Low-Rate TCP-Targeted DoS Attack Disrupts Internet Routing

Ying Zhang
University of Michigan
wingying@umich.edu

Z. Morley Mao
University of Michigan
zmao@umich.edu

Jia Wang
AT&T Labs–Research
jiawang@research.att.com

Abstract

Compared to attacks against end hosts, Denial of Service (DoS) attacks against the Internet infrastructure such as those targeted at routers can be more devastating due to their global impact on many networks. We discover that the recently identified low-rate TCP-targeted DoS attacks can have severe impact on the Border Gateway Protocol (BGP). As the interdomain routing protocol on today's Internet, BGP is the critical infrastructure for exchanging reachability information across the global Internet. We demonstrate empirically that BGP routing sessions on the current commercial routers are susceptible to such low-rate attacks launched remotely, leading to session resets and delayed routing convergence, seriously impacting routing stability and network reachability. This is a result of a fundamental weakness with today's deployed routing protocols: there is often no protection in the form of guaranteed bandwidth for routing traffic. Using testbed and Internet experiments, we thoroughly study the effect of such attacks on BGP. We demonstrate the feasibility of launching the attack in a coordinated fashion from wide-area hosts with arbitrarily low-rate individual attack flows, further raising the difficulty of detection. We explore defense solutions by protecting routing traffic using existing router support. Our findings highlight the importance of protecting the Internet infrastructure, in particular control plane packets.

1 Introduction

There is evidence of increasing occurrences of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks on the Internet today [40]. Most of the widely known attacks target a single host or multiple hosts within a particular edge network, rather than the Internet infrastructure such as routers inside transit ISP networks. The latter type of attack can be quite devastating. For example, attacks against routers can impact significant amount of traffic, as many networks rely on them to reach other destinations. Moreover, attacks on the routing infrastructure can create partition between lower tier ISPs to the rest of the Internet by bringing down several links simultaneously. Thus, it is important to understand attacks against the Internet infrastructure given its critical importance to the well-being of the Internet. In this paper, we focus on examining a particular type of attack against the interdomain routing protocol – the Border Gateway Protocol [39].

The Border Gateway Protocol (BGP), the de facto standard Internet interdomain routing protocol, uses TCP as its transport protocol. A fundamental flaw with routing protocols deployed today is that there is usually no protection in the form of priorities in using router resources for control plane packets. Thus, congestion of other data traffic can adversely affect BGP packets, as shown in the previous study by Shaikh *et al.* [43]. Recent studies [50, 21, 7] have indicated that data congestion can severely impact routing sessions. Thus, any attack that exploits this lack of isolation with an impact on TCP can negatively affect the functioning of BGP.

In this work, we study how the recently identified low-rate TCP-targeted DoS attacks [27] disrupt interdomain routing on today's Internet. This is the first study that systematically examines the impact of this type of attack on interdomain routing, and we discovered the impact can be quite severe. It has been shown that low-rate TCP attacks can severely degrade TCP throughput by sending pulses of traffic leading to repeated TCP retransmission timeout. Given the fundamental susceptibility of TCP to such low-rate attacks due to its deterministic retransmission timeout mechanism, any application using TCP is vulnerable. In particular, the effect on protocols using TCP within the Internet infrastructure is arguably more severe due to the global scope of the impact. Aside from the potential impact on the throughput of BGP packets, a more critical question is whether such attacks are powerful enough to *reset BGP's routing session* as a result of a sufficiently large number of consecutive packet drops. If the session is reset, it can have serious impact on the Internet in the form of routing instability, unreachable destinations, and traffic performance degradation [29, 28]. Note that attackers can launch such attacks *remotely* from end hosts without access to routers nor the ability to send traffic directly to them. Its low-rate nature makes detection inherently difficult. More importantly, the existing best common practice for protecting the Internet routing infrastructure by disallowing access and research proposals such as SBGP [26] are not sufficient to prevent this type of low-rate attack since this attack is exploring a transport layer vulnerability of BGP.

We show empirically using testbed experiments that today's routers with default configurations are susceptible to BGP session resets as a result of low-rate TCP-targeted DoS attacks. We observe that attackers can bring down the targeted BGP session in less than 216 seconds. Session reset probability can be as high as 30% with only 42% utilization of the bottleneck link capacity. And when the session is not reset, BGP table transfer can be increased from 85 seconds up to an hour with only 27% of the link capacity used. Using wide-area experiments, we show the ease with which coordinated low-rate attacks can be launched, resulting in arbitrarily low-rate individual attack flows. This raises the difficulty of attack detection. Fortunately, major peering links with significant available bandwidth are difficult to attack due to required resources. We subsequently explore defense strategies through prevention and demonstrate that *it is possible to significantly lower the risk of such attacks by prioritizing routing traffic using existing router support.* We provide recommendations for better default BGP configurations.

The rest of the paper is organized as follows. We provide the background of low-rate TCP-targeted DoS attacks and BGP in Section 2. Section 3 discusses impact of such attacks on BGP and key factors in determining vulnerability of BGP. We show using testbed experiments that BGP can be disrupted by low-rate TCP attacks in Section 4. Section 5 shows using wide-area experiments how multiple attack hosts coordinate to launch low-rate attacks against a given BGP session. We discuss defense mechanisms in Section 6 and conclude in Section 7.

2 Background

In this section we describe low-rate TCP-targeted DoS attacks and the Border Gateway Protocol susceptible to it.

2.1 Low-rate TCP-targeted DoS Attacks

In their seminal work [27], Kuzmanovic and Knightly showed that TCP's retransmission timeout mechanism can be exploited by using maliciously chosen low-rate DoS traffic to throttle TCP flows to a small fraction of their ideal rate. As shown in Figure 1, the low-rate attack consists of periodic, on-off square-wave of traffic bursts with magnitude of the peak R, burst length L, and inter-burst period T. There are several requirements for the low-rate TCP-targeted attack to be successful: (i) An integer multiple of the inter-burst period coincides with the minimum retransmission timeout value (minRTO) of TCP. (ii) The magnitude of the attack peak traffic is large enough to cause packet loss. (iii) The burst length is sufficiently long to induce loss: It needs to be longer than roundtrip time (RTT) of TCP flows. When these conditions are satisfied, the aggregate TCP flows sharing the bottleneck link will have close to zero throughput. Even if the inter-burst period takes on other values outside the minRTO range, the throughput can still be severely degraded. The reason is that the TCP retransmission timer repeatedly times out due to loss induced by the attack traffic burst, as the timer value exponentially increases for any given flow sharing the bottleneck link with the attack traffic.

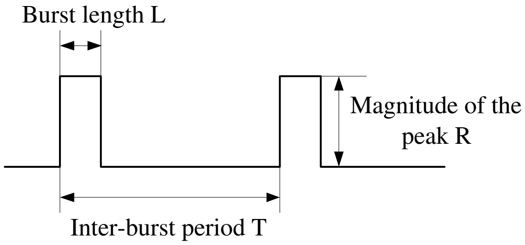


Figure 1. Notation for low-rate TCP-targeted DoS attacks

One way to defend against such attacks is to randomize the minimum retransmission timeout value (minRTO) value; however, this does not fully mitigate the attack due to the inherently limited range for minRTO as shown by Kuzmanovic and Knightly [27]. They also found that even router-assisted mechanisms do not eliminate the attack impact without incurring excessively high false positives. There has also been follow-up work on detecting low-rate attacks [47, 44, 30, 14]. Most of the existing detection algorithms rely on signal analysis. None of the proposed detection algorithms has been shown to be sufficiently accurate and scalable for deployment in real networks. Furthermore, no known solution exists to effectively mitigate such low-rate attacks. Thus, all applications using TCP are inherently susceptible to degraded performance due to such attacks. In this work, we focus on the Border Gateway Protocol as an important "application" using TCP given its critical role as the interdomain routing protocol on the Internet.

2.2 Border Gateway Protocol

The Border Gateway Protocol (BGP) is used as the interdomain routing protocol on today's Internet. In BGP, a routing session is established over a TCP connection between neighboring border routers to exchange reachability information. There are two types of BGP sessions: eBGP and iBGP sessions. The former are between routers within different autonomous systems (ASes) or networks, and usually consist of a single hop, *i.e.,* the two routers are directly connected with a physical link. The latter are within the same AS and can go through multiple router hops.

