Bandwidth Distributed Denial of Service: Attacks and Defenses

Moti Geva, Amir Herzberg, Yehoshua Gev

Computer Science Department, Bar-Ilan University
Ramat Gan 5290002, Israel
{moti.geva,amir.herzberg,yoshigev}@gmail.com

Abstract

Distributed denial of service (DDoS) attacks pose a serious threat to the Internet. We discuss the Internet’s vulnerability to Bandwidth Distributed Denial of Service (BW-DDoS) attacks, where many hosts send a huge number of packets exceeding network capacity and causing congestion and losses, thereby disrupting legitimate traffic. TCP and other protocols employ congestion control mechanisms that respond to losses and delays by reducing network usage, hence, their performance may be degraded sharply due to such attacks. Attackers may disrupt connectivity to servers, networks, autonomous systems, or whole countries or regions; such attacks were already launched in several conflicts.

In this paper we survey BW-DDoS attacks and defenses. We argue that so far, BW-DDoS employed relatively crude, inefficient, ‘brute force’ mechanisms; future attacks may be significantly more effective, and hence much more harmful. We discuss currently deployed and proposed defenses. We argue that to meet the increasing threats, more advanced defenses should be deployed. This may involve some proposed mechanisms (not yet deployed), as well as new approaches. This article is an overview and will be of particular interest to readers who want to learn about bandwidth DDoS attacks and defenses.

Keywords: Denial of Service, DoS, DDoS, DoS Attacks, DoS Mitigation, Bandwidth flooding

I. INTRODUCTION

Internet services are indispensable – and yet, vulnerable to Denial of Service (DoS) attacks, and especially to Distributed DoS (DDoS) attacks. In this paper we focus on DDoS attacks, in which many attacking agents cooperate to cause excessive load to a victim host, service, or network. DDoS attacks have increased in importance, number and strength over the years, becoming a major problem. In recent survey of network operators [1] (2012, Fig. 10), DDoS was the most common identified ‘significant threat’ (76% of respondents). Furthermore, significant growth in size of attacks and in their sophistication is reported (e.g., [1], [2]).

We further focus on Bandwidth Distributed Denial of Service (BW-DDoS) attacks, which disrupt the operation of the network infrastructure by causing congestion, i.e., excessive amount of traffic. Congestion may be due to the total amount of traffic (in bytes), or the total amount of packets (often a lower limit, using short packets). BW-DDoS attacks can cause loss or severe degradation of connectivity, between the Internet and victim network(s) or even whole autonomous system(s), possibly disconnecting whole regions of the Internet. Reported BW-DDoS reached volume of 100 Gbps [1] and recently even reached 300 Gbps [3]; according to [2] 60-86.5% of the BW-DDoS targeted the infrastructure (layer 3), including the mitigation infrastructure itself.

In particular, in this paper we do not discuss DoS/DDoS attacks which are not focused on bandwidth, including exploits of vulnerabilities and limitations of specific application/transport protocol implementations (e.g., SYN flooding), service (e.g., ping of death) or device (e.g., attack exploiting router and firewall vulnerabilities). For a general DDoS attacks and defenses taxonomy (not limited to BW-DDoS), see e.g. [4].

This research was supported by the Israeli Science Ministry and the Israeli Science Foundation (ISF).
BW-DDoS attackers may use different techniques as well as different attacking agent capabilities. A strong attacking agent is a privileged-zombie, i.e., a software agent having high privileges and complete control over the machine on top of which it is being executed, with the ability to make manipulations to the protocol stack, e.g., being able to send spoofed IP packets. Weak agents include puppets, that is, programs downloaded automatically and run within sandboxes, such as JavaScript based web-pages. Next, an attacker may use simple types of bandwidth flooding, or elaborate techniques to amplify its bandwidth, such that uncompromised machines assist it with consuming bandwidth.

In Section II, we discuss significant known BW-DDoS attacks, and compare them in respect to their effectiveness and requirement, considering the attacking agents, protocol manipulations, amplification and attack target. We discuss widely used attacks as per [1], [2], as well as more advanced attacks, which so far were only presented academically. We argue that as more advanced attacks are adopted and cyberwarfare threats increase, BW-DDoS growth rate may further accelerate.

