Verarbeitung einer Single-Stream-Sitzung (Elephant Flow) durch FirePOWER Services

Inhalt

Einführung

In diesem Dokument wird erläutert, warum ein einzelner Datenfluss nicht den gesamten angegebenen Durchsatz einer Cisco FirePOWER-Appliance verbrauchen kann.

Hintergrundinformationen

Das Ergebnis einer Testwebsite für Bandbreitengeschwindigkeiten oder die Ausgabe eines Tools zur Bandbreitenmessung (z. B. iperf) weist möglicherweise nicht die angegebene Durchsatzrate der Cisco FirePOWER-Appliances auf. Ebenso zeigt die Übertragung einer sehr großen Datei über ein Transportprotokoll nicht die angegebene Durchsatzrate einer FirePOWER-Appliance. Sie tritt auf, weil der FirePOWER-Dienst keinen einzelnen Netzwerkfluss verwendet, um den maximalen Durchsatz zu bestimmen.

Datenverkehr nach Snort verarbeiten

Die dem FirePOWER-Dienst zugrunde liegende Erkennungstechnologie ist Snort. Die Implementierung von Snort auf der Cisco FirePOWER-Appliance ist ein einzelner Threadprozess zur Verarbeitung von Datenverkehr. Eine Appliance wird auf der Grundlage des Gesamtdurchsatzes aller Datenflüsse, die die Appliance durchlaufen, für eine bestimmte Bewertung bewertet. Es wird erwartet, dass die Appliances in einem Unternehmensnetzwerk bereitgestellt werden, in der Regel in der Nähe des Grenz-Edge und mit Tausenden von Verbindungen.

FirePOWER Services verwenden Lastenausgleich des Datenverkehrs für eine Reihe verschiedener Snort-Prozesse mit einem Snort-Prozess, der auf jeder CPU der Appliance ausgeführt wird. Im Idealfall gleicht die Systemlast den Datenverkehr gleichmäßig über alle Snort-Prozesse hinweg aus. Snort muss in der Lage sein, angemessene Kontextanalysen für die Next-Generation Firewall (NGFW)-, Intrusion Prevention System (IPS)- und Advanced Malware Protection (AMP)-Inspektion bereitzustellen. Um sicherzustellen, dass Snort am effektivsten ist, wird der gesamte Datenverkehr eines einzelnen Datenflusses Load Balancing auf eine einzige Instanz durchgeführt. Wenn der gesamte Datenverkehr eines einzelnen Datenflusses nicht auf eine einzelne Instanz verteilt wurde, könnte das System umgangen werden, und der Datenverkehr würde so auslaufen, dass eine Snort-Regel möglicherweise weniger übereinstimmt oder Teile einer Datei nicht für die AMP-Prüfung zusammenhängen. Daher basiert der Lastenausgleichsalgorithmus auf den Verbindungsinformationen, die eine bestimmte Verbindung eindeutig identifizieren können.

2-Tuple-Algorithmus in ASA mit FirePOWER Services und NGIPS Virtual

Auf der virtuellen Adaptive Security Appliance (ASA) mit der FirePOWER Service-Plattform und dem Next Generation Intrusion Prevention System (NGIPS) wird der Datenverkehr mit Lastausgleich ausgeglichen, um Snort mithilfe eines 2-Tupel-Algorithmus zu ermöglichen. Die Datenpunkte für diesen Algorithmus sind:

- Quell-IP
- Ziel-IP

3-Tuple-Algorithmus in Softwareversion 5.3 oder niedriger für FirePOWER- und FTD-Appliances

Bei allen vorherigen Versionen (5.3 oder niedriger) wird der Datenverkehr nach Snort mit einem 3-Tupel-Algorithmus ausgeglichen. Die Datenpunkte für diesen Algorithmus sind:

- Quell-IP
- Ziel-IP
- IP-Protokoll

Datenverkehr mit derselben Quelle, demselben Ziel und demselben IP-Protokoll wird für dieselbe Instanz von Snort Load Balancing verwendet.

5-Tuple-Algorithmus in den Softwareversionen 5.4, 6.0 und höher für FirePOWER- und FTD-Appliances

In Version 5.4, 6.0 oder höher wird der Datenverkehr mit einem 5-Tupel-Algorithmus auf Snort verteilt. Folgende Datenpunkte wurden berücksichtigt:

- Quell-IP
- Quell-Port
- Ziel-IP
- Zielport
- IP-Protokoll

Der Zweck, dem Algorithmus Ports hinzuzufügen, besteht darin, den Datenverkehr gleichmäßig auszugleichen, wenn bestimmte Quell- und Zielpaare vorhanden sind, die einen großen Teil des Datenverkehrs ausmachen. Zusätzlich zu den Ports müssen die ephemeren Quell-Ports mit hoher

Reihenfolge je Datenfluss unterschiedlich sein und zusätzliche Entropie gleichmäßiger hinzufügen, um den Datenverkehr auf verschiedene Instanzen zu verteilen.