Because BGP is a stateful protocol, routing information previously received is assumed to be valid until withdrawn. To ensure connection liveness, KeepAlive messages are exchanged periodically. According to BGP's protocol specification [39], each BGP router maintains a *Hold Timer* which limits the maximum amount of time that may elapse between receipt of successive KeepAlive and/or update messages from its neighbor in the BGP session. If the Hold Timer expires, a notification error message is sent and the BGP connection is closed. Upon session reset, all routes previously exchanged in the session are implicitly withdrawn, potentially propagating routing instability to other networks.

Note that one may argue that BGP session reset due to data congestion is actually desirable, given the associated routes are not preferable due to the bad quality of the link. We strongly dispute this claim. Session reset creates significant disruption and can cause global routing instability. Performance based route selection can be used instead. Moreover, ISPs today already perform traffic engineering to load balance traffic.

There are other BGP security problems, such as lack of deployed mechanisms to verify the correctness, authenticity, integrity of the routing information exchanged. Proposed protocols such as SBGP [26], SoBGP [34] address some of these issues. Other attacks against routing protocols such as the link cutting attack described by Bellovin [12] are related. It uses topology information to select specific links to cut so that traffic is rerouted through routers controlled by attackers. The attack described in this paper also uses topology information to identify target links. Router vendors have provided protection against known attacks such as TCP RST and SYN flood attacks [18, 23]. Using testbed experiments we verified none of the routers we tested is vulnerable to TCP RST attacks. Note that unlike RST or SYN flood attacks, it is possible to remotely launch resource-based attacks, such as the attack described in this paper, using packets *passing through* the routers without the ability to send packets destined to them.

3 Low-rate DoS Attacks on BGP

Because BGP runs over TCP for reliability, BGP is also vulnerable to the recently discovered low-rate TCP-targeted DoS attacks. Due to its low-bandwidth property, such attack is much more difficult to detect, and thus it is important to understand it thoroughly. In this paper, we focus on investigating the effect of low-rate attacks on a single-hop BGP session. However, the results can be generalized to multihop BGP sessions. Arguably multihop BGP sessions are more susceptible as they traverse multiple links, thus more likely to experience congestion.

3.1 Impact of Attacks on BGP Sessions

The impact on BGP sessions caused by low-rate TCP-targeted DoS attacks are two fold: *throughput degradation* and *session reset*. First, the throughput of the BGP update messages can be significantly reduced. However, the average BGP update rate is quite low, except during significant routing changes or table transfer upon session establishment. The impact in the form of rate reduction of BGP traffic is less critical, but can further exacerbate the already slow BGP convergence process. The second type of attack impact due to BGP session reset is much more severe. To reset a BGP session, the induced congestion by attack traffic needs to last sufficiently long to cause the BGP Hold Timer to expire. To monitor the attack success, one can analyze traffic traversing the impacted link or routing updates related to the session. Furthermore, it is easier to keep the session down as SYN packets are sent less frequently compared to retransmitted data packets.

BGP session reset can lead to severe churn on the Internet's control plane. This not only impacts both routers involved in the BGP session, as each withdraws all the routes previously advertised by its neighbor, but also many other networks on the Internet due to the propagation of routing changes. For example, the number of routes in a default-free router in the core Internet is around 170,000 based on routing data from RouteViews [5]. A significant fraction of the table can be affected upon a BGP session reset. Withdrawing a large number of routes can cause many destination networks to become temporarily unreachable due to inconsistent routing state [48] and a large amount of traffic to become rerouted, which may further lead to congestion due to insufficient capacity.

A recent proposal to mitigate the potential negative impact of short-lived session resets is termed graceful restart [42]. Routers supporting this mechanism attempt to continue to forward packets using the stale routes. There is, however, an upper bound (by default two or three minutes) on the amount of time a router retains the stale routes to avoid lengthy routing inconsistency. Thus, a session reset that lasts sufficiently long time, possibly due to an intense low-rate attack, can still have severe impact on the data plane.

In general, the impact of an eBGP session reset is larger than that of an iBGP session reset because routing changes received from eBGP sessions are more likely to propagate across multiple networks and the routing table is usually default-free, thus carrying all the destinations to the Internet. The routes exchanged between two routers in an eBGP session consist of all the routes of their respective customers. Thus, for eBGP sessions between two large ISPs, this number can be quite large. We analyzed routing tables from a tier-1 ISP and found that up to 13% of the routing table can come from a single eBGP session versus only 4% from an iBGP session.

3.2 Key Factors in Attacking a BGP Session

We study the key factors that determine the vulnerability of BGP to such attacks to illuminate possible solutions.

1. Priority of routing traffic. The fundamental problem that makes BGP vulnerable to low-rate attacks is that router traffic may not be sufficiently protected from congestion caused by other data traffic. Many of the commercial routers today by default use First-In-First-Out (FIFO) or Drop Tail queueing discipline, giving no priority to routing packets. Even in the case where routing data are protected (*e.g.,* through the RED queue management scheme [22]), there are no default policing mechanisms to prevent attack packets from spoofing packets of higher priority. For example, we observed that many routers will mark the routing packets with an IP precedence value of 6 [8]. However, attack packets can also use the same or even higher IP precedence values given the lack of authentication for such values by default. Packet remarking or TTL value checking [23] can help ensure only routing packets receive higher priority. In this work, we illuminate these issues by experimenting with real commercial routers with various configuration settings. Instead of using simulations, we focus on using experiments to obtain results closer to the reality.

2. Proprietary router implementation. Commercial router behavior is much less understood compared to that of end hosts due to its proprietary nature and lack of source code access. For example, it is unclear how the TCP stack on commercial routers really behaves. Unlike for end-hosts, critical parameters to the attack such as minRTO are unknown, making successful attacks much more difficult. If minRTO is randomized, it would further reduce the probability of a session reset. Even with known router behavior, depending on its configuration, its dynamic behavior may be quite different compared to the default settings. We mainly focus on default settings as most deployed routers probably use default configurations. When we know that the router supports certain features that would help protect against the low-rate attacks, we also examine these features in great details.

3. Capacity of peering links. In order for low-rate TCP attacks to be successful against BGP routing sessions, the traffic burst needs to be sufficiently powerful to cause enough packet loss, so that the TCP flow of the BGP session enters into retransmission timeout state. This may appear to be difficult to achieve, especially for BGP sessions involving Internet core backbone links given the heavily over-provisioned core. However, eBGP sessions involve peering links which may not be as well-provisioned compared to links within an ISP backbone. There has been anecdotal evidence that congestion often occurs on peering links. Previous measurement studies such as [6, 24] have shown that some of the bottleneck links of today's Internet paths occur at the boundary between two networks. Links between stub networks and their providers often have much lower speed, and these networks often use eBGP to obtain routes. Using data from RouteViews [5], we found 23% of 100,482 eBGP peering sessions belong to stub networks. Furthermore, it is not necessary that a single attack host overwhelms the target link. As we show later in Section 5, multiple hosts possibly from a botnet can be used to launch a coordinated attack, as long as they traverse the link involved in the BGP session under attack. In this work, we investigate the necessary conditions and show experimentally how this can be achieved.

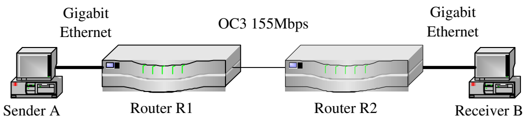


Figure 2. Lab experiment testbed

4 Testbed Experiments

In this section, we describe experiments conducted on a router testbed and empirically show that commercial routers can be severely impacted by low-rate TCP-targeted DoS attacks in the form of session resets and degraded table transfer throughput. We first present our experiment setup, and then inferred TCP characteristics and observed BGP parameters of different commercial routers, followed by detailed analysis of attack impact.

