**Deployment Plan of Martini Applications**

Chen Sun, Shuhe Wang

2018-06-18

Based on our considerations on application classification and current Huawei topology, we consider deploying 10 applications that fall into 5 categories, as listed below.

We use python with scapy to send and receive pkts, the scripts is stored at /home/shuhe

1. TCP SYN flood

* Function: Drop extra SYN packets above threshold in each measurement window
* Test traffic: SYN packets of different speed
* Test topology: Server #1 –> Tofino –> Server #2
* Test topology has changed to: Server204 -> Switch 6810 -> Tofino -> Switch6810 -> Server204
* Expected effect: As the rate of SYN packets increase, the received SYN packet rate by Server #2 increases until it reaches the pre-configured threshold.
* Since the DPDK pkt generator can’t be used, so we can’t adjust packet rate and change the content of packets, we make an adjustment: we send finite pkts to the switch, and ignore the measurement window, only check if pkts will be dropped when accumulative pkt number is over threshold. If the packets are dropped, then it proves this management can function correctly.

1. Heavy hitter / Microburst

* Function: Detect flows whose bytes sent in one measurement window exceeds threshold, and per-packet port balancing detected heavy flows / microbursts
* Test traffic: TCP flows of different sizes (two flow, one above threshold, one below threshold, in enough to demonstrate the effect.
* Test topology: Server #1, Server #2, and Server #3 are connected with Tofino
* Test topology has changed to: Server204 -> Switch 6810 -> Tofino –>Switch6810-> Server204 or Tofino–>Switch5810-> Server205
* Expected effect: Server #1 sends two flows. Only one of Server #2 and Server #3 could receive packets of the small flow, while both Server #2, 3 could receive packets of the heavy flow
* We ignore the measurement window ，send finite packets and set a low threshold to see if this work can separate packet flows to different ports when accumulative pkt number is over threshold. If both server receive packets, then it confirms that this management can function correctly.

1. Superspreader / Port scan / DDoS / TCP incast

* Function: all above four applications follow the same traffic patterns of either one-to-many or many-to-one. Take superspreader as an example. it detects IPs that contact more than a threshold of unique destinations and drop them
* Test traffic: some source IPs that contact < threshold unique destination IPs, while the others that contact > threshold unique destinations IPs.
* Test topology: Server #1 –> Tofino –> Server #2
* Test topology has changed to: Server204 -> Switch 6810 -> Tofino -> Switch6810 -> Server204
* Expected effect: Server #1 send all packets to Server #2. Tofino can detect the superspreader IP
* Expected effect becomes: the receiver will see the receiving number of each flow, and each of the number should be no more than the threshold.

1. DNS reflection attack

* Function: it detects DNS responses without corresponding requests, and drops the illegal requests
* Test traffic: DNS requests, corresponding responses, and illegal responses
* Test topology: Server #1 –> Tofino –> Server #2
* Test topology has changed to: Server204 -> Switch 6810 -> Tofino -> Switch6810 -> Server204
* Expected effect: Only legal DNS responses could reach Server #2.
* Expected effect becomes: Server 204 will only receive requests and legal responses.

1. FTP monitoring / Stateful firewall

* Function: Above two functions are similar. Take FTP monitoring as an example. It detects FTP data channel setup requests when their control channels are not established → drops the requests
* Test traffic: FTP control flows + FTP data flows, and some FTP data flows without corresponding control flows.
* Test topology has changed to: Server204 -> Switch 6810 -> Tofino -> Switch6810 -> Server204
* Test topology: Server #1 –> Tofino –> Server #2
* Expected effect: all data flows without corresponding control flows are dropped. The others are received.