Proceséén-stream grote sessie (olieleverende stroom) met FirePOWER-services

Inhoud

Inleiding

Dit document beschrijft waarom één stroom niet de volledige nominale doorvoersnelheid van een Cisco FirePOWER-apparaat kan gebruiken.

Achtergrondinformatie

Het resultaat van elke website voor het testen van de bandbreedte, of de uitvoer van een willekeurig meetgereedschap (bijvoorbeeld iperf) kan de geadverteerde doorvoerscore van de Cisco Firepower-apparaten niet tonen. Evenzo laat de overdracht van een zeer groot bestand over elk vervoerprotocol de geadverteerde doorvoercapaciteit van een FirePOWER-apparaat niet zien. Dit gebeurt omdat de Firepower Service geen enkele netwerkstroom gebruikt om de maximale doorvoersnelheid te bepalen.

Procesverkeer per Snort

De onderliggende detectietechnologie van de Firepower Service is Snort. De implementatie van Snort op het Cisco FirePOWER-apparaat is een proces met één draad om verkeer te verwerken. Een apparaat wordt beoordeeld voor een specifieke score op basis van de totale doorvoersnelheid van alle stromen die door het apparaat gaan. Verwacht wordt dat de apparaten worden ingezet op een bedrijfsnetwerk, gewoonlijk in de buurt van de grensrand en met duizenden verbindingen werken.

Firepower Services maakt gebruik van een taakverdeling van verkeer naar een aantal verschillende snelprocessen met één Snort-proces dat op elke CPU op het apparaat draait.

Idealiter verdeelt de systeembelasting het verkeer gelijkmatig over alle Snort processen. Snort moet in staat zijn om een juiste contextuele analyse te leveren voor de inspectie van de volgende generatie firewall (NGFW), inbraakpreventiesysteem (IPS) en Advanced Malware Protection (AMP). Om ervoor te zorgen dat Snort het meest effectief is, wordt al het verkeer van één enkele stroom geladen naar één enkele instantie. Als al het verkeer van één enkele stroom niet in één enkele instantie is gebalanceerd, kan het systeem ontdoken worden en zou het verkeer op een zodanige manier worden verschoven dat een enkele regel minder waarschijnlijk zal worden aangepast of dat stukken van een bestand niet contiguent zijn voor een AMP-inspectie. Daarom is het taakverdelingsalgoritme gebaseerd op de verbindingsinformatie die een bepaalde verbinding uniek kan identificeren.

2-kanaals algoritme in ASA met FirePOWER Services en NGIPS virtueel

Op de adaptieve security applicatie (ASA) met FirePOWER Service platform en Next Generation Inbraakpreventiesysteem (NGIPS) virtueel wordt het verkeer gebalanceerd om te kunnen sorteren onder het gebruik van een twee-voudig algoritme. De datapunten voor dit algoritme zijn:

- Bron IP
- IP-bestemming

3-kanaals algoritme in softwareversie 5.3 of lager op FirePOWER en FTD-applicaties

Op alle eerdere versies (5.3 of lager) wordt het verkeer gelijkmatig verdeeld over de lengte van een snor die een drie-uitgangen algoritme gebruikt. De datapunten voor dit algoritme zijn:

- Bron IP
- IP-bestemming
- IP-protocol

Elk verkeer met dezelfde bron-, bestemming- en IP-protocol wordt geladen in hetzelfde exemplaar.

5-kanaals algoritme in softwareversie 5.4, 6.0 en meer over FirePOWER- en FTD-applicaties

Bij versie 5.4, 6.0 of hoger wordt het verkeer geladen om te sorteren en in te delen met een 5-voudig algoritme. De gegevensbanken waarmee rekening wordt gehouden zijn:

- Bron IP
- Bronpoort
- IP-bestemming
- Doelpoort
- IP-protocol

Het doel om havens aan het algoritme toe te voegen is verkeer gelijkmatiger in balans te brengen wanneer er specifieke bron en bestemmingsparen zijn die voor grote delen van het verkeer verantwoordelijk zijn. Door toevoeging van de havens, moeten de hoge bestelhavens per stroom verschillend zijn, en extra entropie gelijkmatiger toevoegen die het verkeer in verschillende korte gevallen in evenwicht brengt.