4.1 Testbed Setup

Our experiment testbed consists of two commercial routers and two PCs shown in Figure 2. The two links connecting the routers and the PCs are full-duplex Gigabit Ethernet. The target link between the routers is Packet Over SONET (POS) with 155 Mbps link capacity. Note that our experiment testbed closely models the real operational scenario of an eBGP session with two key differences. First, we do not model background traffic and select the link types to allow traffic from Sender A to Receiver B to easily overload the link between the two routers. Second, attack hosts

Router type	RouterOS version	TCP properties		BGP/Router parameters (default)			
		minRTO (msec)	SYN retry pattern (sec)	KeepAlive (sec)	Hold Timer (sec)	Queue alg.	Graceful restart timer (sec), range
Cisco 3600	IOS 12.2(25a)	300	2,4,8,16,(2).	60	180	FIFO	Not supported
Cisco 7200	IOS 12.2(28)S3	600	2,4,8,16,(152).	60	180	FIFO	Supported, 120, 1-3600
Cisco 7300	IOS 12.3(3b)	300	2,4,8,16,(152).	60	180	FIFO	Supported, 120, 1-3600
Cisco 12000	IOS 12.0(23)S	600	2,4,8,16,(152).	60	180	FIFO	Supported, 120, 1-3600
Juniper M10	JUNOS[6.0R1.3]	1000	3,6,12,24,(30).	30	90	FIFO	Supported, 180, 1-3600

Table 1. Router TCP behavior, router and BGP parameters.

are usually several IP hops away from the target link, with more variable and longer delays to the target link. Longer delays do not affect attack effectiveness, but more variable delays can make attacks more difficult to control. We describe later in Section 5 how all these difficulties can be overcome using coordinated attacks.

The experiment is conducted as follows. A sender program transmits from Sender A UDP-based low-rate attack traffic[1] of the shape shown in Figure 1 with peak rate at 185 Mbps traversing the link between the two routers arriving at Receiver B. The peak rate is set to be 185 Mbps, as it is the lowest rate needed to successfully reset the session with a burst length of 150 ms. With a shorter burst length, we observe that the session does not reset even with a larger peak rate due to insufficient time to saturate the router buffer to cause congestion. When the bottleneck link between the two routers becomes congested, we observe attack packets are dropped at both the input and the output queue of router R_1. Using default router configurations, locally generated BGP packets from R_1 to R_2 also experience packet loss due to shared router buffer. If one of R_1's BGP packet and its subsequently retransmitted packets are all lost causing the Hold Timer to expire, the BGP session is closed.

We experimented with a wide variety of commercial routers using the latest router OS whenever possible from the Schooner testbed [4] and our own local lab. They consist of the following types: Cisco 3600, 7200, 7300, 12000 (commonly known as GSRs), and Juniper M10. To study the extent of the phenomenon, the same experiments were performed on all these routers, and similar results were observed. One main difference is that lower-end routers such as Cisco 3600 have smaller buffers compared to more powerful routers such as Cisco GSRs, making them more vulnerable to attacks. Another difference is that the Juniper M10 is found to be more vulnerable due to its larger minRTO and smaller KeepAlive and Hold Timer values. We emphasize that susceptibility to low-rate DoS attacks is *a general problem with any router* when not configured with ways to prioritize BGP traffic and has a BGP implementation using TCP with a deterministic retransmission timeout mechanism.

In this paper, we use Cisco GSRs with IOS version 12.0 to illustrate our results because they are are commonly used in Internet backbone networks and are the most powerful routers we examined on our testbed. In particular, the Cisco GSRs used are equipped with Cisco 12410/GRP (R5000 CPU at 200 Mhz) processor, 512 KB L2 cache, and 512 MB memory. The line card on the router has a 4 port POS OC-3c/STM-1 Multi Mode with Engine type 0, a buffer size of 12560 packets for packet sizes matching that of BGP packets.

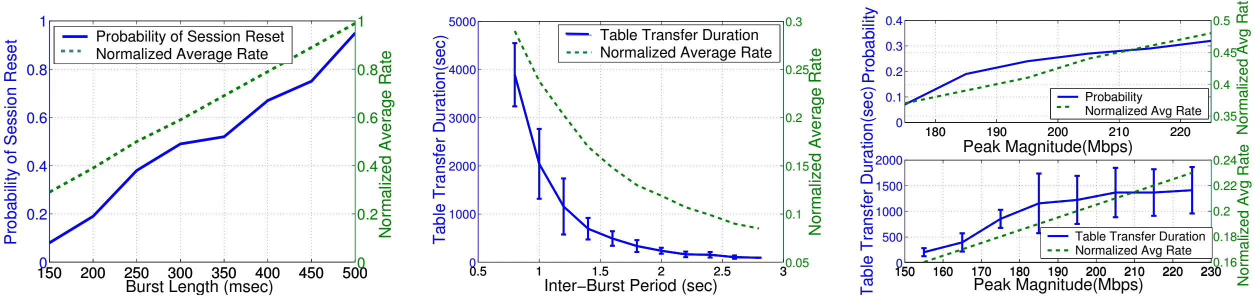
4.2 Router Implementation Diversity

To understand why commercial routers are vulnerable to low-rate attacks, we analyze the TCP behavior and default router configuration settings.

In our work, TCP related parameters are obtained using software we developed based on TBIT (TCP Behavior Inference Tool) [36], which infers TCP properties on Web servers. We enhanced it by integrating BGP-related functionality to establish a BGP session with a commercial router. After the session establishment, the tool constructs packets in special ways to infer router's TCP behavior. The most important TCP property inferred is minRTO which can be accurately determined.

Table 1 also shows the default router BGP configurations for several features relevant to the low-rate attack. Cisco routers use a 60 seconds KeepAlive Timer and a 180 seconds Hold Timer by default, while Junipers have smaller default timer values: 30 seconds for KeepAlive and 90 seconds for Hold Timer. We derive that to reset the BGP session, attackers need to cause at least 8 consecutive packets to be dropped for Cisco GSR and only 6 for Juniper M10 due to the timer values. Thus, Juniper M10 is more vulnerable to low-rate attacks compared to Cisco GSR. The default queuing algorithm for all routers studied is FIFO instead of RED. Weighted RED described later can help protect routing packets. Graceful restart support provided by Cisco [17] is not enabled by default (Graceful restart is not supported in Cisco 3600). It can help routers tolerate short-lived session down time; however, there is a timer limit on the down time, with a default of 2 or 3 minutes, before the stale routes are withdrawn.

[1] UDP is used as opposed to TCP to precisely control the sending rate. TCP packets, without conforming to congestion control can also be used to avoid detection.



(a) Attack burst length
(R=185 Mbps, T=600 ms)

(b) Inter-burst period
(R=185 Mbps, L=200 ms)

(c) Peak magnitude
(Top: T=600 ms, L=200 ms)
(Bottom: T=1.2 second, L=200 ms)

Figure 3. Impact of attack traffic on BGP session reset and table transfer duration: minRTO=600 ms, with Cisco GSRs.

4.3 Experiment Results

As shown in [27], three key factors determine the attack impact: *burst length*, *inter-burst period*, and *peak magnitude*. Intuitively, longer burst length causes the bottleneck queue to be full for longer duration, leading to larger attack impact. Shorter inter-burst period results in larger probability of dropping BGP packets. The larger the attack peak magnitude, the sooner packets fill up the router's queue on the bottleneck link. Attacks with sufficiently high average rate would no longer be considered low-rate. However, such attacks can be composed of many individual low-rate attack flows from a distributed attack described in Section 5.

Next, we analyze attack impact to confirm these intuitions and focus on the attack impact on BGP. We use three metrics to measure the BGP performance under attack: session reset probability, time to reset the session, and BGP table transfer duration. In particular, we conduct four sets of experiments. The results reported in this section are obtained by repeating each individual experiment 100 times.

1. Impact of attack burst length on session reset. In this experiment, we analyze how attack burst length impacts session reset probability. The experiment is set up as follows. Router R_1 periodically sends KeepAlive messages to Router R_2. Sender A starts a low-rate attack with traffic destined to Receiver B with a given burst length, a fixed inter-burst period of 600 ms and peak magnitude of 185 Mbps. Figure 3(a) shows as expected that the session reset probability and the average attack flow rate increase with larger burst length. When the burst length is half of the inter-burst period, the session reset probability is about 50%.

