Size-Based Scheduling: A Recipe for DDOS?

Abdul Serwadda
Louisiana Tech University
Ruston, LA 71272
Louisiana, USA
ase007@latech.edu

Vir V. Phoha
Louisiana Tech University
Ruston, LA 71272
Louisiana, USA
phoha@latech.edu

Idris A. Rai
Makerere University
P.O Box 7062
Kampala, Uganda
rai@cit.mak.ac.ug

ABSTRACT

Internet traffic measurements have shown that the majority of the Internet's flows are short, while a small percentage of the largest flows are responsible for most of the bytes. To exploit this property for performance improvement in routers and Web servers, several studies have proposed size-based scheduling to offer preferential treatment to the shortest flows. In this work, we present analytical and simulation results which confirm that size-based scheduling will ease the task of launching DDOS attacks on the Internet.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General-*Security and protection*

General Terms

Performance, Security

Keywords

Denial of Service, LAS, SRPT, TCP

1. INTRODUCTION

The term Size-based Scheduling (SBS) refers to scheduling policies which make use of job size information when making scheduling decisions. In the Internet, SBS policies have been proposed to offer preferential treatment to short flows while ensuring that long flows do not starve. Prominent examples include proposals for the use of Shortest Remaining Processing Time (SRPT) in Web servers [3, 4] and the use of Least Attained Service (LAS) in routers [7] and wireless LANs [9]. Other hybrid size-aware policies have also been proposed [2]

Unfortunately, owing to a priority mechanism which segregates between connections by virtue of their size/age, SBS has the potential to facilitate a previously unknown type of DDOS against Internet systems. In this new type of DDOS, an attacker aiming to overrun an SBS target (server or router) would bombard it with packets from very short flows, exploiting the priority mechanism and subsequently incurring a much lower cost than would be the case under the legacy First-In-First-Out (FIFO) or Processor sharing (PS) schedulers in routers and servers respectively.[1]

By taking advantage of the priority mechanism, the attacker's packets always go to the head of the queue, avoiding the baggage of having to overflow the router/server buffers. This is the reason as to why the attacker can use a lower attack-traffic rate (a.k.a lower cost) than would be required against a FIFO or PS scheduler. Moreover, this exploitation of the priority mechanism (and reduced cost of attack) also applies to existing attacks such as SYN floods which use a single packet to open each connection.

As revealed by a recent study [1], a large number of address blocks still permit IP address spoofing, owing mostly to inconsistent application of ingress and egress filtering. This serves important evidence that attacks in which an attacker may seek to cycle through a range of IP addresses, with short sequences of packets being disguised as distinct flows can still be realistically launched today.

We illustrate SBS's inherent vulnerability to DDOS attacks. Although our research focuses on LAS in comparison to FIFO scheduling in routers, we believe that all SBS schemes that favor short flows should not deviate much from the behavior of LAS under the kinds of attacks studied.

Over the past decade, several papers have confirmed that SBS offers answers to a number of performance deficiencies in Internet systems. However, to the best of our knowledge, *this is the first paper to provide an insight into a major security vulnerability of SBS in the Internet.* The contribution of this work is summarized below:

- First, we expose a new mechanism of DDOS attack that may arise out of the implementation of SBS in the Internet.

- Second, we show that to completely shut down a FIFO router in a *Mice DDOS attack*, the minimum required sending rate of attack traffic *must at least exceed* that required to shut down a LAS router

- Third, we show that even for attack traffic rates which are not high enough to completely shut down FIFO/LAS routers, the amount of service degradation seen at LAS is always greater or equal to that seen under FIFO. This, and the second contribution underline why it may be much easier for attackers to compromise the quality of Internet communications if LAS/SRPT ever get deployed.

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

[1] Because it uses very short flows, we use the terms *Mice DDOS attack* and *Mice attack* to refer to this attack in the subsequent sections of the of the paper

2. OPERATION OF FIFO AND LAS

Under FIFO, packets are processed in the order in which they arrive at the router queue. When a packet arrives at a full queue, it is dropped.[2]

Under LAS, the next packet to be served belongs to the flow that has received the least amount of service. When the buffer is full, the packet dropped belongs to the flow that has received the most service. Thus if a connection uses only a few packets (such as in attacks using extensive spoofing), these packets will always be treated as priority packets.

