Inter-domain DDoS mitigations: potentials, challenges, and solutions

Min Suk Kang*

Assistant Professor
Computer Science
National University of Singapore

16 November 2017

(*) joint-work with Prateek Saxena, Deli Gong, Shweta Shinde
Large-scale **volumetric** attacks are common and often congest **more than one** networks

- Escalation in **volume** of DDoS attack traffic; e.g., 300 Gbps in 2013 – 1.2 Tbps in 2016.
- Volumetric attacks often flood **upstream** autonomous systems (ASes) [WISR’16]
- Advanced **link-flooding** attacks **congest multiple** ASes concurrently

**Inter-domain** DDoS mitigation becomes **necessary** for large-scale volumetric attacks

- **Inter-domain packet filtering** is often necessary:
  - e.g., inter-AS links are flooded, large portion of AS is flooded

- **Packet-filtering outsourcing**: an AS asks another AS for packet filtering

- **State-of-the-art**:
  - **AT&T and CenturyLink***: *automated* packet-filtering outsourcing between two ASes for DDoS mitigation
  - **IETF DOTS**: standardization effort for common channels for inter-domain coordination

Holy grail of inter-domain mitigation: *source-end filtering*

```

Please drop X,Y,Z packets
```

- **Ideal** DDoS mitigation: *stops* attacks *earlier* by outsourcing filtering to source AS (e.g., D-WARD, StopIt)

- **Advantages**
  - *Reduction of bandwidth* waste (thus *cost* saving)
  - *Resource-demanding filtering* operations
  - *Local contexts* may be utilized (e.g., list of usual suspects)
Yet, source-end filtering has *not* been deployed due to the *lack of trust* between ASes

- **Mutually untrusted** ASes may launch attacks
  - Source AS can *modify* or *leak* the requested policy
- **No strong incentives** for source AS
  - Filtering incurs *non-negligible cost*.
- **Risk** of dropping packets
  - Source AS may be *blamed* for dropping its own customer packets
From *outsourcing* to *collaboration*: source and destination *collaboratively* determine filtering policy

- **Source** AS can also *express its own policy* for its customers (e.g., preference, black/white lists)

- Security concerns:
  1) *how to guarantee fair policy* composition?
  2) *how to protect the sensitive filtering policies*?
  3) *what if source AS bypasses* the packet filtering?
Desired property 1)

**Fair composition** must be **verified by both** source and destination ASes

- **Collaboration** platform requires **fair** policy composition and enforcement
- **Fair**: we find a **middle ground** of two policies, favoring neither of the policies
  - Example: ASes express flow preferences

  - Two ASes should be able to **verify** the fair composition and enforcement of their policies
Desired property 2)

**Filtering policy** of each AS must be *protected* from each other with privacy guarantees

- Filtering policies are inherently *sensitive*; e.g.,
  - Preference due to *private contracts, proprietary algorithms*
  - *Attack* information, *vulnerable points* of the network
  - Internal *white/black policies*
- Two ASes should be able to *negotiate* and determine the *guaranteed degree of privacy*

Desired property 3)

**Filtering operations** must *not* be *bypassed* for *any packet* from source to destination

- Source AS *can evade* the filtering when it wishes to ignore the composed filtering policy
- *Non-bypassability:* filtering operation *must be invoked* for *all* packets from source AS to destination AS
Middlebox-based filtering: a practical design choice for rapid and widespread deployment

- **Commodity** hardware for multi-Giga-bps throughput
- **Trusted execution environment (TEE)** capabilities (e.g., memory isolation, remote attestation) from commodity CPUs
  \[\Rightarrow\] verifiable control-/data-plane operations
**TEE-based middlebox** can satisfy three desired **security properties**

- **Policy composition and enforcement** are **isolated** and **verifiable** via remote attestation
  - *fair packet filtering is guaranteed*

- Two ASes can **negotiate** the desired level of **privacy** and policy **fairness**
  - **policy inference attack**: \( \sim O(\log(N)) \)
  - **tradeoff**: degree of anonymity vs. fairness

- **Bypass** is **immediately detected** by **efficient sketch** and **MAC** operations
  - **only 5-tuple** information copied to **TEE**
    - \( \Rightarrow \) **multi-Giga-bps** performance
Preliminary results: multi-Giga-bps filtering with Intel SGX platform

• Filtering up to ~1.8 Gbps with 3 Intel SGX cores
  • CPU: Intel® Core™ i5-6400
  • Memory: 8 GB (128 MB reserved for EPC)
  • NIC: Intel® 10-Gigabit X540-AT2

• **TCB** – 1,369 SLoC, 1.9 MB binary.

• SGX-integration of **DPDK**
  • libraries ported to SGX: mempool, mbuf, ring, sched

• Plan: *scale out* with load balanced parallel middleboxes for *10 Gbps* or higher throughput
Conclusion

**Collaborative source-end packet filtering**

**Potentials:**

*Collaborative source-end filtering* is an ideal defense for ever-increasing volumetric attacks

**Challenges:**

*Lack of trust* between ASes makes existing solutions impractical

**Solutions:**

*TEE-based middlebox* solution can offer *three security properties* (i.e., verifiable fairness, privacy, non-bypassability) necessary for secure and practical collaborative DDoS solution
We are open for *feedback* and *collaboration*

kangms@comp.nus.edu.sg
http://www.comp.nus.edu.sg/~kangms/