In Section III, we discuss important results from the vast body of research regarding BW-DDoS defenses, developed in the past 20 years. Some of these defense techniques are widely deployed defenses in practice, and other defenses were only proposed academically. We discuss defenses from both groups, and compare them in respect to their response to the attack, the location in which they are deployed, their require infrastructure adaptations, and their dependency on cooperation. Considering reported attacks, we argue that existing defenses may not suffice in the future. Hence, further research, standardization and development of new practical defense mechanisms is required to ensure that defenses can withstand the potential increasingly-stronger attacks.

II. BANDWIDTH DENIAL OF SERVICE ATTACKS

BW-DDoS attacks are usually generated from a large number of compromised computers (zombies or puppets). According to recent surveys, e.g., [1], [2], Bandwidth Distributed Denial of Service are the most frequently used DoS method. Most BW-DDoS attacks use few simple ideas, mainly, flooding, i.e., many agents sending packets at the maximal rate, and reflection, i.e., sending requests to a server with fake (spoofed) sender IP address, resulting in server sending (usually longer) packet to the victim. Table II summarizes the different attacks discussed in this section.

Flooding attacks have created significant damages, since the attackers used a sufficient number of agents, to cause massive bandwidth consumption and packet losses. It seems that gradually, attackers are adopting more complex and effective attacks. E.g., the largest attacks reported in recent years in [1], [3], consisting of 100 Gbps (2010), 60 Gbps (2011, 2012) and 300 Gbps (2013) [3]. Attacks on 2010,2011 and 2013 were DNS reflection/amplification attacks, described below. In 2012 the attack was aimed at the DNS infrastructure itself. This trend, of using more effective attacks, is alarming, since even significantly more effective BW-DDoS techniques, e.g. with higher amplification factors, were discovered by researchers (and more may exist); see Section II-B.

Nevertheless, inducing a significant percent of packet loss is no easy task for an attacker. Generally, packet delivery probability is the ratio between the available bottleneck link bandwidth and the attack’s rate. However, as depicted in Figure 1, congestion or (small) packet loss probability causes TCP connections a dramatic performance degradation. This performance degradation is due to TCP’s congestion control mechanism which drastically reduces TCP’s sending rate on packet loss. It follows that BW-DDoS damage may be worse than the mere consumed bandwidth.

Figure 2, depicts the results of an Internet scale simulation we have conducted, which can be used to emphasize the potential damage of various sized BW-DDoS attacks. The simulated topology was constructed based on the AS topology from CAIDA [5], taking into consideration client-provider and sibling constraints. To derive links’ bandwidths, we took an approach similar to the one proposed by the PAWS simulator [6]: each AS was categorized as either large, medium and small, based on the number of links it has with other ASs. The AS sizes were then used to derive the bandwidth capacity of each link as described in Table I. We have simulated BW-DDoS attacks, consisting of between 200 and 102, 400
Fig. 1. Experimental results of delivered rate vs. congestion over a bottleneck link. The diamonds (♦) line is TCP, the pluses (+) and squares (□) lines are constant rate UDP, and the dashed lines are theoretical rates for constant rate UDP flows. TCP reduces its rate as a function of available bandwidth, whereas UDP suffers from packet loss. In the topology: the top solid line (blue) represents the legitimate traffic, and the bottom dashed line (red) represents the attacker. All links are 100 Mbps.

Fig. 2. Percent of available Autonomous Systems (AS) vs. attack size (Mbps). White bars are ASs with available bandwidth to support TCP connections from all incoming routes. The gray bars are ASs with available bandwidth for only part of their incoming links. The black bars are fully congested ASs which cannot sustain TCP. The figure implies that most of the Internet is prone to DDoS at attack scales seen to date.
randomly distributed zombies; each zombie sends 1 Mbps of UDP traffic to the victim AS, resulting in attack rates between 200Mbps and 100Gbps. Large scale attacks, within the order of magnitude seen to date, can cripple even large ASs let alone specific hosts or networks.

