Traiter une grande session à flux unique (flux d'éléphants) par Firepower Services

Contenu

Introduction

Ce document décrit pourquoi un flux unique ne peut pas consommer l'intégralité du débit nominal d'un appareil Cisco Firepower.

Informations générales

Le résultat d'un site Web de test de la vitesse de la bande passante ou le résultat d'un outil de mesure de la bande passante (par exemple, iperf) peut ne pas présenter le débit annoncé des appliances Cisco Firepower. De même, le transfert d'un fichier très volumineux sur n'importe quel protocole de transport ne démontre pas le débit annoncé d'un appareil Firepower. Cela se produit parce que le service Firepower n'utilise pas un seul flux réseau afin de déterminer son débit maximal.

Trafic de processus par Snort

La technologie de détection sous-jacente du service Firepower est Snort. La mise en oeuvre de Snort sur l'appliance Cisco Firepower est un processus de thread unique afin de traiter le trafic. Une appliance est évaluée pour une évaluation spécifique basée sur le débit total de tous les flux qui passent par l'appliance. Les appliances devraient être déployées sur un réseau d'entreprise, généralement à proximité de la frontière et fonctionner avec des milliers de connexions.

Firepower Services utilise l'équilibrage de charge du trafic vers un certain nombre de processus

Snort différents avec un processus Snort qui s'exécute sur chaque processeur de l'appliance. Idéalement, la charge système équilibre le trafic de manière égale sur tous les processus Snort. Snort doit être en mesure de fournir une analyse contextuelle appropriée pour l'inspection du pare-feu de nouvelle génération (NGFW), du système de prévention des intrusions (IPS) et de la protection avancée contre les programmes malveillants (AMP). Afin de s'assurer que Snort est le plus efficace, tout le trafic d'un seul flux est équilibré de charge à une seule instance de snort. Si l'ensemble du trafic d'un seul flux n'était pas équilibré à une seule instance de snort, le système pourrait être évité et le trafic se répandrait de telle manière qu'une règle de Snort pourrait être moins susceptible de correspondre ou que des morceaux d'un fichier ne sont pas contigus pour l'inspection AMP. Par conséquent, l'algorithme d'équilibrage de charge est basé sur les informations de connexion qui peuvent identifier de manière unique une connexion donnée.

Algorithme à 2 boutons dans ASA avec Firepower Services et NGIPS Virtual

Sur la plate-forme ASA (Adaptive Security Appliance) avec Firepower Service et la plate-forme virtuelle NGIPS (Next Generation Intrusion Prevention System), le trafic est équilibré en charge afin de Snort avec l'utilisation d'un algorithme de 2 tuples. Les points de données de cet algorithme sont les suivants :

- Adresse IP source
- Adresse IP de destination

Algorithme 3-Tuple dans la version 5.3 ou inférieure du logiciel sur les appliances Firepower et FTD

Dans toutes les versions antérieures (5.3 ou inférieures), le trafic est équilibré en charge par Snort qui utilise un algorithme à 3 tuples. Les points de données de cet algorithme sont les suivants :

- Adresse IP source
- Adresse IP de destination
- Protocole IP

Tout trafic avec la même source, la même destination et le même protocole IP sont équilibrés de charge à la même instance de Snort.

Algorithme 5-Tuple dans les versions 5.4, 6.0 et ultérieures du logiciel sur les appliances Firepower et FTD

Sur les versions 5.4, 6.0 ou ultérieures, le trafic est équilibré en charge avec Snort avec un algorithme de 5 tuples. Les points de données pris en compte sont les suivants :

- Adresse IP source
- Port source
- Adresse IP de destination
- Destination Port (port de destination)
- Protocole IP

L'objectif de l'ajout de ports à l'algorithme est d'équilibrer le trafic de manière plus égale lorsqu'il existe des paires source et de destination spécifiques qui représentent une grande partie du trafic. En plus des ports, les ports source éphémères de haut niveau doivent être différents par flux et doivent ajouter une entropie supplémentaire plus égale qui équilibre le trafic à différentes instances de snort.