2. Impact of attack inter-burst period on BGP table transfer. As observed above, an attack might not reset the session. When the session is not reset, attack flows can delay updates due to increased queueing and packet loss, resulting in longer BGP convergence delays. We use the BGP table transfer duration as a measure to study the impact of varying inter-burst period of low-rate attacks. In this experiment, we fix the peak magnitude at 185 Mbps and burst length at 200 ms. The smallest inter-burst period is set at 800 ms given minRTO of 600 ms to prevent session reset. We conduct the experiment as follows. First, we load R_1 with a randomly chosen default-free BGP table of 166,527 routing entries obtained from RouteViews [5]. Then, the BGP session between R_1 and R_2 is configured and established. Subsequently within at most 3 seconds of session establishment, Sender A starts the low-rate attack with traffic destined to Receiver B. We record the time when R_2 receives the entire table. Figure 3(b) illustrates the average and standard deviation of BGP table transfer durations with varying inter-burst period. For inter-burst period of 0.8 second and less than 30% average utilization of the link capacity, it takes on average more than one hour to finish transferring the BGP table, which normally lasts only about 85 seconds! As the inter-burst length increases to 1.2 seconds, BGP table transfer still needs 21 minutes to finish on average. However, the impact on BGP table transfer diminishes quickly with increasing inter-burst period.

3. Impact of attack peak magnitude on session reset and table transfer. To evaluate the impact of peak magnitude on session reset probability, we fix the burst length at 200 ms and the inter-burst period at 600 ms matching the minRTO value. Cisco GSR allocates buffer space in chunks of varying sizes, matching packets of different sizes. We use a fixed attack packet size of 30 bytes, of similar size to the KeepAlive packets, so that they are placed in the same buffers. We vary the peak magnitude from 175 Mbps to 225 Mbps by changing the sending rate. 175 Mbps is chosen as it is the lowest rate needed to reset the session for our setup. As shown in the top plot of Figure 3(c), the session reset probability increases gradually with larger attack

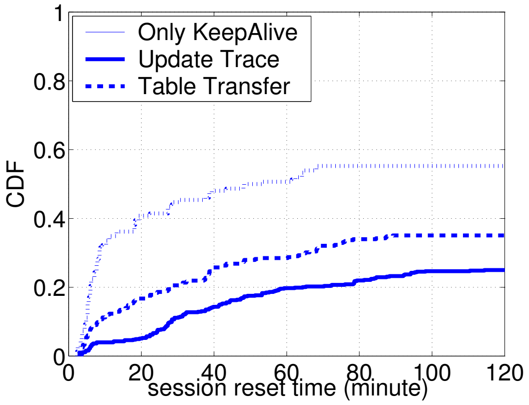


Figure 4. Attack duration to cause session reset with Cisco GSRs (T=600 ms, L=200 ms, R=185 Mbps, minRTO=600 ms).

rate. Based on simple calculations, increasing peak magnitude from 175 Mbps to 215 Mbps (or the excess bandwidth relative to the bottleneck link from 20 Mbps to 60 Mbps) reduces the time to fill up the 3 Mbit queue[2] from 150 ms to 50 ms. This effect is equivalent to increasing the burst length by 100 ms, shown in Figure 3(a). The impact of peak magnitude on BGP table transfer duration was analyzed by fixing the burst length at 200 ms and the inter-burst period at 1.2 seconds, intentionally chosen to be different from the minRTO value to prevent session resets. Attack UDP packet size is set to 1,500 bytes, matching the size of BGP packet for table transfer. The attack peak rate varies from 155 Mbps to 225 Mbps. The bottom plot of Figure 3(c) shows as expected that with increasing peak rate, BGP table transfer duration gradually increases.

4. Impact of BGP update behavior on session reset. The BGP session is brought down as soon as the Hold Timer expires. This requires losing all the BGP packets for a duration at least as long as the Hold Timer value. To trigger the first set of packet loss, a BGP packet must encounter congestion possibly induced by the attack traffic. To cause retransmission timeout, the attack flow needs to cause all the packets within one TCP window to be dropped to avoid TCP fast retransmission. If BGP packets are exchanged infrequently and happen to always miss congestion, it will be impossible to reset the session. Thus, the more frequently BGP messages are exchanged, the more likely the BGP session is reset given a regularly occurring low-rate attack pattern. In all the above experiments, we focus on the overly conservative scenario where only KeepAlive messages are exchanged. In reality, routers frequently send updates associated with routing changes. Thus, "busier" routers, routers with larger BGP tables, containing more unstable routes, are more likely impacted.

We examine the attack duration needed for session reset in the following three scenarios with increasing BGP message frequency. (i) Only KeepAlive messages are sent (every 60 seconds). (ii) One typical day's BGP update trace from RouteViews is played back. (iii) One default-free table from RouteViews is transferred. The scenario (ii) is the common case. Figure 4 shows the distribution of the attack duration needed for session reset (cut off at 2 hour time limit). As expected, on average it takes the least time to reset the session for scenario (iii) when updates are most frequently exchanged. Scenario (i) requires on average more time compared to the other scenarios. In the best case, it takes only 216 seconds to reset the session.

To summarize, the experiments described above analyze in detail how various parameters of low-rate attacks affect the attack effectiveness on BGP. We observe that increasing burst length and peak magnitude, and reducing inter-burst period increase attack effectiveness. These results confirm the danger that a low-rate attack can reset a BGP session.

4.4 Router Architecture: Explaining Packet Drops

The router architectures from different vendors and types vary in details; however, packet drops occur whenever any buffer becomes full with the default FIFO queuing. There are usually multiple buffers in a router, traversed by forwarded packets. BGP packets share with attack traffic and other background traffic some of these buffers. More specifically, locally generated BGP packets by the router shares with other packets the buffer space on the output interface. BGP traffic forwarded by the local router, in the case of iBGP sessions, experiences resource sharing in all these buffers with other traffic. *Thus, protection for BGP traffic needs to be provided at both incoming and outgoing interfaces.*

In our experiments, BGP traffic is associated with a single-hop eBGP session and thus is locally generated. Some attack traffic is observed to be dropped at the input interface, and we observe that BGP packets experience loss at output queues. Under the default configurations, there is no protection for the BGP traffic which thus competes for buffer space with all other traffic. Recent proposals [11, 10] on reducing router buffer size partly to improve delays may endanger BGP packets, as transient congestion either intentional through attacks or unintended by regular traffic can result in BGP packet loss, especially with smaller buffers.

5 Coordinated Low-rate Attacks

In previous sections, our focus was on a simplified network setting with two topological advantages from at-

[2] The queue size of 3 Mbit is derived from a buffer of 12,560 packets with 30 byte packets.

tacker's view point: (i) The network path p from the attacker to a selected destination host (which the attacker is not required to have access to) goes through at least one link l of the BGP session under attack. (ii) The bottleneck link for the path p is link l between the two BGP routers involved in the session. Besides, the attack hosts need to send sufficient amount of burst traffic to congest the bottleneck link l but with sufficient inter-burst period to avoid detection, which requires correctly guessing minRTO and estimating target link capacity. However, these restrictions are difficult to impose, as the link of interest may be close to the core network, likely with much higher bandwidth compared with the case of attacking end hosts. We now explore how attackers can succeed *without these conditions*.

One way to overcome these difficulties is to launch a coordinated attack with multiple attack hosts. Attackers can identify hosts whose network paths for selected destinations traverse the links involved in the BGP session and synchronize attack flows to avoid detection. However, the bottleneck link does not need to be shared, as long as the combined attack flows is sufficient to overload the link. Furthermore, each attack flow does not even need to match router's minRTO. Attack hosts can be used to send *overlapping traffic bursts*. The overlap can occur both in attack amplitude as well as occurrence in time. The feasibility of such coordinated low-rate attacks depends on sufficient number of attack flows, the target link's available bandwidth, and the time synchronization granularity among different hosts.