BW-DDoS attacks come in many flavors, and utilize various mechanisms to induce excessive bandwidth consumption. We shortly discuss the main features which can be used to differentiate between different attackers, and discuss which capabilities are required by the attacker to launch such attacks.

A. Attacking Agent

We consider three types of zombies: puppets, zombies, and root-zombies; the difference between the various agents is explained below. Acquiring puppets is relatively easy, and can be done by fooling users to browse to the attacker’s website. Zombies are harder to take over, as they require the attacker to install a malware on the zombie machine by exploiting some vulnerability or tricking users into installing the malware themselves. Root-zombies require that either an installed zombie was initially installed with high privileges, or have a privilege escalation exploit that can gain such high privileges.

Let us first consider a naive BW-DDoS attack, where the attacker sends as many packets as possible directly to the victim, or from an attacker controlled machines called ‘zombies’ or ‘bots’. The simplest scenario is one in which the attacker is sending multiple packets using a connectionless protocol such as UDP. In UDP flood attacks, the attacker commonly has a user-mode executable on the zombie machine which opens a standard UDP sockets and sends many UDP packets towards the victim.

For UDP floods, and many other BW-DDoS attacks, the attacking agents must have zombies, i.e., hosts running adversary-controlled malware, allowing the malware to use the standard TCP/IP sockets. Other attacks require only puppets [7], i.e., scripts, applets, etc., downloaded and run automatically by client agents such as web browsers. Being untrusted, puppets operations are restricted by a sandbox; e.g., they cannot send UDP packets let alone spoof packets, and they are limited in establishing TCP connections. Nevertheless, even though puppets cannot induce as much bandwidth as zombies, they can still induce significant bandwidth usage. For example the maxSYN attack described in [7], aims to maximize SYN packets by setting the sources of several JavaScript image objects to be nonexistent URLs repeatedly every 50 milliseconds. Every time the script writes into the image source URL variable, old connections are stalled by the browser and new connections are established to fetch the newly-set image URL.

On the other hand, other types of attacks require zombies to have administrative privileges for execution. We refer to privileged zombies as root-zombies. For example, to send packets with spoofed source IP address, a zombie commonly requires the ability to open raw sockets, which is permitted for privileged users only. It is also important to note that some protocol manipulations require network support, or more accurately: lack of prevention. Specifically, spoofing is commonly filtered by ingress filtering as discussed in Section III-A.

B. Attack Mechanism

We consider three types of attack mechanisms: direct-flooding, amplification and reflection. In the naive attack the attack traffic is limited by the bandwidth capacity of the compromised machines, and the entire

<table>
<thead>
<tr>
<th>Src/Dst</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>OC-3</td>
<td>OC-12</td>
<td>OC-24</td>
</tr>
<tr>
<td></td>
<td>155.52 Mbps</td>
<td>622.08 Mbps</td>
<td>1.244 Gbps</td>
</tr>
<tr>
<td>Medium</td>
<td>OC-12</td>
<td>OC-48</td>
<td>OC-192</td>
</tr>
<tr>
<td></td>
<td>622.08 Mbps</td>
<td>2.488 Gbps</td>
<td>9.953 Gbps</td>
</tr>
<tr>
<td>Large</td>
<td>OC-24</td>
<td>OC-192</td>
<td>OC-768</td>
</tr>
<tr>
<td></td>
<td>1.244 Gbps</td>
<td>9.953 Gbps</td>
<td>39.813 Gbps</td>
</tr>
</tbody>
</table>

TABLE I

Bandwidth estimation based on AS size. AS size is estimated based on the number of links it has to other ASes. Small AS has up to 4 links. Medium AS has up to 300 links. Large AS has more than 300 links.
load on the victim is due to the direct-flooding induced by packets sent by the zombies. Amplification
attacks use the attacking agents’ bandwidth more effectively, such that on average every packet sent by a
zombie causes transmission of multiple and/or larger packets to the victim by non-compromised machines.
Specifically, the attacker chooses a request $r$ of size $|r|$ which results in longer response $r'$ of size $|r'|$,
achieving amplification factor of $\frac{|r'|}{|r|}$. Hence, direct-flooding BW-DDoS attacks have no amplification or
amplification factor = 1, and sophisticated attacks (e.g. DNS amplification attacks discussed shortly) can
have factors greater than 1.

