www.cmand.org

Adversarial TCP: An Offensive TCP Stack to Penalize Abusive Connections

Ryan Craven, Kristina Foster, Robert Beverly {rcraven,kmfoster,rbeverly}@nps.edu

Naval Postgraduate School

Motivation

Penalize abusive hosts, spam bots, DoS attacks, scam infrastructure, etc. Cause suspected abusive connections to:

- Send more traffic
- Consume more bandwidth/time
- Induce more congestion
- Be more visible (bandwidth, congestion, $$, etc.)

Prior Work

- TCP "tarpits" to artifically slow abusive connections (we aim to do the opposite)
- Exploiting traffic congestion characteristics of abusive hosts (often bots with asymmetric bandwidth)

Hypothesis

An "adversarial" TCP stack (A-TCP) can cause a remote TCP to perform more work.

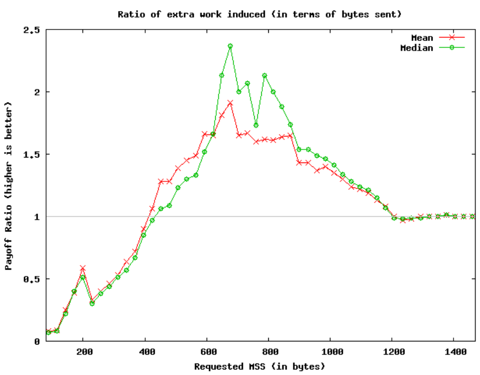
Questions

Initial research highlights interesting questions:

- How to induce extra work?
- Metric of work: packets, bytes, time, etc?
- Ratio of extra work performed by A-TCP versus induced remote work?
- Differences in A-TCP's effects against various operating systems?
- Can abusive hosts distinguish between normal and A-TCP?

Approach 1: TCP MSS

- **Idea**: reduce the advertised maximum segment size (MSS)
- Abusive host sends more packets with less data per packet = higher header overhead
- Higher header overhead = more work
- Hook TCP via iptables NFQUEUE bindings
- Scapy script overwrites MSS in SYN-ACK



Experiment

- Isolated test-bed with real hardware, different OS, dummynet, etc.
- 60 runs of 8MB transfer at different A-TCP MSS
- Different A-TCP loss rates to trigger fast-rexmit
- Define "Asynchronous Payoff Ratio" (APR):

$S^{TCP}(N)$ = TCP bytes xmit'd to send N byte data

$R^{TCP}(N)$ = TCP bytes xmit'd to recv N byte data

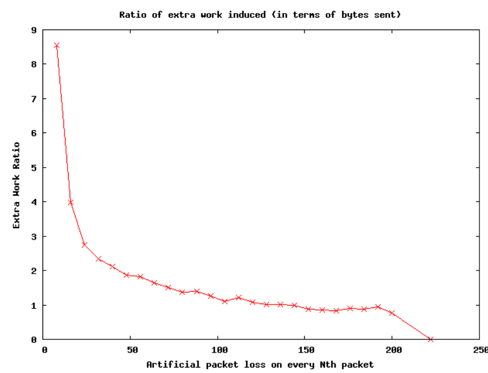
$$APR = \frac{S^{ATCP}(N) - S^{TCP}(N)}{R^{ATCP}(N) - R^{TCP}(N)} = \frac{\text{Attacker extra bytes}}{\text{A-TCP extra bytes}}$$

Early Results

- Significant OS differences (e.g. Win7 MSS)
- Large feasible MSS range with APR>2
- MSS<400 requires extra ACKs leading to APR<1
- A-TCP artificial loss + fast rexmit can produce large APR – challenge is congestion window
- We believe order of magnitude higher APRs possible – subject of our current research

Approach 2: RFC2581

- **Idea**: fake loss and induce remote side fast-retransmit/fast-recovery
- Abusive host must retransmit lost data or entire outstanding window = work
- Challenge is to prevent remote TCP from fully closing congestion window
- Remote TCP cannot differentiate real packet loss from A-TCP's artificial loss



Supported by: Cisco Systems and NSF (OCI-1127506)