Totale productie

De totale doorvoersnelheid van een apparaat wordt gemeten aan de hand van de totale doorvoersnelheid van alle korte apparaten die optimaal werken. De industrie standaardpraktijken om de doorvoersnelheid te meten zijn voor meerdere HTTP connecties met verschillende objectgrootte. De NSS NGFW testmethodologie meet bijvoorbeeld de totale doorvoersnelheid van het apparaat met 44k, 21k, 10k, 4.4k en 1.7k objecten. Deze vertalen naar een reeks gemiddelde pakketformaten van ongeveer 1k & bytes tot 128 bytes vanwege de andere pakketten die betrokken zijn bij de HTTP-verbinding.

U kunt de prestatieclassificatie van een individuele instantie van de Kort schatten. Geef de nominale doorvoersnelheid van het apparaat op en verdeel dit door het aantal uitgevoerde korte instellingen. Als een apparaat bijvoorbeeld is voorzien van een capaciteit van 10 Gbps voor IPS met een gemiddelde pakketgrootte van 10 k bytes en dat apparaat 20 exemplaren van Snort heeft, is de gemiddelde maximale doorvoersnelheid voor één exemplaar 500 Mbps per snort. Verschillende types van verkeer, netwerkprotocollen, groottes van de pakketten samen met verschillen in het algemene veiligheidsbeleid kunnen allen de waargenomen doorvoersnelheid van het apparaat beïnvloeden.

Testresultaat van derden voor tools

Wanneer u test met om het even welke website van het snelheidstest, of om het even welk bandbreedte meetgereedschap, zoals, iperf, wordt één grote enkele stroom TCP stroom gegenereerd. Dit type van grote TCP flow wordt een olifantenstroom genoemd. Een Elephant Flow is één sessie, relatief lange lopende netwerkverbinding die een grote of disproportionele hoeveelheid bandbreedte verbruikt. Dit type stroom wordt aan één instantie toegewezen, zodat het testresultaat de doorvoersnelheid van één enkele instantie weergeeft, en niet de totale doorvoersnelheid van het apparaat.

210 Ge

Waargenomen hoge CPU

Een ander zichtbaar effect van olifantenstromen kan zijn de instantie met een hoog cpu-gehalte. Dit kan worden gezien via "show asp inspect-dp snort" of met het shell "top" gereedschap.

```
> 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
```

Met 5-Tuple algoritme hierboven beschreven, zal een lange levendige stroom altijd naar dezelfde gesnorinstantie worden gestuurd. Als er uitgebreide actieve AVC, IPS, File, enz.-beleid in poorten zijn, kan de CPU gedurende een bepaalde periode hoog (>80%) in een gecompliceerde instantie worden gezien. Wanneer u SSL-beleid toevoegt, wordt het CPU-gebruik verder toegenomen naar de computationeel dure aard van SSL-decryptie.

Hoge CPU's bij weinig van de vele gesorteerde CPU's zijn geen reden voor kritiek alarm. Het is het gedrag van het NGFW-systeem bij het uitvoeren van diepe pakketinspecties in een stroom en dit kan van nature grote delen van een CPU gebruiken. Als algemeen richtsnoer bevindt de NGFW zich niet in een kritieke uitgangssituatie van de CPU's totdat de meeste ongesorteerde CPU's meer dan 95% bedragen en meer dan 95% behouden blijven en pakketdalingen worden gezien.

De onderstaande oplossingen helpen bij een situatie met een hoge CPU als gevolg van olieleveringen.

Remedities

Intelligent Application Bypass (IAB)

De softwarerelease 6.0 introduceert een nieuwe optie genaamd IAB. Wanneer een FirePOWER-apparaat een van te voren vastgestelde prestatiedrempel bereikt, zoekt de IAB-functie naar stromen die voldoen aan specifieke criteria om op een intelligente manier te omzeilen die de druk op de detectiemachines verlichten.

> **Tip**: Meer informatie over de configuratie van de IAB is hier te vinden.

Grote stromen identificeren en vertrouwen

Grote stromen zijn vaak gerelateerd aan verkeer met een lage inspectiewaarde, bijvoorbeeld back-ups, gegevensreplicatie, enzovoort. Veel van deze aanvragen kunnen niet aan een inspectie worden onderworpen. Om problemen met grote stromen te voorkomen, kunt u de grote stromen identificeren en toegangscontroleregels voor deze stromen creëren. Deze regels zijn in staat om grote stromen uniek te identificeren, deze stromen ongeinspecteerd door te laten lopen en niet te worden beperkt door het gedrag van één enkele instantie.

> Opmerking: Om grote stromen voor vertrouwensregels te identificeren, neemt u contact op met Cisco Firepower TAC.

Gerelateerde informatie

- Toegangsbeheer met behulp van intelligente toepassingsbereik
- Technische ondersteuning en documentatie – Cisco Systems