**DNS amplification** attacks [8] rely on the fact that DNS responses may be larger in size than DNS
requests. DNS requests are pretty short, e.g., 40 bytes, whilst responses may be much longer. Originally,
DNS responses over UDP were limited to 512 bytes; however, DNS extensions (EDNS) allows longer
responses, e.g., 1500 and even 4000 bytes. Hence, DNS amplification factor can be $\frac{512}{40} = 12.8$, and even
up to $\frac{4000}{40} = 100$ with EDNS. With the uptake of DNSSEC, which relies on the long packet capabilities
of EDNS, long responses are likely to become more common.

Theoretically, based on an amplification factor = 100 an attacker requires roughly 100 zombies, each
sending DNS requests at 100Kbps, to achieve a 1Gbps attack. For a 10Gbps attack 1,000 zombies are
required, etc. Alarming, significantly larger botnets, consisting of hundreds of thousands of zombies,
have already been discovered.

Reflection attacks cause legitimate hosts to be fooled into sending unsolicited responses to victim hosts.
For example, DNS reflection attacks are based on the DNS protocol, which is a UDP-based request-
response protocol. A DNS resolver (server) will return responses to clients which issue DNS requests.
The clients’ return address is determined using the source IP address appearing in the request. DNS
reflection attacks exploit this behavior: the attacker sends a spoofed DNS request to a DNS server making
the server issue a response packet to the spoofed address. Commonly, the attacker uses the victim’s IP
address as the spoofed source address. Further, the attacker commonly amplifies its attack, causing the
reflected response to be longer.

### C. Protocol Manipulations

We discuss two types of protocol manipulations which can be used by an attacker. The first attempts
to avoid detection, and the second tries to exploit legitimate protocol behavior and cause legitimate
clients/server to excessively misuse their bandwidth against the attacked victim. Typically, protocol
manipulation for bandwidth attacks require a strong zombie with administrative privileges, as the
manipulation is commonly done at the low protocol layers handled by the operating system, usually
IP and TCP. Note that “no manipulation” is also an option for some BW-DDoS attacks.

For naive attacks such as UDP floods, the sources of the attack are visible, that is, the victim host can
see the zombies’ source addresses, making it relatively easy to block packets and to take technical or
legal measures against the attacking machines and their owners. Therefore, an attacker may try to avoid
detection by manipulating its source IP address, commonly referred to as spoofing. Thus an attacker can
send multiple packets each contains a different spoofed source address, making it harder for the victim
to identify the attacker source, and block them.

The second type of attacks exploits legitimate protocol behavior by misusing packet fields, or acting
dishonestly, causing a legitimate hosts to send unwanted or excessive packets to the victim. For example,
using IP spoofing an attacker can induce a DNS reflection attack as discussed in Section II-B.

Some attacks, such as optimistic-acknowledge (opt-ack) [9], take advantage of other mechanisms, such
as TCP congestion control. Generally, the congestion control mechanism adapts TCP transmission rate
based on available bandwidth. Typically, the assumption behind the congestion control mechanism is that
the (main) reason for packet loss is congestion. Hence, whenever TCP packets are successfully received
the rate increases, and whenever packets are lost the rate is reduced. The way that TCP knows whether
a packets has been received or not, is based on ACK(nnowledge) packets sent by the destination. The
opt-ack attack idea is therefore simple: a malicious client sends some request to a server, then, as the
server sends the response packets, the client optimistically acknowledges receiving them by sending ACK packets, without actually receiving the packets themselves. Thus, very low bandwidth is required to cause servers to send lots of traffic, limited mainly by the servers’ bandwidth.

ACK-Storm [10] attacks TCP’s acknowledge mechanism. The attacker eavesdrops on an existing TCP connection. Next, the attacker spoofs ACK packets with a higher sequence number than what was actually sent; this induces an ACK packet back containing the real sequence number. Sending such packets to both connection ends respectively, induces a repeated back and forth exchange of ACK packets, until either end eventually terminates the connection.