In the following, we focus on the algorithm for two key steps in launching coordinated low-rate attacks, attack host selection and time synchronization, in completing such coordinated low-rate attacks given a link of interest l involved in a BGP session. Guessing minRTO and link capacity accurately is out of the scope of this paper. Instead, we randomly guess minRTO and estimate link capacity using existing tools such as Pathneck [24]. We then demonstrate the feasibility using wide-area experiments, including an actual attack performed against a locally constructed BGP session using wide-area attack hosts.

5.1 Attack Host Selection and Synchronization Algorithm

There are two key steps in completing such coordinated low-rate attacks given a link of interest l involved in a BGP session: (i) Selecting hosts whose network paths to chosen destinations traverse l. (ii) Synchronizing attack traffic sending time so that attack traffic arriving at l follows the desired square wave pattern. The first step is nontrivial, as unlike end hosts, IP interface addresses associated with the target link l cannot be directly reached from end hosts. For protection, most networks do not globally advertise the infrastructure addresses used to number network equipment such as routers. The second step is not strictly necessary, as the aggregate attack flow can be consistently high-rate to overload the target link. However, it is useful to make detection more challenging.

5.1.1 Selection of Attack Hosts and Destinations

We assume that an attacker has identified a target link l involved in a BGP session between two routers on the Internet.[3] The BGP session can be any of the following: an eBGP session between two ISP peers or a customer and its provider, or an iBGP session within an AS. Here we only focus on eBGP sessions as the impact of such session resets is generally larger compared to iBGP sessions. We highlight the key steps in selecting attack hosts such that for their selected destination hosts the network paths traverse the link l. In the following we illustrate the steps to select attack hosts which needs to be repeated over time due to routing changes.

1. Identify the target link's geographic location and AS(es). Given the target link l denoted by its two router interface IP addresses, an attacker can map them to their ASes [33] and the approximate geographic location [37, 45]. We denote the IP addresses as IP_1 and IP_2, and the associated AS numbers as AS_1 and AS_2.

2. Identify host to destination prefix pairs whose path traverses the link of interest at AS level. We first identify at AS level host to prefix pairs whose network paths traverse the AS_1-AS_2 link from the BGP data of either the attack hosts' local network or other public sources as RouteViews [5] and RIPE [3]. By taking advantage of destination-based forwarding, we find prefixes whose AS path contains the following pattern $[AS_x \cdots AS_1\ AS_2 \cdots]$, where AS_x is the upstream provider or the origin AS of the attack host, and "$\cdots$" denotes zero or more ASes.

3. Identify IP-level paths. To select the ones whose paths traverse IP_1-IP_2 given AS level path, we traceroute from each attack host to a randomly selected IP from each of its destination prefixes to check if l is traversed. To reduce probing overhead, we can further narrow down the attack hosts by selecting those that are geographically close to the link l. This is especially useful if the target link is between two peers, which usually uses the hot-potato or early exit routing strategy. Known IP aliasing resolution techniques [45] can be used to identify interfaces belonging to the same router to find more paths traversing the target link.

5.1.2 Time Synchronization

Time synchronization ensures that aggregate attack flows from diverse hosts arriving at the target link l follow the

[3] Similar to the minRTO discovery, we do not discuss how to identify such links, which can be achieved using network topology information.

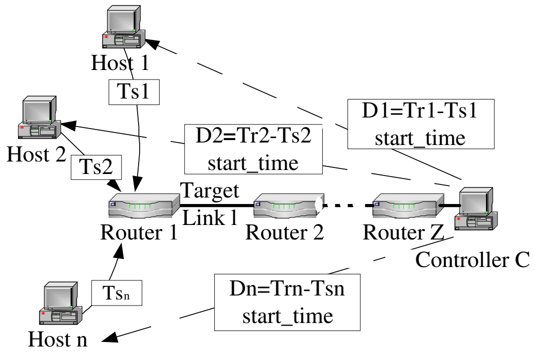


Figure 5. Synchronizing coordinated low-rate attacks.

square wave shape for maximal attack efficiency, making attack source detection more challenging. The differences in each host's local clock and the network delay from each host to l necessitate time synchronization. Next we describe our algorithm for attack synchronization amongst n attack hosts, as illustrated in Figure 5. The idea of using one control host or router at the other end of the path to synchronize is similar to what was proposed in [13].

1. Select a reference time. Attackers need to select one local clock as the reference time for computing the relative time to launch the attack. If attackers have access to a controller host C, such that network paths from each attack host to C traverse l as shown in Figure 5 and also match the actual attack network path, C's local clock can be used as the reference time. C records the arrival time of packets from each attack host. Alternatively, an attacker can use a router Z, such that network paths from each attack host to Z traverse l. ICMP timestamp replies from Z [32, 9] serve as the reference time. If no such router is found, the attacker might find a live destination and send to it ICMP or IP timestamp request [1]. Ideally, the host or router serving as the reference time should be close to l, so that the delay variability of the network path segment from l to the host or the router will have minimal effect on synchronization.

2. Synchronize with the reference time. Once the controller C is identified, each host H_x sends a packet to C, embedding the sending local time Ts_x as the payload. C records the receiving time Tr_x based on its local time. Then C computes the time difference: $D_x = Tr_x - Ts_x$, capturing the difference in both the local clocks and the network one-way delays from H_x to C. After C receives a packet from each host, C obtains n time difference values $D_1, D_2, \ldots, D_n$. The controller C decides to start the attack at time $start\_time$ based on its local clock, allocating sufficient time for the packet to be received by each host. Subsequently, C sends a message to each host H_x with the value of D_x and $start\_time$. Receiving the message, host H_x starts attack at time $start\_time - D_x$ based on H_x's local clock. Alternatively, if router ICMP timestamps or host ICMP/IP timestamps are used as the reference time, upon receiving ICMP timestamp replies, the information needs to be aggregated to coordinate the attack starting time. Even if the attacker does not know the exact minRTO, the controller can randomize the $start\_time$ on each host within a range to cause overlapping in the aggregated flow on the target link.

5.2 Case Studies: Wide-Area Experiments

Our wide-area experiments illustrate the feasibility of selecting attack hosts and synchronizing coordinated attacks targeted at various types of links using the process outlined above. We further demonstrate the feasibility of coordinated attacks using Internet experiments.

Practical difficulties in executing the attack. There are some practical challenges in successfully executing the attack. Identifying a link involved in a target BGP session requires network topology information. Router IP aliasing, parallel links, and inconsistency between BGP paths and IP forwarding paths can complicate the process of identifying attack hosts and destinations. To estimate the number of attack hosts needed, the attacker needs to assess the available bandwidth of the target link. Moreover, attack flows may share common bottleneck links, reducing the overall attack rate. Attackers could be limited by the attack hosts they control, which may not be sufficient to overload the target link, due to limited bandwidth and the inability to traverse the target link. However, the case studies below show that most of these challenges can be overcome using various heuristics, except those associated with resource limitations.

Experiment methodology. As attack host candidates, we choose all the available PlanetLab [38] hosts located in different organizations. The resulting 88 hosts reside in 64 distinct ASes, connected to 58 distinct upstream providers. These providers belong to diverse levels of Internet hierarchy [46]. We focus on BGP sessions associated with a large tier-1 ISP X (anonymized for privacy concerns) and the Abilene network AS 11537 for our study, as we have access to most of their router configurations used for identifying eBGP peering links. We leave for future work a more complete study on how to identify links involved in BGP sessions, but we show later accuracy of simple heuristics in identifying eBGP sessions validated using router configurations. BGP AS paths from RouteViews [5] are used for identifying potential host and destination prefix pairs that traverse the target link. We traceroute to ".1" IP address in each candidate destination prefix to collect IP level paths from associated PlanetLab hosts. For our study, we probed 6157 destinations altogether. For the paths that traverse the

Category	Target links	AS-level filtered	IP-level filtered	Attack hosts min,max,avg	Dest. prefixes min,max,avg	Bottleneck link (Avg BW Mbps)	AS relations (%) (enter | exit) (peer,customer,provider)	Guessed BGP session(%)
Two tier-1 ISPs	100	11	5	1,25,9	1,1853,234	1 (65.8)	(29,71,0 | 8,92,0)	85
Two small ISPs	60	6	7	1,57,32	1,553,179	2 (63.6)	(15,63,22 | 28,63,9)	77
Multi-connected customer (ISP X)	250	33	34	1,46,13	1,225,98	9 (42.7)	(16,84,0 | 2,98,0)	50
Multi-connected customer (Abilene)	40	1	5	22,64,47	1,466,114	4 (33.7)	(23,57,20 | 9,83,8)	68
Single-connected customer	40	1	13	88	1,28,9	3 (27.3)	(48,52,0 | 0,100,0)	35

Table 2. Summary results for selecting attack hosts and destinations.

target link, we use Pathneck [24] to measure and locate the bottleneck link.

