Processing of Single Stream Large Session (Elephant Flow) by the Firepower Services

Contents

Introduction

The result of any bandwidth speed testing website, or the output of any bandwidth measurement tool (for example, *iperf*) may not exhibit the advertised throughput rating of the Cisco Firepower appliances. Similarly, the transfer of a very large file over any transport protocol does not demonstrate the advertised throughput rating of a Firepower appliance. It occurs because the Firepower service does not use a single network flow to determine its maximum throughput. This document describes why a single flow cannot consume the entire rated throughput of a Cisco Firepower appliance.

Contributed by Nazmul Rajib, and Foster Lipkey, Cisco TAC Engineers.

Processing of Traffic by Snort

The underlying detection technology of the Firepower service is Snort. The implementation of Snort on the Cisco Firepower appliance is a single thread process for traffic processing. An appliance is rated for a specific rating based on the total throughput of all flows going through the appliance. It is expected that the appliances are deployed on a Corporate network, usually near the border edge and works with thousands of connections.

Firepower Services uses load balancing of traffic to a number of different Snort process with one Snort process running on each CPU on the appliance. Ideally, the system load balances traffic evenly across all of the Snort processes. Snort needs to be able to provide proper contextual analysis for NGFW, IPS, and AMP inspection. To ensure Snort is most effective, all traffic from a single flow is load balanced to one snort instance. If all traffic from a single flow was not balanced to a single snort instance, the system could be evaded by splitting the traffic in such a way that a Snort rule may be less likely to match or pieces of a file are not contiguous for AMP inspection. Therefore, the load balancing algorithm is based on connection information that can uniquely identify a given connection.

2-Tuple Algorithm in ASA with FirePOWER Services and NGIPS Virtual

On the ASA with FirePOWER Service platform and NGIPS virtual, traffic is load balaned to Snort using a 2-tuple algorithm. The datapoints for this algorithm are:

- Source IP
- Destination IP

3-Tuple Algorithm in Software Version 5.3 or Lower on Firepower and FTD appliances

On all prior Versions (5.3 or lower), traffic is load balaned to Snort using a 3-tuple algorithm. The datapoints for this algorithm are:

- Source IP
- Destination IP
- IP Protocol

Any traffic with the same source, destination, and IP Protocol are load balanced to the same instance of Snort.

5-Tuple Algorithm in Software Version 5.4, 6.0, and Greater on Firepower and FTD appliances

On Version 5.4, 6.0 or greater, traffic is load balaned to Snort using a 5-tuple algorithm. The datapoints that are taken into account are shown below:

- Source IP
- Source Port
- Destination IP
- Destination Port
- IP Protocol

The purpose of adding ports to the algorithm is to balance traffic more evenly when there are specific source and destination pairs that account for large portions of the traffic. By adding the ports, the high order ephemeral source ports should be different per flow, and should add additional entropy more evenly balancing traffic to different snort instances.

Total Throughput

The total throughput of an appliance is measured based on the aggregate throughput of all the snort instances working to their fullest potential. Industry standard practices for measuring throughput are for multiple HTTP connections using various object sizes. For example, the NSS NGFW test methodology measures total throughput of the device using 44k, 21k, 10k, 4.4k, and 1.7k objects. These translate to a range of average packet sizes from around 1k bytes to 128 bytes because of the other packets involved in the HTTP connection.

You can estimate the performance rating of an individual Snort instance by taking the rated throughput of the appliance and dividing that by the number of Snort instances that are running. For example, if an appliance is rated at 10Gbps for IPS with an average packet size of 1k bytes,and that appliance has 20 instances of Snort, the approximate maximum throughput for a

single instance would be 500 Mbps per Snort. Different types of traffic, network protocols, sizes of the packets along with differences in the overall security policy can all impact the observed throughput of the device.

Test Result of a Third Party Tool

When you test with any speed testing website, or any bandwidth measurement tool, such as, *iperf*, one large single stream TCP flow is generated. This type of large TCP flow is called an **Elephant Flow**. An Elephant Flow is a single session, relatively long running network connection that consumes a large or disproportionate amount of bandwidth. This type of flow is assigned to one Snort instance, therefore the test result displays the throughput of single snort instance, not the aggregate throughput rating of the appliance.

Remediations

Intelligent Application Bypass (IAB)

The software version 6.0 introduces a new feature called **Intelligent Application Bypass** (IAB). When a Firepower appliance reaches a pre-defined performance threshold, the IAB feature looks for flows that meet specific criteria to intelligently bypass that alleviates pressure on the detection engines.

> **Tip**: More information on configuring the IAB can be found here.

Identify and Trust the Large Flows

Large flows are often related to high use low inspection value traffic for example, backups, database replication, etc. Many of these applications may not be benefited from inspection. To avoid issues with large flows, you can identify the large flows and create Access Control trust rules for them. These rules are able to uniquely identify large flows, allow those flows to pass uninspected, and not to be limited by the single snort instance behavior.

> **Note**: To identify large flows for trust rules, please contact the Cisco Firepower TAC.

Related Documents

- Access Control Using Intelligent Application Bypass