D. Attack Target

We discuss two types of attack targets: last-mile links and backbone-links. While most BW-DDoS target the last-mile links, new type of attacks are also trying to attack backbone-links. For example, Coremelt [11] uses a peer-to-peer model in which zombies communicate directly with each other. Amongst N zombies there exist $O(N^2)$ routes, some of which use the victim backbone-link. The attacker can then create excessive traffic on the link using only inter-zombie communications. Coremelt requires regular zombies without high privileges, as it can use the standard TCP/IP stack without any protocol manipulation. Coremelt type of attacks is mainly theoretical since the Internet backbone-links are highly provisioned and would require huge peer-to-peer networks to clog. Moreover, based on CAIDA datasets [5], we can say that there are more than a 100,000 links between more than 35,000 different autonomous systems, making it very hard to take down almost any specific backbone-link. Nevertheless, assuming enough zombies can be obtained, Coremelt may prove difficult to detect and filter, since each connection can use a small amount of bandwidth.

Concluding this section we argue that current attacks use relatively crude methods while future attackers are likely to be able to use significantly more effective attacks, with higher amplification factors. It is very hard to determine how difficult it is for an attacker to actually launch such advanced attacks in the Internet, but the basic is knowhow is studied and to some extent even demonstrated. Nevertheless, existing attacks and especially advanced attacks may challenge currently deployed defense mechanisms, motivating investigation of new mechanisms.

III. Network-Level Defense Mechanisms

BW-DDoS defense mechanisms focus on several types of schemes, including detecting, filtering, absorbing, and cooperating. In this section we survey defense schemes of both deployed and academically proposed mechanisms, and provide examples for the various schemes. We discuss different mechanisms, their deployment location in the network, the infrastructure adaptation and type of cooperation they require, if any. Note that many defense mechanisms rely on the ability to differentiate between attack and legitimate flows, however, in this paper we omit the discussion regarding the extensive research on differentiation techniques as they were surveyed before, e.g. in [12]. Table III summarizes the defense mechanisms discussed in this section.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Agent</th>
<th>Mechanism</th>
<th>Protocol Manipulations</th>
<th>Target link</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP Flood</td>
<td>zombie</td>
<td>direct-flooding</td>
<td>No manipulation</td>
<td>last-mile</td>
</tr>
<tr>
<td>maxSyn [7]</td>
<td>puppet</td>
<td>direct-flooding</td>
<td>No manipulation</td>
<td>last-mile</td>
</tr>
<tr>
<td>DNS Reflection [8]</td>
<td>root-zombie</td>
<td>reflection (&lt; 100) IP (spoof)</td>
<td>last-mile</td>
<td></td>
</tr>
<tr>
<td>opt-ack [9]</td>
<td>root-zombie</td>
<td>amplification (&gt; 1,600 [9]) TCP (congestion control)</td>
<td>last-mile</td>
<td></td>
</tr>
<tr>
<td>ACK Storm [10]</td>
<td>root-zombie</td>
<td>amplification (&gt; 40,000 [10]) IP (spoof), TCP (ACK mechanism)</td>
<td>last-mile</td>
<td></td>
</tr>
</tbody>
</table>

TABLE II
COMPARISON BETWEEN BW-DDoS ATTACKS
A. Response Mechanism

We consider four types of defense mechanisms: filtering, rate-limiting, detouring and breaking-through. Assuming the offending flows are identified, they can be filtered out, that is, blocked. Filtering can take place in various network locations: close to destination, at the core (i.e., routers), or close to source. Usually, to be effective in BW-DDoS mitigation, filtering should occur before the congested link, as the victim itself is usually not in position to hold the attack back.

One filtering example, is preventing source IP spoofing. RFCs 2827 and 3704 (BCPs 38, 84), recommend that ISPs employ *ingress filtering* and filter packets with IP addresses that are external to that network. This is performed by many ISPs, however, [1], [13] indicate that approximately 15% of Internet addresses can still send spoofed packets. LOT [14] is another solution to mitigate spoofing by opportunistically establishing tunnels between gateways and adding a random tag to tunneled packets, making it difficult for an attacker to guess the correct tag value. Packets not carrying a correct tag are discarded, hence preventing the spoofing of packets which originate from incorrect networks.