Attack host and destination selection. We define a *multi-connected customer* to be one with more than one physical connection to one or more providers. To help explain results, we categorize target links into five types of eBGP peering links, between the following two entities: 1. Two tier-1 ISPs; 2. Two small ISPs; 3. A multi-connected customer and its tier-1 ISP provider; 4. A multi-connected customer and its small ISP provider; 5. A single-connected customer and its provider. Here we summarize our main findings:

- For links associated with big ISPs, there are fewer attack hosts to choose from compared to small ISPs.
- Except for single-connected customers, there are many destination prefixes to be used by attackers for most links, increasing attack detection difficulties.
- For big ISPs, most paths enter from and exit to customer networks. For small ISPs, paths are more spread out across customer, peer and provider links, but mostly exit to customer networks.

Table 2 shows the results supporting the above findings. A set of links are randomly selected for each link type, shown in the "Target links" column, based on router configurations. We first identify prefixes at the AS path level as described before. For some links, we are unable to identify any destination prefix shown in the "AS-level filtered" column. The "IP-level filtered" column shows the number of links we are unable to find any path traversing via traceroute. Limited vantage points, incomplete traceroute, and IP aliasing account for their existence. We use router configurations to more accurately deal with IP aliasing for routers with a single BGP peering session with another AS. Identifying that the IP level path goes through such routers, we can determine if the target link is traversed.

The attack hosts and destinations for each link type are shown in the columns labeled as "Attack hosts" and "Dest. prefixes." The "Bottleneck link" column shows it is difficult to overload the target link via a single link. The AS relationship between X or Abilene and the network which a particular path enters from or exits to are shown in the second to last column. Finally, we observe high accuracies of determining peering links based on the two adjacent IP hops in the traceroute path with different AS numbers using BGP origin AS to IP mapping for most target links except for single-connected customers most likely due to numbering customer network equipment using provider addresses [33].

We now elaborate on our findings for the target links between two tier-1 ISPs and two small ISPs. Even with the limited vantage points of 88 PlanetLab hosts, we are able to find some IP level paths to traverse most target links. The results indicate that it is more challenging to find paths traversing target links associated with the ISP X compared to Abilene. We found 85% of target links between two tier-1 ISPs can only be traversed from no more than 10 attack hosts and reaching fewer than 300 destination prefixes. This can be explained by the multiple connections between large ISPs and the hot potato or early-exit routing policy, making it difficult to find hosts going through the specific target peering link. Furthermore, we only discovered one target link between tier-1 ISPs to be the bottleneck link on one of the paths.

Compared to peering links between large ISPs, we found after filtering, among the 47 target links between Abilene and its 25 peers, more candidate attack hosts can be used: 51% of these target links can be reached from more than 20 hosts. Compared to large ISPs, smaller ISPs naturally have less rich network connectivity, thus fewer choices to route traffic between them. On average, there are slightly more destination prefixes to select from for links between large ISPs, explained by a larger number of customer prefixes, also supported by the second last column.

The next two types are links between a multi-connected customer and its provider. Most are multihomed customers to different providers. Compared to the first two types of peering links, we identify more attack hosts: 40% of target links in AS X can be traversed from more than 35 hosts, while 40% in Abilene can be reached from more than 60 hosts. This is due to fewer choices to reach a customer network compared to a peer ISP network. Interestingly,

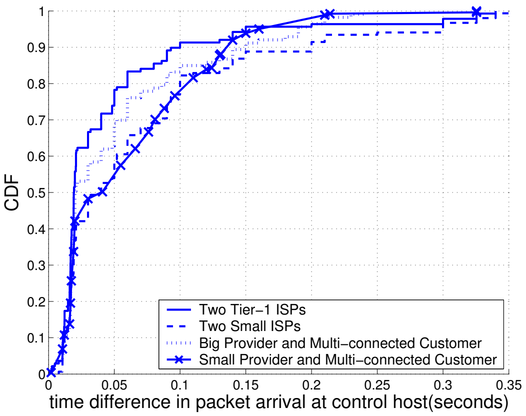


Figure 6. Attack time synchronization granularity (100 runs).

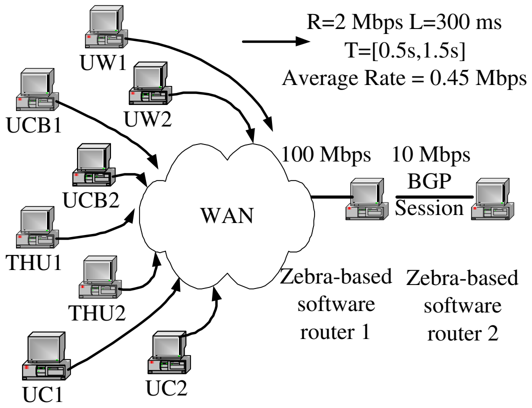


Figure 7. WAN testbed to reset a local BGP session.

there are slightly more destination prefixes to select from for multi-connected customers of Abilene than those of ISP X due to richer network connectivity to X. In summary, it is harder to find network paths to traverse links associated with larger networks with more diverse network connectivity. This applies to selecting both attack hosts as well as destination prefixes.

For the last type of link, the single-connected customer-provider link, we sample 2 links from Abilene and 38 links from AS X. This attack is analogous to DDoS attacks against a single customer network and is more easily detected. Except for the filtered 14 links, the remaining links can be traversed from all the attack hosts, as by definition. Mostly there exist only one or two prefixes as destinations.

Note that the distribution of potential attack paths shown in Table 2 is partly biased by the PlanetLab probe locations, especially for entry points. It is also important to note that attack traffic enters from and exits to a diverse types of links.

Time synchronization. We select a subset of peering links for which we have access to a controller host needed for calibration, from each category listed in Table 2 to evaluate the accuracy of the synchronization algorithm presented in Section 5.1.2. In particular, we select 2, 3, 5, and 13 links respectively from the first four types. The number of attack hosts for a target link varies from 9 to 27. We used 19 distinct controller hosts. Due to the limited location of PlanetLab hosts, we are unable to identify a controller host for the single-connected customer links to evaluate time synchronization granularity. This however does not prevent attackers from successfully attacking such links. As we discuss earlier, attackers could use ICMP timestamp replies from routers or live hosts as the reference time.

Figure 6 shows the distribution of the time difference between the earliest and latest packets arriving at a controller host from corresponding attack hosts for 100 experiments. We observe that attack hosts can be synchronized within 100 ms in more than 80% of the runs, and within 130 ms in more than 90% of the experiments.

Wide-area coordinated attacks against a local BGP session. We have shown above the feasibility of attack host selection and synchronization for a wide range of target links, we now perform an actual attack against a locally constructed BGP session, which we set up between two PCs running Zebra software [2], with one PC serving as the gateway for the other as shown in Figure 7. To increase the attack difficulty, only BGP KeepAlives are exchanged. We use Planetlab hosts as attack hosts.[4] In the following, we mimic how the attacker is able to launch the attack without the information of minRTO and link capacity. Without knowing minRTO, the attacker guesses a value within a certain range for inter-burst period each time. Instead of synchronizing all hosts to start attack at the exact same time, the controller synchronize each host to start within a certain range using burst traffic with a fixed burst period and a random burst-free time period. As a result, the aggregated flow causes congestion on the targeted link. The link capacity can be estimated from the maximum available bandwidth measured by other measurement tools.

The link capacity between the two PCs with the local BGP session is 10 Mbps. Each attack flow has a burst length of 300 ms, an inter-burst period randomly selected between 0.5 second to 1.5 seconds[5]. The peak magnitude is adjusted with different number of attack hosts chosen. We vary the number of hosts from 6 hosts each sending 3 Mbps peak rate to 16 hosts with 1.1 Mbps. In order to reset the session,

[4] Given this artificial setup, any host can be used, as each host sending traffic destined to *router 2* traverses the target link.