Gesamtdurchsatz

Der Gesamtdurchsatz einer Appliance wird anhand des Gesamtdurchsatzes aller Instanzen gemessen, die ihr volles Potenzial ausschöpfen. Branchenübliche Verfahren zur Messung des Durchsatzes sind für mehrere HTTP-Verbindungen mit verschiedenen Objektgrößen vorgesehen. Die NSS NGFW-Testmethodik misst beispielsweise den Gesamtdurchsatz des Geräts mit 44.000, 21.000, 10.000, 4.400 und 1.700 Objekten. Diese Pakete übersetzen aufgrund der anderen Pakete, die an der HTTP-Verbindung beteiligt sind, eine durchschnittliche Paketgröße von etwa 1.000 Byte bis zu 128 Byte.

Sie können die Leistungsbewertung einer einzelnen Snort-Instanz schätzen. Teilen Sie den Nenndurchsatz der Appliance durch die Anzahl der Snort-Instanzen, die ausgeführt werden. Wenn beispielsweise eine Appliance für IPS mit einer durchschnittlichen Paketgröße von 1.000 Byte auf 10 Gbit/s bewertet wird und diese Appliance über 20 Instanzen von Snort verfügt, beträgt der ungefähre maximale Durchsatz für eine Instanz 500 Mbit/s pro Snort. Unterschiedliche Arten von Datenverkehr, Netzwerkprotokolle, Paketgrößen sowie Unterschiede in den allgemeinen Sicherheitsrichtlinien können sich auf den beobachteten Durchsatz des Geräts auswirken.

Testergebnisse von Drittanbietern

Beim Testen mit einer beliebigen Website für Geschwindigkeitstests oder mit einem beliebigen Bandbreitenmessungstool, wie z. B. iperf, wird ein großer TCP-Datenstrom mit einem Stream generiert. Dieser große TCP-Fluss wird als Elephant Flow bezeichnet. Ein Elephant Flow ist eine einzelne Sitzung, eine relativ lange Netzwerkverbindung, die eine große oder unverhältnismäßige Menge an Bandbreite beansprucht. Dieser Flow-Typ wird einer Snort-Instanz zugewiesen, daher zeigt das Testergebnis den Durchsatz einer einzelnen Snort-Instanz an, nicht die aggregierte Durchsatzrate der Appliance.

Beobachtete Symptome

Beobachtete hohe CPU

Ein weiterer sichtbarer Effekt von Elephant Flows kann die schnelle, instanzhohe CPU sein. Dies ist über "show asp inspect-dp snort" oder mit der Shell "top" zu sehen.

```
> 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
```

Mit dem oben beschriebenen 5-Tuple-Algorithmus wird immer ein langer Datenfluss an dieselbe Snort-Instanz gesendet. Wenn umfassende AVC-, IPS-, Datei- usw. Richtlinien aktiv sind, kann die CPU für einen bestimmten Zeitraum hoch (>80 %) auf einer kurzen Instanz angezeigt werden. Durch das Hinzufügen von SSL-Richtlinien wird die CPU-Auslastung weiter erhöht, da SSL-

Entschlüsselung rechnerisch kostspielig ist.

Eine hohe CPU auf einigen der vielen Snort-CPUs ist keine Ursache für kritische Alarme. Das NGFW-System führt Deep Packet Inspection (Deep Packet Inspection) in einem Datenfluss durch, wobei natürlich große Teile einer CPU verwendet werden können. Generell befindet sich die NGFW erst dann in einer kritischen CPU-Auslastung, wenn die meisten kleinen CPUs über 95 % und über 95 % liegen und Paketverluste auftreten.

Die folgenden Problembehebungen helfen bei einer hohen CPU-Situation aufgrund von Elephant-Strömen.

Problembehebung

Intelligentes Umgehen von Anwendungen (IAB)

Mit der Softwareversion 6.0 wird eine neue Funktion namens IAB eingeführt. Wenn eine FirePOWER-Appliance einen vordefinierten Leistungsschwellenwert erreicht, sucht die IAB-Funktion nach Strömen, die bestimmte Kriterien erfüllen, um den Druck auf die Erkennungs-Engines intelligent zu umgehen.

> **Tipp**: Weitere Informationen zur Konfiguration des IAB finden Sie hier.

Identifizierung und Vertrauen großer Datenflüsse

Große Datenflüsse sind häufig auf Datenverkehr mit geringem Überprüfungswert zurückzuführen, z. B. Datensicherungen, Datenbankreplikation usw. Viele dieser Anwendungen können nicht von der Überprüfung profitieren. Um Probleme mit großen Datenflüssen zu vermeiden, können Sie die großen Datenflüsse identifizieren und Zugriffskontroll-Vertrauensregeln für diese erstellen. Diese Regeln sind in der Lage, große Datenflüsse eindeutig zu identifizieren, die ungeprüfte Übertragung dieser Datenflüsse zu ermöglichen und nicht durch das Verhalten einer einzelnen Snort-Instanz beschränkt zu sein.

> **Hinweis**: Um große Datenflüsse für Vertrauensregeln zu identifizieren, wenden Sie sich an das Cisco FirePOWER TAC.

Zugehörige Informationen

- Zugriffskontrolle mit intelligenter Anwendungsumgehung
- Technischer Support und Dokumentation - Cisco Systems