Additional deployed filtering mechanisms include Access Control Lists (ACL), Remote-Triggered Black-hole (RTBH), and firewalls. ACLs are router mechanisms which allow or deny matching flows. ACLs are often configured manually, however some Intrusion Prevention Systems (IPS) can configure ACLs automatically. Each ACL entry takes a significant amount of memory and some time to process, therefore a router should limit the ACL rules it holds, both in number as well as the processing time they take. Both memory and CPU usage increase as more ACL entries are used, which may be an additional cause for DDoS – not necessarily bandwidth based.

RTBH (RFC5635) uses the router’s forwarding tables, such that *all traffic* to the victim, or from attacking sources, is forwarded to a “black-hole”, completely denying access to the target. The main reasons for using RTBH is that it uses small amount of memory and its processing is faster than ACL processing. However, RTBH filtering is significantly more aggressive, and may help an attacker to disconnect its victim from its sources/destination, thereby potentially achieving the attacker’s goal, with little resources.

In contrast to completely blocking the attacking flows, *rate-limiting* schemes let the offending flows transmit their *typical rate*, or obey some other limit.

*Rate-limiting* at routers was proposed in the literature in several main forms, *capabilities, packet tagging*, and *scheduling based*. *Capabilities* are tokens, issued by the destination (server) to the source (client). The capability informs that the destination is willing to accept traffic from the source. The issued capability is attached to packets sent by the source, allowing routers en route to identify and prioritize approved flows. Note that packets without capabilities are not filtered; instead they get lower delivery probability which effectively limits their rate during attack periods.

SIFF [15] proposed stateless capabilities, in which the capability is calculated using (keyed) hash. Routers check and prioritize flows carrying verified capabilities. TVA [16] keeps a (small) state in routers, and allows servers to request specific restrictions per flow. Capabilities based solutions assume that the victim will only authorize legitimate sources, and not cooperate with the attacker. Deployment of capabilities based solutions will require change to both end-hosts and routers.

PSP [17] collects network statistics at the provider level, and infer the typical traffic rates between origin–destination pairs. Packets are tagged upon arrival to the provider as either normal or excessive. Whenever a router gets congested, packets tagged excessive are discarded first, effectively prioritizing packets tagged as normal. PSP deployment is only within the provider boundaries, and requires changing routers’ software for the packet tagging and prioritizing, or otherwise take advantage of existing IP packet fields, which may be used by different applications and hence potentially cause damage to some flows.

The third type of scheduling based filtering in routers was suggested by BTT [18]. Like PSP, BTT collects network traffic statistics, however it does not make any packet manipulation. Instead, BTT employs a *weighted fair queuing* (WFQ) scheduling scheme which prioritizes flows transmitting at their typical rate, over flows using excessive traffic. Next, BTT requests upstream BTT nodes to shape their traffic going through the congested link, for example by using token buckets. The main advantage of BTT is that it merely configures the dataplane routers using existing router mechanisms, without requiring any
router software/firmware change. However, despite the fact that BTT can be gradually deployed, it requires cooperation between autonomous systems (AS) (see Section III-D), and extensive deployment to become really effective. Both PSP and BTT assume that the typical rate exists and that it is measurable.

Additional schemes use overlay networks and cloud computing. Overlays mitigating BW-DDoS can be generally divided into two types, detouring overlays and absorption overlays. Detouring overlays bypass the network end-to-end routing, overcoming BGP’s shortcomings, such as update speed, route selection under different matrices, or utilizing special network features such as multihoming; see [19]. Detouring overlays can implicitly mitigate BW-DDoS, only when some routes are congested while other are not, as depicted by the gray bars of Figure 2.

