Computer Networks 105 (2016) 180–193

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Tumbler: Adaptable link access in the bots-infested Internet☆

Yao Zhang[a,b,*], Xiaoyou Wang[c], Adrian Perrig[b], Zhiming Zheng[a]

[a] LMIB and School of Mathematics and Systems Science, Beihang University, Beijing, China
[b] Institute of Information Security, Department of Computer Science, ETH Zurich, Switzerland
[c] Information Networking Institute, Carnegie Mellon University, Pittsburgh, USA

ARTICLE INFO

Article history:
Received 14 April 2015
Revised 2 June 2016
Accepted 5 June 2016
Available online 7 June 2016

Keywords:
DDoS attack
Capability scheme
Bandwidth allocation
Competition factor

ABSTRACT

Despite large-scale flooding attacks, capability-based defense schemes provide end hosts with guaranteed communication. However, facing the challenges of enabling scalable bandwidth fair sharing and adapting to attack strategies, none of the existing schemes adequately stand. In this paper we present Tumbler, a flooding attack defense mechanism that provides scalable competition-based bandwidth fairness at the Autonomous System (AS) granularity, and on-demand bandwidth allocation for end hosts in each AS. Tumbler enforces adaptability in the capability establishment via competition factors that are calculated upon leaf ASes' bandwidth utilization and reputation. Transit ASes independently manage each competition factor based on the corresponding feedback from dedicated bandwidth accounting and monitoring policies. Through Internet-scale simulations, we demonstrate the effectiveness of Tumbler against a variety of attack scenarios and illustrate the deployment benefits for ISPs.

1. Introduction

Individuals, industries, and governments are increasingly relying on Internet availability for dependable services. At the same time, Distributed Denial-of-Service (DDoS) attacks remain persistent threats in the current Internet. Emerging DDoS attacks are launched by millions of bots [1], flooding victim links with huge amounts of traffic [2–4]. Recent attacks have even been observed with startling 500 Gbps [5].

In response to these attacks, network capability-based mechanisms [6–14] have emerged as a promising class of DDoS defenses. In a capability-based scheme, a source initiates a capability request to a destination with all the routers on the transiting path adding authentication tokens to the packet header. Upon the request's arrival, the destination can explicitly authorize a desired flow with priority and return the packet to the source. Then generated tokens will act as a capability for the traffic of the authorized flow. By ensuring end-to-end privileged flows, such approaches isolate attack traffic. However, to achieve viable link-flooding defense, capability-based schemes face two critical challenges: *scalability* and *adaptability*.

Scalability refers to the fundamental problem of providing fair access, for the aspects of both capability bootstrapping and bandwidth allocation. Concerning capability bootstrapping, what if capability establishment is interfered by attackers? Given the fact that capability requests are forwarded by best effort, flooding on the requesting channel, namely Denial-of-Capability (DoC) attacks [15], could easily prevent benign sources from obtaining capabilities. Similarly, concerning bandwidth allocation, what if attackers leverage authorized capability packets to flood a link? For any per-source [10] or per-destination [8] fair-sharing mechanism, m end hosts only obtain $1/m$ of the capacity at a given link. The available bandwidth in per-flow schemes is limited to only $1/m^2$ [7]. In a nutshell, with millions of bots competing for the limited capacity of a link, the obtained link access for a legitimate end host becomes infinitesimal, too small to provide useful end-to-end guarantees.

Adaptability refers to another issue on how the ISPs can not only economically maximize their link utilization, but also dynamically provide each customer with deserved link access even under persistent DDoS attacks. The PSP scheme [16] leverages historical traffic to adjust bandwidth allocation for core network flows. But this model would hardly scale to end-to-end guarantees in the Internet as billions of flows have to be considered in the iterative computation. Moreover, previous defenses achieve mostly one-time resilience rather than adjustable protection. A sophisticated attack (e.g., replay flooding) would likely cause perpetual damage to the victim link. Intuitively, an approach to improve adaptability is to combine capability establishment with the observation of traffic.

☆ This project was conducted at ETH Zurich while the first author was a visiting student with the network security group.

* Corresponding author.
E-mail addresses: yaozhang@buaa.edu.cn (Y. Zhang), xiaoyouw@andrew.cmu.edu (X. Wang), adrian.perrig@inf.ethz.ch (A. Perrig), zzheng@pku.edu.cn (Z. Zheng).

http://dx.doi.org/10.1016/j.comnet.2016.06.005
1389-1286/

However, as ISP distance increases, both mutual trust and business incentives diminish. In such cases, cooperation between ISPs becomes an impractical requirement.

To overcome the above limitations, we propose Tumbler, a novel mechanism that provides scalable link access for end hosts and adaptable DDoS-resilience for ISPs in the Internet. Tumbler is designed with the following insights and solutions.

