snort. Se tutto il traffico proveniente da un singolo flusso non viene bilanciato in base a una singola istanza di Snort, è possibile evitare il sistema e il traffico viene suddiviso in modo che sia meno probabile che una regola Snort corrisponda o che parti di un file non siano contigue per l'ispezione AMP. Pertanto, l'algoritmo di bilanciamento del carico si basa sulle informazioni di connessione che possono identificare in modo univoco una determinata connessione.

Algoritmo a 2 tuple in ASA con servizi Firepower e NGIPS Virtual

Sull'appliance ASA (Adaptive Security Appliance) con piattaforma Firepower Service e virtuale Next-Generation Intrusion Prevention System (NGIPS), il traffico viene bilanciato in base al carico per poter ruotare con l'algoritmo a 2 tuple. I punti dati per questo algoritmo sono:

- IP di origine
- IP di destinazione

Algoritmo a 3 tuple nel software versione 5.3 o inferiore su appliance Firepower e FTD

In tutte le versioni precedenti (5.3 o precedenti), il traffico viene bilanciato in base al carico per Snort che utilizza un algoritmo a 3 tuple. I punti dati per questo algoritmo sono:

- IP di origine
- IP di destinazione
- Protocollo IP

Tutto il traffico con la stessa origine, destinazione e protocollo IP viene bilanciato per il carico nella stessa istanza di Snort.

Algoritmo a 5 tuple nel software versione 5.4, 6.0 e successive su appliance Firepower e FTD

Nella versione 5.4, 6.0 o successiva, il traffico viene bilanciato in base al carico per Snort con un algoritmo a 5 tuple. I punti dati presi in considerazione sono:

- IP di origine
- Porta di origine
- IP di destinazione
- Porta di destinazione
- Protocollo IP

Lo scopo dell'aggiunta di porte all'algoritmo è quello di bilanciare il traffico in modo più uniforme quando vi sono coppie specifiche di origine e destinazione che rappresentano grandi porzioni del traffico. Con l'aggiunta delle porte, le porte di origine effimere di ordine superiore devono essere diverse per flusso e devono aggiungere un'entropia aggiuntiva in modo più uniforme per bilanciare il traffico a diverse istanze di snort.

Throughput totale

Il throughput totale di un accessorio viene misurato in base al throughput aggregato di tutte le istanze snort che funzionano al massimo delle loro potenzialità. Le procedure standard del settore per misurare il throughput sono per più connessioni HTTP con diverse dimensioni dell'oggetto. Ad esempio, la metodologia di test NSS NGFW misura il throughput totale del dispositivo con oggetti da 44k, 21k, 10k, 4.4k e 1.7k. Questi si traducono in una gamma di dimensioni medie dei pacchetti da circa 1k & byte a 128 byte a causa degli altri pacchetti coinvolti nella connessione HTTP.

Èpossibile stimare la valutazione delle prestazioni di una singola istanza di Snort. Prendere il throughput nominale dell'accessorio e dividerlo per il numero di istanze Snort in esecuzione. Ad esempio, se un accessorio ha una velocità di 10 Gb/s per IPS con una dimensione media dei pacchetti di 1.000 byte e dispone di 20 istanze di Snort, il throughput massimo approssimativo per una singola istanza sarà di 500 Mb/s per Snort. Traffico di tipo diverso, protocolli di rete, dimensioni dei pacchetti e differenze nei criteri di sicurezza globali possono influire sul throughput osservato del dispositivo.

Risultati test strumento di terze parti

Quando si esegue il test con qualsiasi sito Web di test della velocità o qualsiasi strumento di misurazione della larghezza di banda, ad esempio iperf, viene generato un flusso TCP singolo di grandi dimensioni. Questo tipo di flusso TCP di grandi dimensioni è detto flusso elefante. Un flusso elefante è una connessione di rete a sessione singola e relativamente lunga che utilizza una quantità elevata o sproporzionata di larghezza di banda. Poiché questo tipo di flusso viene assegnato a un'istanza Snort, il risultato del test indica il throughput di una singola istanza Snort e non il valore di throughput aggregato dell'accessorio.

Sintomi osservati

CPU alta osservata

Un altro effetto visibile dei flussi di elefante può essere la cpu alta istanza snort. Questa condizione può essere rilevata tramite "show asp inspect-dp snort" o con lo strumento "top" della 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 l'algoritmo 5-Tuple descritto sopra, un flusso di lunga durata sarà sempre inviato alla stessa istanza snort. Se in uno snort sono attive numerose policy AVC, IPS, File, ecc., la CPU può essere vista alta (>80%) su un'istanza snort per un certo periodo di tempo. L'aggiunta del criterio SSL aumenterà ulteriormente l'utilizzo della CPU a causa della natura dispendiosa a livello di calcolo della decrittografia SSL.

Un elevato livello di CPU su alcune delle molte CPU snort non è una causa di allarme critico. È il

comportamento del sistema NGFW nell'eseguire l'ispezione approfondita dei pacchetti in un flusso, e questo può naturalmente usare grandi porzioni di una CPU. Come regola generale, la NGFW non si trova in una situazione critica di carenza della CPU fino a quando la maggior parte delle CPU snort sono oltre il 95% e rimangono oltre il 95% e si assiste a una diminuzione dei pacchetti.

Le correzioni riportate di seguito aiuteranno a risolvere le situazioni di elevata CPU dovute ai flussi di elefante.

Correzioni

Intelligent Application Bypass (IAB)

La versione 6.0 del software introduce una nuova funzione chiamata IAB. Quando un accessorio Firepower raggiunge una soglia di prestazioni predefinita, la funzione IAB cerca flussi che soddisfino criteri specifici in modo da ignorare in modo intelligente la pressione sui motori di rilevamento.

> **Suggerimento**: Qui sono disponibili ulteriori informazioni sulla configurazione dell'IAB.

Identificazione e attendibilità dei flussi di grandi dimensioni

I flussi di grandi dimensioni sono spesso correlati a un traffico elevato e di basso valore di ispezione, ad esempio backup, replica di database e così via. Molte di queste applicazioni non possono beneficiare di ispezioni. Per evitare problemi con i flussi di grandi dimensioni, è possibile identificare i flussi di grandi dimensioni e creare per essi regole di attendibilità per il controllo dell'accesso. Queste regole sono in grado di identificare in modo univoco i flussi di grandi dimensioni, di consentire a tali flussi di passare senza essere ispezionati e di non essere limitati dal comportamento della singola istanza snort.

> **Nota:** Per identificare flussi di grandi dimensioni per le regole di attendibilità, contattare Cisco Firepower TAC.

Informazioni correlate

- Controllo dell'accesso tramite bypass intelligente delle applicazioni
- Documentazione e supporto tecnico – Cisco Systems