2.1 Difference in Attack-traffic Rates

Assume an attacker who perpetually cycles through a range of IP addresses to disguise each packet as a new flow[3]. For a *Mice attack* launched against a LAS router in this way, each of these packets will be seen as the first packet of a connection, and will be treated with highest priority. If the LAS router operates at a line speed of C bits/s (a.k.a service rate of the router), it is trivial to note that an attack-traffic rate of C bits/s suffices to completely stop legitimate flows from accessing the service of the router. The first packet of each legitimate flow would also be processed, however, this is a negligible amount of service as compared to the size of a flow. Below, we show that the attack-traffic rate required to *completely* deny access to a FIFO router exceeds the $Cbits/s$ required against LAS.

Theorem 1. *For a FIFO router operating at a line rate of C bits/s, and traversed by TCP flows of mean round trip time $\overline{RTT}$, the minimum required attack-traffic rate R to completely shut down the router in an amount of time T_1 is $R = \frac{C.\overline{RTT}(1-\alpha)}{T_1} + C$, where $\alpha \approx 0$ is a fraction representing the proportion of legitimate UDP traffic that still finds a free slot in the buffer despite the attack.*

Proof. Assume that the router is traversed by only UDP and TCP traffic. This is reasonable considering that other protocols continue to occupy less than 1% of the Internet's total transport protocol mix [6]. By the bandwidth-delay product, the router buffer size is $C\overline{RTT}$. Suppose the first attack packets arrive at the router queue at t=0, and the router buffer is completely overrun by attack-traffic at t=T_1. During the interval [0, T_1), the proportion of the buffer occupied by legitimate packets may vary due to arrival of new flows and (or) exit of completed and timed-out flows. Let $\beta \in (0,\ 1)$ represent the *average* buffer proportion occupied by legitimate packets in the interval [0, T_1). At $t = 0$, the buffer space to be filled by attack-traffic may hence be estimated as $C.\overline{RTT} - \beta.C.\overline{RTT} = C.\overline{RTT}(1-\beta)$. To fill this proportion of the buffer, an attacker must send traffic at a rate $R > Cbits/s$ for a time t_1, such that $t_1(R-C) = C.\overline{RTT}(1-\beta)$. The buffer is filled up after a duration approximately given by

$$t_1 = \frac{C.\overline{RTT}(1-\beta)}{R-C} \qquad (1)$$

At $t = t_1$, legitimate TCP flows encountering a full buffer begin to time out with high probability since they start to lose packets. As some TCP connections timeout, the buffer space to be filled by the attacker increases, since the router is still sending out packets at the line rate, $Cbits/s$. As attack-traffic occupies the space vacated by the timed out TCP connections, the buffer again gets filled up and more TCP connections timeout with high probability on encountering a full buffer. Eventually, all TCP connections will time out while legitimate UDP flows continue to occupy the proportion $\alpha < \beta$.

On average, during the interval between t=t_1 and the instant when the last legitimate TCP connection is forced to time out, the attacker needs to fill the proportion $(\beta - \alpha)$ left behind by the departing TCP flows. Following the approach used in Equation 1, this proportion is filled in time t_2, where

$$t_2 = \frac{C.\overline{RTT}(\beta-\alpha)}{R-C} \qquad (2)$$

The total time $(t_1 + t_2)$, required to overrun the buffer may be estimated as: $T_1 = \frac{C.\overline{RTT}(1-\alpha)}{R-C}$, where $\alpha \approx 0$ for $t \geq T_1$, since legitimate UDP flows lose a lot of packets at the full buffer. The attack-traffic rate required to initiate full DOS is thus always greater than C, and is approximated as:

$$R = \frac{C.\overline{RTT}(1-\alpha)}{T_1} + C \qquad (3)$$

□

After sustaining the rate R bits/s for a time T_1 seconds, a reduced rate of C bits/s is enough to keep the buffer full.