First, resource fair sharing at an Autonomous System (AS) granularity provides a scalable approach of solving the link access problem. Although adversarial botnets can be widely distributed in the Internet, the massive number of bots (on the order of millions) resides in ASes whose scale is always significantly smaller. By September 2015, the total AS number is about 50,000 [17]. Therefore, defending against DDoS attacks by throttling bandwidth among ASes will efficiently limit the damage over the contaminated ASes (i.e., ASes initiate the flooding attacks), regardless of their internal botnet size. Tumbler enforces a competition-based weighted fair sharing among leaf ASes. Namely, each transit AS records a periodically-updated *competition factor* for each leaf AS, based on which an active AS (i.e., ASes have valid allocation on the link) obtains its weighted-shared link access. Link access enables both capability requests and capability-enabled data packets. Subsequently, aggregated access will be further shared by the individual flows from the same AS.

Second, a simple per-AS fair sharing [11,12] is not an optimal strategy. Indeed, the *bandwidth utilization* from different ASes varies significantly due to its connection popularity. Even during different hours of a day, bandwidth utilization from an AS fluctuates [18]. In Tumbler, bandwidth consumption of a leaf AS at the given link is considered as the main factor in Tumbler's competition-based weighted fair sharing, which enables effective management of the link bandwidth. Historical bandwidth allocation statistics are kept at each router via local off-line accounting, and serve as a feedback via competition factor to influence the next round of bandwidth allocation.

Furthermore, despite the lack of inter-AS cooperation, local network analysis at one AS still enables adaptable capability establishment. In Tumbler, an *AS's reputation* is evaluated through regional traffic monitoring, and considered as the other factor in the competition-based weighted fair sharing. Such evaluation is performed independently at every traversing AS, thus eliminating the requirement of mutual trust between ISPs. Specifically, each AS monitors its inbound/outbound traffic. Based on defined traffic policies of bandwidth violation, timely feedback will be returned to adjust each AS's reputation and further regulate the subsequent capability establishments for each AS.

The rest of the paper is organized as follows. We specify the goals and assumptions in Section 2, and present the design of Tumbler in Sections 3 and 4. In Section 5, we analyze the security and overhead aspects of Tumbler. In Section 6, we compare Tumbler with related approaches through Internet-scale simulations, and confirm experimentally the properties of our scheme. More discussions are given in Section 7. Related work is given in Section 8, and we conclude in Section 9.

2. Design goals and assumptions

We first give the basic design goals. Then we specify the assumptions and the threat model that Tumbler aims to combat. For clarity, we denote the end host who sends as "source", and the end host who receives as "destination". We refer to the ASes who contain end hosts as leaf ASes, while other ASes on the routing paths as transit ASes. We call the leaf AS where the source (destination) resides the source (destination) AS.

2.1. Design goals

The following design goals enable a lightweight and deployable capability-based DDoS defense framework in today's Internet.

Scalability. We desire a defense mechanism that achieves scalable link access under the presence of botnets. In addition, an Internet-scale mechanism must be lightweight and incur minimal overhead on the deployed routers.

Adaptability. The mechanism should be able to optimize the utilization of link capacity in terms of economical benefits, and to dynamically adjust its defense strategy based on the attacks evolving over time.

Deployability. The mechanism is expected to be functional even if a few entities adopt. Moreover, the deployment plan should incentivize the early adopters.

2.2. Assumptions

First of all, we assume that when a source sends a capability request, it has the knowledge of network routing, namely a valid inter-domain path to the destination AS. Note that the requirement of path control is not a necessity here. Although selecting a specific routing mechanism remains outside the scope of this paper, multiple routing mechanisms can be leveraged to obtain AS-level paths: A straightforward approach is Border Gateway Protocol (BGP) [19] routing (further discussed in Section 7.1). Additionally, several solutions like NIRA [20], Pathlets [21], and SCION [22] present a fix. In these schemes, source could obtain and specify a valid routing path in the header of packets.

Additionally, we assume that all flows from an AS can be assigned with a unique and unforgeable source-AS identifier. This goal is achievable via some lightweight source authentication protocols [23,24]. In Tumbler, for simplicity, we set a source-AS identifier as the hash of the domain's public key.

2.3. Threat model

We assume that both end hosts and ASes may be malicious: It is possible for any end host to get compromised and support DDoS attacks. The ASes containing such botnets or even other transit ASes may tolerate the malicious traffic thus to assist the cooperative attacks. Yet, cases that ASes intentionally forge/distort packet information, or delay/drop packets are outside our paper scope. We require no restriction on the distribution of botnets, and within a contaminated AS, the botnet could have an arbitrary number of attackers. When evaluating a given end-to-end communication, both source and destination ASes are assumed to be non-contaminated.

In addition to these settings, we consider two classes of DDoS attacks: (1) Request packet flooding (DoC attacks [15]) and (2) capability packet flooding, where botnets may collude and present a coordinated behavior with different strength scenarios (pulsing attack [10]), or location scenarios (rolling attack [11]).

3. Tumbler overview

We present the high-level overview of Tumbler in this section. The Tumbler protocol includes the following two phases:

- Phase 1 (capability establishment): Source and destination set up a communication channel. A communication capability is generated hop-by-hop on the routing path according to the request configuration of the source and the link access admission policies of the transit routers.
- Phase 2 (data transmission and feedback regulation): Source sends data on the channel by adding its latest capability into the packets. Meanwhile, transit ASes perform accounting and