Débit total

Le débit total d'un appareil est mesuré en fonction du débit total de toutes les instances de snort qui fonctionnent au maximum de leur potentiel. Les pratiques standard du secteur pour mesurer le débit concernent plusieurs connexions HTTP avec différentes tailles d'objet. Par exemple, la méthodologie de test NSS NGFW mesure le débit total du périphérique avec des objets 44 k, 21 k, 10 k, 4,4 k et 1,7 k. Ils se traduisent par une plage de tailles moyennes de paquets allant d'environ 1 Ko et d'octets à 128 octets en raison des autres paquets impliqués dans la connexion HTTP.

Vous pouvez estimer la performance d'une instance Snort individuelle. Prenez le débit nominal de l'appliance et divisez-le par le nombre d'instances Snort qui s'exécutent. Par exemple, si une appliance est évaluée à 10 Gbit/s pour IPS avec une taille de paquet moyenne de 1 000 octets et que cette appliance possède 20 instances de Snort, le débit maximal approximatif pour une instance unique serait de 500 Mbit/s par Snort. Différents types de trafic, protocoles réseau, tailles de paquets, ainsi que des différences dans la stratégie de sécurité globale peuvent tous avoir un impact sur le débit observé du périphérique.

Résultat du test d'outil tiers

Lorsque vous effectuez un test avec un site Web de test de vitesse ou un outil de mesure de la bande passante, tel que iperf, un flux TCP à flux unique important est généré. Ce type de flux TCP important est appelé flux d'éléphants. Un flux d'éléphants est une connexion réseau à session unique, relativement longue et qui consomme une bande passante importante ou disproportionnée. Ce type de flux est attribué à une instance Snort. Par conséquent, le résultat du test affiche le débit d'une instance Snort unique, et non le débit agrégé de l'appliance.

Symptômes observés

CPU élevé observé

Un autre effet visible de Elephant Flows peut être snort instance high cpu. Ceci peut être vu via « show asp inspect-dp snort », ou avec l'outil « top » du 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
```

Avec l'algorithme 5-Tuple décrit ci-dessus, un flux de longue durée sera toujours envoyé à la même instance de snort. Si des stratégies AVC, IPS, File, etc étendues sont actives dans le snort, le CPU peut être vu haut (>80%) sur une instance de snort pendant un certain temps. L'ajout d'une stratégie SSL augmentera encore l'utilisation du CPU à la nature coûteuse du déchiffrement SSL.

Une CPU élevée sur quelques-uns des nombreux processeurs sourds n'est pas une cause d'alarme critique. C'est le comportement du système NGFW lors de l'inspection approfondie des paquets dans un flux, et cela peut naturellement utiliser de grandes parties d'un processeur. En

règle générale, le pare-feu de nouvelle génération n'est pas dans une situation critique de pénurie de CPU tant que la plupart des processeurs sourds ne dépassent pas 95 % et restent supérieurs à 95 % et que des pertes de paquets ne sont pas observées.

Les corrections ci-dessous vous aideront à gérer une situation CPU élevée due aux flux Elephant.

Corrections

Contournement intelligent des applications

La version 6.0 du logiciel introduit une nouvelle fonctionnalité appelée IAB. Lorsqu'un appareil Firepower atteint un seuil de performances prédéfini, la fonction IAB recherche les flux qui répondent à des critères spécifiques afin de contourner intelligemment les contraintes sur les moteurs de détection.

> **Astuce** : Vous trouverez plus d'informations sur la configuration du CCI ici.

Identification et confiance des flux importants

Les flux importants sont souvent liés à un trafic à faible valeur d'inspection à forte utilisation, par exemple, les sauvegardes, la réplication de bases de données, etc. Bon nombre de ces applications ne peuvent pas bénéficier d'une inspection. Afin d'éviter les problèmes avec les flux volumineux, vous pouvez identifier les flux volumineux et créer des règles d'approbation de contrôle d'accès pour eux. Ces règles sont capables d'identifier de manière unique les flux importants, de permettre à ces flux de passer sans inspection, et de ne pas être limité par le comportement d'une seule instance de snort.

> **Note**: Afin d'identifier les flux importants pour les règles de confiance, contactez le TAC Cisco Firepower.

Informations connexes

- Contrôle d'accès à l'aide du contournement d'application intelligent
- Support et documentation techniques - Cisco Systems