These results confirm that a minimum attack-traffic rate of C bits/s can completely deny access to a LAS router but cannot achieve the same feat against a FIFO router. As shown by Equation 3, the difference between R and C is only significant if T_1 is small. As an example of a case for which R$\gg$ C, we discuss the Shrew attack [8, 5], a router DDOS attack that strictly demands a value of $T_1 \leq \overline{RTT}$.

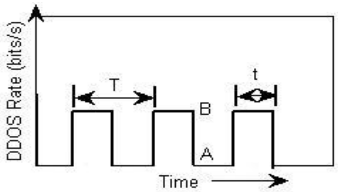


Figure 1: Mechanism of the Shrew attack

2.1.1 Case Study 1: The Shrew Attack

Figure 1 illustrates a Shrew attack with period T (typically$\approx$ 1s). In each attack burst, attack traffic overflows the router buffer, forcing TCP senders to timeout. In between bursts the attacker sends no traffic. AB represents the *peak sending rate* which the attacker must realise almost instantaneously. Traffic is sent at this peak rate for 20 - 200ms [8], an interval which approximates mean Internet RTTs . This interval includes both the time to fill the buffer (equal to T_1 in Equation 3), and the duration of buffer occupancy (t).

As shown in Equation 3, to fill the buffer in time $T_1 \leq \overline{RTT}$, the value of R (at FIFO router) must be a large multiple of C, since $\alpha \approx 0$. Comparing to the case of LAS, a rate of $Cbits/s$ using the *Mice* mechanism is enough to carry out the Shrew attack at a router operating at $Cbits/s$. At this rate, all legitimate packets get queued (and/or) dropped

[2] We focus on FIFO/Drop-tail since the attacks we discuss utilize extensive spoofing and are highly distributed and thus cant be throttled by Random Early Detection (RED)

[3] In practice several packets may comprise a flow

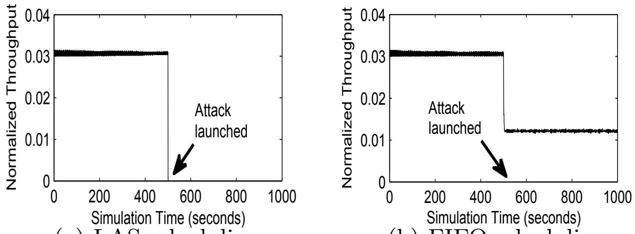
at the router buffer. If attack-traffic arrives at $Cbits/s$ for about an $\overline{RTT}$, legitimate flows have their packets trapped for about an $\overline{RTT}$, the approximate burst length required for the Shrew attack. These trapped packets will then trigger most TCP end-systems to timeout, since they are under the illusion that packets have been dropped.

2.1.2 Case Study 2: Ordinary Bandwidth Attacks

During ordinary bandwidth attacks, the attacker's traffic needn't necessarily satisfy Equation 3, since the aim may be degradation of a router's performance, as opposed to completely shutting down the router. The attacker depends on the combination of the attack traffic load, and the load offered by legitimate traffic to cause a router overload, which in turn causes service degradation to legitimate traffic.

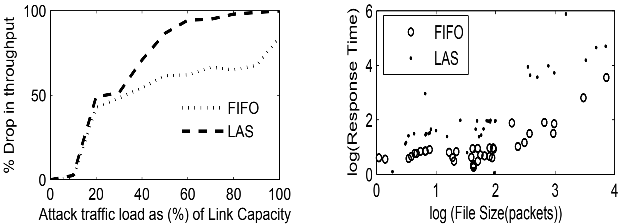
Under LAS (or any SBS scheduler), during times of overload, its the lowest priority packets which are dropped. For an attacker using the *Mice* mechanism, all attack packets have the highest priority (since each attack-flow uses only a few packets). Hence for regular bandwidth attacks conducted in this way, it is the legitimate flows which see losses when the link gets overloaded, since most of their packets are seen as low priority packets. In cases when the link is only moderately loaded, it is still the legitimate flows which see increased delays, since their low priority packets are queued. Since TCP uses delays and loss events to set its sending rate, we conjecture that for a DDOS attack using priority packets, any attack-traffic rate causes more performance degradation under LAS than under FIFO where the delays and losses would not be restricted to only the legitimate flows.