[5] The range is based on the guess of possible minRTO since minRTO should not be too big for performance concerns nor too small for unnecessary re-transmission.

the peak rate required decreases roughly linearly with the number of hosts. Figure 7 illustrates an experiment of using 8 attack hosts from various geographic locations. In this case, the peak magnitude is 2 Mbps based on our experiments. We launch 50 attacks from these 8 hosts with the average rate of 0.47 Mbps from each host. Every attempt led to a BGP session reset. The required time to bring down the BGP session varies from 4.2 minutes to 43.3 minutes with an average of 18.9 minutes and standard deviation of 95.

The above experiments demonstrate the feasibility to remotely reset a BGP session possibly in the core Internet from multiple end hosts. With coordinated attacks, the burst durations can overlap among different flows so that each flow can have a shorter burst length and a longer inter-burst period. The peak magnitude for a single flow can be small as long as there are enough hosts sending traffic to fill up the available bandwidth at the target link. We next discuss how this and other low-rate attacks against BGP can be prevented.

6 Defense Mechanisms

No known detection techniques exist today that can accurately identify *coordinated low-rate attacks* given arbitrarily low-rate individual attack flows. We focus on prevention mechanisms so that low-rate TCP-targeted DoS attacks cannot impact BGP sessions. To achieve this we enumerate the necessary conditions for such attacks on BGP to be successful. (1) Ability to infer the minRTO value and send low-rate traffic with minRTO as the inter-burst period. (2) Ability to identify the location of the BGP session and send traffic traversing the target link involved in the BGP session. (3) Ability to congest the target link. We describe two general approaches: hiding the necessary information needed for the attack and protecting BGP packets from other traffic. These prevention solutions are themselves not novel, and fortunately some of them can be readily deployed today. We encourage ISPs to immediately adopt them as default configurations and best common practices.

6.1 Hiding Information

To successfully reset a BGP session, the attacker must know the minRTO value for the TCP stack on the target router. As shown earlier in Section 4, such information can be obtained by experimentally studying commercial routers. Different vendors and router types can have dissimilar minRTOs. Fingerprinting techniques such as nmap [35] or simply trial and error may be used for inference.

There are two ways related to minRTO to thwart the low-rate attack. First, if the minRTO, determining the attack inter-burst period, is small relative to the minimum burst length or smaller than the attack synchronization granularity, it would be impossible to launch low-rate attacks. The burst length needs to be sufficiently long to induce packet drops, and is usually several hundred milliseconds. However, small minRTOs might result in unnecessary retransmissions especially for multi-hop BGP session. Another way to mitigate the attack is to randomize the minRTO value as suggested in [27]. The minRTO value is specified by a range $[a, b]$. A random value within the range is assigned as the minRTO for each flow. Randomization reduces the likelihood that attack flows will hit all consecutive retransmitted packets required to reset the session. However, it does not eliminate the impact on BGP throughput degradation.

Related to hiding minRTO values, another way to prevent low-rate attacks is to conceal network topologies from end-hosts, so that it becomes impossible for attackers to identify target links of BGP sessions. In fact, many routers in edge networks already disable ICMP TTL Time Exceeded replies using firewalls, which are needed for traceroute to discover topologies. The deployment of MPLS also makes discovering internal ISP topologies difficult. The disadvantage is that legitimate use to discover topology is also denied. Related to disabling ICMP TTL Time Exceeded replies, disallowing ICMP timestamp replies would make coordinated attacks more difficult to synchronize.

6.2 Prioritizing Routing Traffic

A direct approach to defeat low-rate DoS attacks against BGP sessions is to provide bandwidth guarantee and prioritized scheduling for BGP traffic so that it is not affected by congestion caused by other traffic. This approach protects routing traffic from both intentional attacks as well as unintended traffic surges. Thus, this is the recommended approach. More importantly, this can be achieved today using existing router support. From a high level, there are two essential components: (1) *Prioritization*: each BGP router and any other router that may forward BGP traffic needs to prioritize BGP traffic at both input and output queues. (2) *Marking*: the edge router must ensure that non-BGP traffic is not marked with the high priority needed to differentiate BGP traffic. If differentiation is based on router source IP addresses, source spoofing can be prevented using either ingress filtering or the Generalized TTL Security Mechanism (GTSM) [23].[6] TTL and priority checking can also be combined to help prevent spoofing of priority markings. Next we empirically examine several widely implemented features for traffic differentiation on today's commercial routers in their effectiveness at protecting BGP traffic against low-rate attacks: Random Early Detection

[6] This works by dropping packets with smaller than expected TTLs based on the number of hops between the two routers in a routing session.

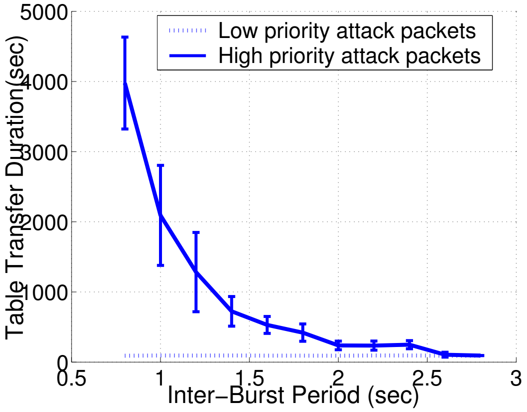


Figure 8. Impact of inter-burst period on BGP table transfer with WRED enabled (R=185 Mbps, L=200 ms)

(RED) [22, 15, 31], Committed Access Rate (CAR) [19], and class-based policing [20]. These mechanisms either provide prioritized buffer access or link scheduling.

Random Early Detection (RED): RED [22] has been widely implemented in routers to help maintain small queue sizes and prevent TCP synchronization. Weighted RED (WRED) [15] allows traffic differentiation based on IP precedence by setting different RED parameters for each traffic class. In our experiments, we take Cisco recommendations [15] to give lower priority to packets marked with IP precedence of 0 (default) and higher priority to those marked with precedence of 6. Newly arriving high priority packets are not discarded until all incoming low priority packets are dropped. By default, routers we studied do not enable WRED and mark locally generated BGP packets with a precedence value of 6. WRED is applied only at output queues, and is thus unable to protect routed BGP traffic for iBGP sessions.

We use the same experimental setup as in Figure 2 in Section 4 except WRED is enabled for router R1's output queue. Under low-rate attacks configured with R=185 Mbps and L=200 ms, Figure 8 compares the table transfer duration when attack traffic uses IP precedence value of 0 (low priority) with the scenario when it uses IP precedence value of 6 (high priority). In the former case, BGP table transfer is not impacted. In the latter case, WRED cannot protect BGP traffic resulting in similar performance as in Figure 3(b) without WRED. This illustrates the importance of policing the IP precedence marking on packets, so that attack packets are treated with lower priority. We also found that WRED can prevent session reset for low-priority attack traffic.

Class-based queueing and traffic marking: Today's routers generally support packet marking and class-based queueing using several criteria. For example, Committed Access Rate (CAR) [19] supported by Cisco routers we studied limits both the input and output transmission rates on an interface based on criteria such as incoming interface, IP precedence, QoS group, or IP access list, and also classifies packets by setting the IP precedence or QoS groups.

In our experiments, we configure CAR on the input interface to reset incoming attack packets to have IP precedence of 0, preventing attack packets from spoofing higher precedence values. We also configure CAR on the output interface to drop the packets with IP precedence of 0 when its burst rate exceeds 100 Mbps. We found that CAR is very effective in isolating BGP packets from attack traffic. The performance is similar to the curve marked with "low-priority attack packets" in Figure 8. Class-based policing [20] is a similar mechanism which we experimented on Cisco routers with the same effectiveness.

