**Mitigation Techniques**

**Ritesh Kumar**

---

**Introduction**

The “Worm Space”

Network as a Victim

TCP Congestion Control

---

**Network as Victim**

- **Protocol Attacks**
  - Misbehaving Receiver Attack
  - The Shrew Attack

- **Packet Flooding attacks**
  - DoS on Network (TCP Congestion Control)
  - DoS on Hosts (Under-provisioned Host)

---

**TCP Congestion Control**

![Diagram showing TCP Congestion Control](image)

- **DropTail Queueing**
  - Instantaneous Queue length
  - Average Queue length
  - Probabilistic Dropping
  - The Queue drops packets
  - TCP treats packet drops as a congestion signal

- **RED Queueing**
  - End to End Latencies are high
  - Synchronization of flows

---

**Mitigation Techniques**

**Ritesh Kumar**
**RED Queueing**

- High bitrate flows
  Punished by proportionally dropping more packets.
- Responsive flows
  Punished by signalling more congestion events.
- Mixed flows
  Attack traffic (High Bitrate) + Benign Traffic (Responsive)
  Responsive traffic suffers the most.

**Attacking TCP**

Weaknesses in TCP
Previous Related Presentations

**How to Attack TCP?**

- Usage of network traffic which is **Unresponsive** and **High Bitrate**
- Single or multiple sources
- Not a network attack...
- Most TCP SYN floods
- Misbehaving TCP Receiver and Shrew

**Past Presentations**

- Attack traffic analysis
- Spectral Analysis (Fourier, PCA, Wavelets)
- Network Telescopes and Backscatter
- Statistical Classification
- Signature Detection
- Snort, Bro, Signal Processing, AI

**Mitigation**

Aggregate Congestion Control (ACC)

Local ACC Pushback

"Controlling High Bandwidth Aggregates in the Network"
*Computer Communications Review*
Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis
Vern Paxson and Scott Shenker

- Differentiate attack and benign traffic
- Network level “Congestion Signature”
- Rate limit traffic matching “congestion signature”
- Collateral Damage
- Benign traffic matching congestion signature
Detecting aggregates at the “local” router.

- triggered on “sustained congestion” (drop rate > $P_{\text{high}}$ for $K$ seconds)

Local ACC Setup

Detecting Aggregates

- Destination Address Prefixes
  - Empirically found to be a better fit
  - Find prefixes with...
    - High Aggregates
    - Least Collateral Damage

Rate Limiting

- Many Challenges
  - How much? till ambient drop rate at output queue is $P_{\text{target}}$.
  - False positives and negatives
  - Rate limit “congesting” flows by same amount (Limit > max “non-congesting” flow)
  - Flash Crowds: bound no. of rate limited flows

Simulations

- 5 aggregates
  - aggregates 1-4 are CBR flows
  - we change aggregate 5’s rate

Rate Limiting

- Measure arrival rates of rate limited flows
- Revisit rate-limiting periodically
- Remove potentially benign flows from limiter
- Avoid sudden changes in rate limits (?)
- Rate limited packets also go through RED

Contd...
Simulations

Without Local ACC

Local ACC

Simulations

With Local ACC

Local ACC

A Packet Dropped is a Packet Lost

So why not drop it right in the beginning?

Pushback

Rate limiting is propagated upstream

Triggered when an aggregate’s arrival rate is high for several seconds

Local ACC

Pushback

“Congestion Signature”
**Pushback**

- Congestion Contributing Links
  - Aggregate’s arrival rate division
  - Max - Min division
  - Pushback requests to upstream routers
  - Rate limit traffic going to congested router

- Upstream routers send feedback messages
  - Includes “total arrival rate for aggregate”
- Pushback refresh messages
  - Downstream routers calculate new rates
  - Auto terminate pushback rate limiting
  - Attacking pushback by rotating on-off flow

**Simulations**

- Vary no. of bad flows
- Experiments
  - No ACC
  - Local ACC
  - Local ACC with Pushback

- Local ACC
  - Local ACC + Pushback
Simulations

Comparing sparse and diffuse DDoS attacks

Sparse vs Diffuse DDoS

Good + Poor

Discussion

Fear of Collateral Damage

Studies using actual traffic traces

Possibility of using Local ACC itself for a DoS attack

Security of Pushback Protocol itself

Any more comments...?