To verify this argument, we simulated a simple dumbbell topology in which 1000 UDP attack sources bombarded a 155Mbps (OC3) link which was also being traversed by 50 legitimate flows. To implement spoofing of attack packets, we designed a variant of the UDP traffic agent in which sequence numbers for all generated packets were always zero, to create the illusion of each packet being the first packet of a flow. Note that in ns2, UDP packets also have sequence numbers, and the ns2 LAS implementation uses the packet sequence number to determine the age of the flow to which the packet belongs. Figure 2(a) shows that when the DDOS attack was launched at t=500s, the studied TCP flow had its throughput reduced to zero at the LAS router, yet it retained about a third of its mean throughput when the scheduling policy was FIFO (Figure 2(b)). The attack traffic rate was equal to the router line rate (155 Mbps), and the total link load before attack was 30 % of the link capacity. Figure 3(a) shows the relationship between attack traffic load and the throughput reduction for one of the legitimate flows in another set of experiments. Before the attacks were launched, the load due to the legitimate flows was 90% of the link capacity. This choice of heavy load was meant to simulate a scenario of a busy link being attacked. Results confirmed that LAS suffered more severe throughput degradation than FIFO for all attack traffic rates.



(a) LAS scheduling (b) FIFO scheduling

Figure 2: Throughput for a single TCP flow (before and after attack) as a fraction of Link Capacity



(a) Long lived TCP traffic (b) HTTP traffic

Figure 3: Effects of DDOS on TCP traffic

In the experiments on HTTP traffic, the Web traffic model in [5] was used. Clients downloaded web pages from randomly selected web sites, with each page containing several objects. The object sizes followed a Pareto distribution with shape parameter 1.2, and the inter-page and inter-object time distributions were exponential with means 9 seconds and 1 milli seconds respectively. While the Web transfers were occuring, the access router to the servers was bombarded with *Mice DDOS* traffic at a rate equal to 60% of the link capacity (same as the load before attack). During the attack, we measured and averaged the response times for different sized objects, before normalizing them out of the response times seen without DDOS. Figure 3(b) shows the results, which confirm the earlier observation of LAS suffering worse degradation than FIFO under DDOS.

3. CONCLUSION

We have exposed a security weakness of SBS. Our work compliments the general understanding of SBS and stimulates research on the design of secure SBS schemes.

4. REFERENCES

[1] R. Beverly and S. Bauer. The spoofer project: Inferring the extent of source address filtering on the internet. In *Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet (SRUTI) Workshop*, pages 53–59, July 2005.

[2] M. Bottigliengo, C. Casetti, C.-F. Chiasserini, and M. Meo. Short-term fairness for tcp flows in 802.11b wlans. In *INFOCOM*, 2004.

[3] X. Chen and J. S. Heidemann. Preferential treatment for short flows to reduce web latency. *Computer Networks*, 41(6):779–794, 2003.

[4] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal. Size-based scheduling to improve web performance. *ACM Trans. Comput. Syst.*, 21(2):207–233, 2003.

[5] A. Kuzmanovic and E. W. Knightly. Low-rate tcp-targeted denial of service attacks: the shrew vs. the mice and elephants. In *SIGCOMM '03*, New York, NY, USA, 2003.

[6] D. Lee, B. E. Carpenter, and N. Brownlee. Observations of udp to tcp ratio and port numbers. *International Conference on Internet Monitoring and Protection*, 1:99–104, 2010.

[7] I. A. Rai, E. W. Biersack, and G. Urvoy-Keller. Size-based scheduling to improve the performance of short tcp flows. *IEEE Network*, 19(1):12–17, 2005.

[8] A. Shevtekar and N. Ansari. A router-based technique to mitigate reduction of quality (roq) attacks. *Computer Networks*, 52(5):957–970, 2008.

[9] Q. Wu, M. Gong, and C. L. Williamson. Tcp fairness issues in IEEE 802.11 wireless lans. *Comput. Comm.*, 31(10):2150–2161, 2008.