Absorption overlays are over-provisioned with bandwidth, and are able to absorb the BW-DDoS. Absorption overlays construct a perimeter around the victim server, which can be penetrated only by selected nodes; unauthorized traffic is filtered. “In the cloud” (practical) or “overlay” (academic) solutions, route traffic via the cloud/overlay, which “scrubs” the attack flows. Absorption overlays/clouds are specifically designed to mitigate BW-DDoS, and were investigated in several works, such as SOS [20]. Note that usually, overlay solutions introduce new protocols, and hence typically require updating host software. Other solutions, mainly the deployed, make no protocol changes, and instead rely on configuring BGP or DNS records as to divert the traffic to a cloud-based scrubbing service.

The final category for BW-DDoS mechanism is breaking-through the congestion by using aggressive clients. Aggressive clients use TCP-friendly protocols as long as they can sustain enough goodput. Whenever TCP’s goodput drops below some threshold, aggressive clients commence using protocols without congestion control, such as UDP, thereby exploiting the real network delivery probability, as depicted in Figure 1. An important design goal of aggressive clients is to avoid self generated BW-DDoS.

QoSoDoS [21] uses an assumption that even under a strong BW-DDoS, there remains a non-negligible packet delivery probability. Hence, whenever TCP’s goodput drops, QoSoDoS retransmits packets using UDP. Assuming the attacker’s rate can be bound, it is possible to assure modest QoS in high probability, while limiting the number of retransmissions. By controlling the number of QoSoDoS clients and their transmission rate, QoSoDoS can avoid self-created BW-DDoS. Deployment of QoSoDoS would require changes to end-hosts.

B. Defense Mechanisms Location

There are various defense mechanisms, which can be deployed at different network locations. A defense mechanism can be deployed close to the destination, that is, by the victim. Note that defense mechanisms close to the destination may get a good idea about some attack’s properties, but for mitigation of BW-DDoS they might not be the well positioned, since many packets already get discarded near the victim. Hence, many defense mechanisms try to mitigate the attack closer to its sources.

Router or backbone based defense mechanisms are usually located near an over-provisioned link, and try to make sure that the traffic reaching the victim will be mostly from legitimate sources rather than attack flows. Similarly, source based defense mechanisms try to prevent an attacker from sending excessive traffic, especially during a BW-DDoS.

Additional network location may be in the cloud or overlay networks. In such solutions, traffic is routed via an over-provisioned cloud service which “scrubs” the attacking flows and forwards only legitimate traffic to the victim.

C. Infrastructure Adaptations

A concern that may affect the ability to deploy a BW-DDoS solutions, is the amount of changes that the infrastructure has to undergo. For example, some solutions require installing new software at end-hosts, some require software updates to routers, while other solutions may be satisfied with mere configuration of networking equipment. Additional changes may take place by using overlay networks or in the cloud.
<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Response</th>
<th>Location</th>
<th>Infra. Adapt.</th>
<th>Cooperation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress Filter</td>
<td>filter</td>
<td>router</td>
<td>configuration</td>
<td>standalone</td>
</tr>
<tr>
<td>ACL</td>
<td>filter</td>
<td>router</td>
<td>configuration</td>
<td>standalone</td>
</tr>
<tr>
<td>RTBH</td>
<td>filter</td>
<td>router</td>
<td>configuration</td>
<td>standalone</td>
</tr>
<tr>
<td>Capabilities</td>
<td>rate-limiting</td>
<td>dst, router, src</td>
<td>router</td>
<td>inter AS (BGP)</td>
</tr>
<tr>
<td>PSP</td>
<td>rate-limiting</td>
<td>router</td>
<td>router software/IP fields</td>
<td>intra AS</td>
</tr>
<tr>
<td>Pushback</td>
<td>filter</td>
<td>router</td>
<td>router software</td>
<td>inter AS</td>
</tr>
<tr>
<td>BTT</td>
<td>rate-limiting</td>
<td>router</td>
<td>configuration</td>
<td>inter AS</td>
</tr>
<tr>
<td>RON</td>
<td>detour</td>
<td>src, cloud</td>
<td>end-hosts software/cloud</td>
<td>end-host and overlay</td>
</tr>
<tr>
<td>SOS</td>
<td>filter</td>
<td>src, cloud</td>
<td>end-hosts software/cloud</td>
<td>end-host and overlay</td>
</tr>
<tr>
<td>Scrubbing</td>
<td>filter</td>
<td>router, cloud</td>
<td>configuration, cloud</td>
<td>inter AS (BGP), cloud</td>
</tr>
<tr>
<td>QoSDefS</td>
<td>break-through</td>
<td>src, dst</td>
<td>end-hosts</td>
<td>end-hosts</td>
</tr>
<tr>
<td>LOT</td>
<td>filter, rate-limiting</td>
<td>router</td>
<td>router software</td>
<td>inter LOT-routers</td>
</tr>
</tbody>
</table>

