Using Packet Symmetry to Curtail Malicious Traffic

Christian Kreibich
christian.kreibich@cl.cam.ac.uk

Andrew Warfield       Jon Crowcroft

Steven Hand           Ian Pratt
The “Bad Traffic” Problem

* Malicious traffic abounds on the Internet
  * Scans, DDoS, botnets, spam, etc...
* So what exactly is malicious traffic?
  * It's anomalous
  * It's often high-volume
* Bellovin was right: we really want the Evil-Bit!
  * A simple, immediate characteristic
    * That allows, denies, or at least limits atypical behaviour at the net ingress
* And use it proactively!
  * Reactive responses to DDoS are too slow and complicated
Packet Symmetry

* At the packet level, most flows are roughly symmetric
  * Well-behaved flows do see bidirectional traffic
  * For \( n > 0 \) packets sent you get \( m > 0 \) packets back within a reasonable interval
  * Response traffic is a receiver consent signal!
  * Easy to measure and enforce at the source
  * Remarkably robust
  * And surprisingly universal
A Metric for Symmetry

\[ S = \ln \left( \frac{tx + 1}{rx + 1} \right) \]

- Zero for \( tx = rx \), symmetric around it
- Remains tractable as asymmetry grows
- Note: \( tx \) and \( rx \) are **packet counts**, not byte counts
- Needs to be measured near transmitter to avoid
  - potential path asymmetry
  - source identification difficulty (NAT, spoofing)
A Penalty for Asymmetry

* Delay grows exponentially with asymmetry
* Delay, then drop

Illustration of asymmetry-based rate limiting

Asymmetry
Delay

Unacknowledged packets transmitted

Delay (ms)
Let's give it a try

* Linux netfilter/iptables, libipq

* We fixed a threshold $S = 2$
  * Asymmetry of 8:1 – quite liberal

* If $S > 2$
  * Start outstanding-packet counter $n$
  * Delay $n$th subsequent packet by 2 ms

* If $S$ goes below 2, decay delay back to zero

* Let’s see some data
UDP Flood

Unacknowledged UDP flood of 1KB packets (no limiting)

Packet Rate (K packets/s)

Asymmetry

Elapsed Time (s)
A UDP Flood, no more
TCP is symmetric
Host-based Symmetry

Cumulative host packet symmetry, 170K hosts, varying windows

- 24h window
- 1h window
- 1min window
- 1s window

Percentage of hosts vs. $\log\left(\frac{t+1}{R_t+1}\right)$
Host-pair Symmetry

Cumulative host pair symmetry for responding hosts, 360K pairs, 60s window
Flow-based Symmetry

Cumulative maximum TCP flow asymmetry, 55,000 flows, 60s window

- Red: Ignoring no packets
- Green: Ignoring first 10 packets
- Blue: Ignoring first 100 packets

Percentage of flows

$log(\frac{tx+1}{rx+1})$
UDP Flow Symmetry

Cumulative maximum UDP flow asymmetry, 4400 flows, 60s window

Percentage of flows

$log(\frac{tx+1}{rx+1})$

- Red: Ignoring no packets
- Green: Ignoring first 10 packets
- Blue: Ignoring first 100 packets
Evasive Manoeuvres

* “Fly under the radar” attacks
  * Reasonably sensitive threshold would make current DDoS levels much harder

* Botnet collusion is a tricky problem
  * Source address spoofing
    * Increasingly hard with deployed ingress filtering
    * For best effect, apply combined
  * IP ID prevents cross-traffic, unless randomised
  * Bots need to do TTL estimation

* We can raise the bar so things get significantly harder for the bad guys
Deployment Considerations

* Part of Xen toolkit
* Server farm mindset
  * Dangerous source potential
  * Deployment instantly benefits operators
* Could be put in NIC
* Michael Dales (Intel) designed it into his optical switch port controller (Xilinx), 200 lines VHDL
* Also possible in ADSL DSLAM equipment
A Principle

I will make all my traffic symmetric.
I will make all my traffic symmetric.
I will make all my traffic symmetric.
I will make all my traffic symmetric.
I will make all my traffic symmetric.
I will make all my traffic symmetric.
I will make all my traffic symmetric.
Summary

* We propose a traffic shaper that is

  simple, adaptive, always-on, edge-located,
  packet-symmetry driven, ingress-applied

* Symmetry. It's a Good Thing. (TM)

* Left as an exercise for the authors:
  * State vs. accuracy/asymmetry tradeoffs?
  * Problematic traffic? (Certain protocols, RSTs, etc)
  * Second-level effects, e.g. on traffic matrices

* Real deployment planned
  * Cambridge students = lab rats

* Questions?