进程单个流大会话(大象流)由Firepower服务

Contents

[Introduction](#)
[背景信息](#)
[由喷鼻息的进程数据流](#)
[在虚拟ASA与Firepower服务和的NGIPS的2元组算法](#)
[在软件版本5.3的3元组算法或在Firepower和FTD工具降低](#)
[在软件版本5.4， 6.0和极大的5元组算法在Firepower和FTD工具](#)
[总吞吐量](#)
[第三方工具测试结果](#)
[被观察的症状](#)
[被观察的高CPU](#)
[补救](#)
[智能应用程序旁路(IAB)](#)
[识别并且委托大流](#)
[Related Information](#)

Introduction

本文描述单个流为什么不能消耗Cisco Firepower工具的整个额定吞吐量。

背景信息

测试网站的任何带宽速度的结果或者任何带宽测量工具的输出(例如， iperf)也许不陈列Cisco Firepower工具的做通告的吞吐量规定值。同样地，一个非常大文件的转移过渡任何传输协议不展示Firepower工具的做通告的吞吐量规定值。因为Firepower服务不使用单个网络流为了确定其最大吞吐量，它发生。

由喷鼻息的进程数据流

Firepower服务的基础检测技术是喷鼻息。喷鼻息的实施在Cisco Firepower工具上的是单个线程进程为了处理数据流。工具为根据通过工具的总吞吐量的特定规定值是额定的所有流。在边界边缘附近通常预计工具配置在一个公司网络，并且与千位连接一起使用。

Firepower服务数据流使用负载均衡对一定数量的另外喷鼻息进程的与在工具上的每个CPU运行的一个喷鼻息进程。理论上来讲，系统负载在所有喷鼻息进程间均匀地平衡数据流。喷鼻息需要能为下一代防火墙(NGFW)，入侵防御系统(IPS)和先进的Malware保护(AMP)检查提供适当的上下文分析。为了保证喷鼻息是最有效的，从单个流的所有数据流是负荷被均衡对一个喷鼻息实例。 如果从单个流的所有数据流未被平衡对单个喷鼻息实例，系统可能被逃避，并且数据流会溢出，在这种情况下喷鼻息规则也许是不太可能匹配或文件的部分为AMP检查不是连续。所以，负载均衡算法根据能独特识别特定连接的连接信息。

在虚拟ASA与Firepower服务和的NGIPS的2元组算法

在可适应的安全工具(ASA)上与Firepower虚拟服务平台和下一代的入侵防御系统(NGIPS)，数据流是被均衡的负荷为了打鼾与使用2元组算法。此算法的datapoints是：

- 来源IP
- 目的地IP

在软件版本5.3的3元组算法或在Firepower和FTD工具降低

在所有以前的版本(5.3或降低)，数据流是使用3元组算法被均衡的负荷打鼾。此算法的datapoints是：

- 来源IP
- 目的地IP
- IP协议

与同一个来源的所有数据流，目的地和IP协议是负荷被均衡对喷鼻息同一实例。

在软件版本5.4， 6.0和极大的5元组算法在Firepower和FTD工具

在版本5.4， 6.0或者更加极大，数据流是balaned的负荷打鼾与5元组算法。被考虑到的datapoints是：

- 来源IP
- 源端口
- 目的地IP
- 目的地端口
- IP协议

目的添加端口到算法将均匀平衡数据流，当有占数据流的大部分的特定源及目的地对时。由端口的添加，高位短暂源端口一定是不同的每流，并且必须均匀地添加平衡数据流对不同的喷鼻息实例的另外的熵。

总吞吐量

工具的总吞吐量被测量根据工作对他们的全部潜在的总吞吐量所有喷鼻息实例。工业标准运作为了测量吞吐量是为与多种对象大小的多个HTTP连接。例如， NSS NGFW测试方法测量设备的总吞吐量有44k、21k、10k、4.4k和1.7k对象的。这些转换为平均信息包大小&字节的范围从在1k附近的为128个字节由于在HTTP连接涉及的其他信息包。

您能估计一个单个喷鼻息实例的业绩评级。采取工具的额定吞吐量并且由的喷鼻息实例的数量运行分开那。例如，如果工具是额定的在IPS的10Gbps与平均信息包大小1k字节和该工具有喷鼻息20个实例，单个实例的近似最大吞吐量是500 Mbps每个喷鼻息。不同类型的流量，网络协议，信息包的大小与区别一起的在整体安全策略上能所有影响设备的被观察的吞吐量。

第三方工具测试结果

当您用所有速度测试网站，或者任何带宽测量工具测试，例如， iperf，一大单个流TCP流时生成。此种大TCP流称为大象流。大象流是单个会话，相对消耗带宽一个大或不成比例的数量的长期的网络连接。此种流分配到一个喷鼻息实例，因此测试结果显示吞吐量单个喷鼻息实例，工具的不是总吞吐量规定值。

被观察的症状

被观察的高CPU

大象流另一可视效果可以是喷鼻息实例高CPU。这能被看到通过“显示asp Inspect DP喷鼻息”，或者用shell “顶层”工具。

```
> 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
```

当5元组算法如上所述，长期活的流永远将被发送到同一个喷鼻息实例。如果有广泛的AVC、IPS、文件等等策略活动在喷鼻息， CPU在喷鼻息实例能被看到高(>80%)在某段时期。添加SSL策略进一步增加CPU使用情况将做对SSL解密的计算上消耗大的本质。

在少量的高CPU许多喷鼻息CPU不是严重告警的一个起因。它是NGFW系统的工作情况在实行的深度信息包检验的到流，并且这能自然使用CPU的大部分。作为总指导大纲， NGFW不在一个重要CPU缺乏情况，直到多数喷鼻息CPU 95%并且保持95%，并且信息包丢弃被看到。

下面补救将帮助以高CPU情况由于大象流。

补救

智能应用程序旁路(IAB)

软件版本6.0介绍称为IAB的一个新功能。当Firepower工具达到一个预定义的性能门限值时， IAB功能寻找满足特定标准为了智能绕过缓和在检测引擎的压力的流。

> **提示**：可以找到关于IAB的配置的更多信息这里。

识别并且委托大流

大流经常与高使用低检查值数据流，备份、例如数据库复制等等有关。许多这些应用程序不可能从检查有益于。为了避免大流的问题，您能识别大流和创建访问控制他们的信任规则。这些规则能独特识别大流，允许那些流通过未经检查和不由单个喷鼻息实例工作情况限制。

> Note:为了识别信任规则的大流，请与Cisco Firepower TAC联系。

Related Information

- 访问控制使用智能应用程序旁路

- Technical Support & Documentation - Cisco Systems