There are several concerns that ISPs have about deploying BW-DDoS mitigation solutions. One concern is that it is very difficult to estimate what will be the impact of deploying different solutions. Also, ISPs don’t feel the actual problem – DoS attacks disables their customers not themselves and it is very rare for a given victim. However, we can assume that some changes are easier to make than others. For example, configuration changes should be significantly easier to make than most other changes, as the existing software and hardware are used. Next, software changes at end-hosts are assumed to be more difficult than configuration. Finally, we assume that any change to routers, that is software/firmware and especially hardware are very difficult to make and widely deploy.

### D. Cooperation Schemes

Pushback schemes focus on pushing the attack back, away from the victim and closer to its source. This is done by sending requests to upstream routers, asking them to filter the identified offending flows. The cooperation may be intra AS, inter AS, between end-hosts or with an overlay network or a cloud service. Other solutions may be standalone solutions and require no cooperation.

In Pushback [22], the victim identifies the attacking flows profile, followed by “pushing the attack back”, and freeing the victim’s resources to handle legitimate traffic. FlowSpec (RFC 5575), describes an operational implementation similar to key ideas in Pushback. Basically, Pushback and FlowSpec are ACL-like filtering scheme, but instead of having the ACL entries employed within a single AS, they are distributed and pushed back upstream.

Pushback based solutions allow under-provisioned nodes to filter offensive traffic away from the victim. However, the assumption that victim nodes can identify the attack profile might prove very difficult. Furthermore, like other ACL schemes, many Pushback requests leading to many filtering rules and ACL entries, may result in a DoS attack on routers’ processing capabilities. This decoy attack could exhaust filtering rules, followed by an attack on the real target. Alternatively, this type of cooperation may let an attacker issue Pushback request, disconnecting the victim.

BTT is also a Pushback scheme, however, it is less prone to attack than Pushback, as BTT only requests shaping from upstream BTT nodes, which can get down to their typical rate, but should never be less than that. Hence, BTT quality of service may be degraded, but not completely denied.

Finally, cooperation based schemes, such as Pushback and BTT, assume that the cooperating nodes are honest with each other in the sense that they will only propagate upstream requests upon BW-DDoS, which is something which is debatable. Alternatively, the signaling plane between cooperating nodes may by itself be a target for an attack.
IV. CONCLUSION

BW-DDoS consist of the majority of DDoS attacks [2], which in turn are found at the top of the security concerns of Tier 1, Tier 2, and other IP network operators around the world [1]. In this paper, we have reviewed various BW-DDoS attacks and defenses. So far, BW-DDoS employed relatively crude, inefficient, ‘brute force’ mechanisms. However, known attacks, which are not commonly used, allow attackers to launch sophisticated attacks, which are difficult to detect and may considerably amplify attackers’ strength. Furthermore, recent largest BW-DDoS attacks indeed used more advanced techniques; this may indicate that attackers may adopt more effective, advanced BW-DDoS attacks in the future.

We argue that deployed and proposed defenses may struggle to meet the increasing threats, hence more advanced defenses should be deployed. This may involve some proposed mechanisms (not yet deployed), as well as new approaches. Note that some of the proposed defenses, may raise operational and political issues; these are beyond the scope of the current manuscript, but should be carefully considered. Finally, we argue that for defense mechanism to be practical it has to meet two key features, i.e. the mechanism should be easy to deploy and require minor changes, if any, to the Internet’s core routers.

REFERENCES