In summary, prevention mechanisms described here can be readily configured in today's routers. Complementary router-supported features, such as Graceful Restart [42] and those proposed by the research community FRTR [49] can help reduce the overhead due to session resets. The existing focus of the network community [16, 25, 41] has been on practices such as preventing unauthorized router access by setting up access control lists and preventing router CPU overload by rate-limiting ICMP replies. However, this is not sufficient in protecting routers from remotely launched resource-based attacks such as low-rate attacks described here. We provide here suggestions to protect routing traffic from general DoS attacks including low-rate attacks beyond existing proposals. Operators need to configure routers to provide class-based queueing or prioritized buffer access for BGP traffic marked with higher priority. Edge routers must set up necessary filters to prevent attack packets from spoofing higher priority. Finally, hiding the network topology and infrastructure IP addresses also help protect the network. We recommend that router vendors enable such protection for routing traffic as the default configuration.

Alternatively, a better transport protocol for BGP can help prevent BGP from low-rate attacks. BGP only require reliable transfer for the latest update message for each individual prefix as well as in-order delivery for messages belonging to the same prefix. Furthermore, transport for BGP should be more aggressive than TCP flows of regular data traffic. Such transport modification will prioritize BGP traffic whose flow will less likely back off during congestion.

7 Conclusion

Attacks against the Internet infrastructures such as routers can have devastating global impact on network stability and robustness. A fundamental weakness in today's Internet is that the control plane or routing packets by default is not protected from other traffic. Thus, data congestion due to either intentional attacks or unintended traffic bursts can adversely impact routing sessions. In this work, we examine the impact of low-rate TCP-targeted DoS attacks on BGP. Such attacks can be launched remotely without access to routers. Using detailed experimentation, we show that routers using default configurations are vulnerable to such low-rate attacks. The attacked BGP session can suffer severe impact in the form of session reset and increased convergence delays, resulting in global network instability, unreachable destinations, and data plane performance degradation. Moreover, we illustrate that coordinated low-rate attacks are feasible from multiple end-hosts, further raising the detection difficulty given even lower individual attack flow rate. As defense mechanisms, we advocate prevention techniques to eliminate the possibility of any DoS attacks which exploit traffic congestion to impact routing protocols. Using testbed experiments we demonstrates the effectiveness of such solutions to prevent low-rate attacks.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We are particularly grateful to Dan Pei, Jennifer Rexford, and Aman Shaikh for their insightful comments. Suggestions from Jon Andersen, Xu Chen, Evan Cooke, Ranga Vasudevan, and Kevin Borders help significantly improve the presentation of the paper. Our work also benefited from the feedback of Albert Greenberg, Tom scholl, Jean-Franois Le Pennec, Leonard Ciavattone, Xiangqun Liu, Aswatnarayan Raghuram, and Mark Lyn. Our work would not have been possible without the dedication from the support staff of the Wisconsin Schooner router testbed. This work was supported in part by the National Science Foundation under the grant CNS-0430204.

References

[1] clockdiff. `http://www.linuxforum.com/man/clockdiff.8.php`.

[2] GNU Zebra-routing software. `http://www.zebra.org`.

[3] RIS Raw Data. `http://www.ripe.net/projects/ris/rawdata.html`.

[4] Schooner User-Configurable Lab Environment. `http://www.schooner.wail.wisc.edu/index.php3?stayhome=1`.

[5] University of Oregon Route Views Archive Project. `http://www.routeview.org`.

[6] A. Akella, S. Seshan, and A. Shaikh. An Empirical Evaluation of Wide-Area Internet Bottlenecks. In *Proc. Internet Measurement Conference* , 2003.

[7] J. Aldridge and A. Vural. A first look at Saturday's MS-SQL worm as seen by BGP activity recorded by RIS project. RIPE 44 Meeting, January 2003.

[8] P. Almquist. Type of Service in the Internet Protocol Suite. RFC 1349, July 1992.

[9] K. G. Anagnostakis, M. Greenwald, and R. S. Ryger. Measuring Network-internal Delays using only Existing Infrastructure. In *Proc. IEEE INFOCOM*, 2003.

[10] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *Proc. ACM SIGCOMM*, 2004.

[11] G. Appenzeller, N. McKeown, J. Sommers, and P. Barford. Recent Results on Sizing Router Buffers. In *Network Systems Design Conference*, 2004.

[12] S. M. Bellovin and E. R. Gansner. Using Link Cuts to Attack Internet Routing. ATT Research, Tech. Rep., 2004.

[13] H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. 2000.

[14] Y. Chen, Y.-K. Kwok, and K. Hwang. Filtering Shrew DDoS Attacks Using A New Frequency-Domain Approach. In *Proc. IEEE LCN Workshop on Network Security*, 2005.

[15] Cisco Systems. Weighted Random Early Detection (WRED), 1998.

[16] Cisco Systems. NSA/SNAC Router Security Configuration Guide, 2001.

[17] Cisco Systems. BGP Nonstop Forwarding (NSF) Awareness, 2005.

[18] Cisco Systems. Cisco Security Advisory: TCP Vulnerabilities in Multiple IOS-Based Cisco Products, 2005.

[19] Cisco Systems. Configuring Committed Access Rate, 2005.

[20] Cisco Systems. Class-Based Policing, 2006.

[21] J. Cowie, A. Ogielski, and B. Premore. Internet Worms and Global Routing Instabilities. In *Proc. SPIE*, 2002.

[22] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1993.

[23] V. Gill, J. Heasley, and D. Meyer. The Generalized TTL Security Mechanism (GTSM). RFC 3682, February 2004.

[24] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang. A Measurement Study of Internet Bottlenecks. In *Proc. IEEE INFOCOM*, 2005.

[25] Juniper Networks. Best common practices for hardening the infrastructure, 2002.

[26] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE J. Selected Areas in Communications*, 2000.

[27] A. Kuzmanovic and E. W. Knightly. Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants). In *Proc. ACM SIGCOMM*, 2003.

[28] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental Study of Internet Stability and Wide-Area Network Failures. In *Proc. International Symposium on Fault-Tolerant Computing*, June 1999.

[29] C. Labovitz, R. Malan, and F. Jahanian. Internet routing stability. *IEEE/ACM Trans. Networking*, 6(5):515–528, October 1998.

[30] X. Luo and R. K. C. Chang. On a New Class of Pulsing Denial-of-Service Attacks and the Defense. In *Proc. Network and Distributed System Security Symposium*, 2005.

[31] R. Mahajan, S. Floyd, and D. Wetherall. Controlling High-Bandwidth Flows at the Congested Router. In *Proc. International Conference on Network Protocols*, 2001.

[32] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet Path Diagnosis. In *Proc. ACM Symposium on Operating Systems Principles*, 2003.

[33] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an Accurate AS-level Traceroute Tool. In *Proc. ACM SIGCOMM*, 2003.

[34] J. Ng. Extensions to BGP to Support Secure Origin BGP (soBGP). IETF Draft: draft-ng-sobgp-bgp-extensions-01.txt, November 2002.

[35] Nmap–Network Mapper. `http://www.insecure.org/nmap/`.

[36] J. Padhye and S. Floyd. Identifying the TCP Behavior of Web Servers. In *Proc. ACM SIGCOMM*, 2001.

[37] V. N. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proc. ACM SIGCOMM*, 2001.

[38] PlanetLab. `http://www.planet-lab.org`.

[39] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, March 1995.

[40] R. Richmond. Firms Join Forces Against Hackers. *Wall Street Journal*, March 28, 2005.

[41] Ryan McDowell. Implications of Securing Backbone Router Infrastructure. NANOG Meeting, May 2004.

[42] S. R. Sangli, Y. Rekhter, R. Fernando, J. G. Scudder, and E. Chen. Graceful Restart Mechanism for BGP. IETF Internet Draft, June 2004.

[43] A. Shaikh, L. Kalampoukas, R. Dube, and A. Varma. Routing Stability in Congested Networks: Experimentation and Analysis. In *Proc. ACM SIGCOMM*, 2000.

[44] A. Shevtekar, K. Anantharam, and N. Ansari. Low Rate TCP Denial-of-Service Attack Detection at Edge Routers. *IEEE Communications Letters*, April 2005.

[45] N. Spring, R. Mahajan, and D. Wetherall. ISP Topologies with Rocketfuel. In *Proc. ACM SIGCOMM*, 2002.

[46] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proc. IEEE INFOCOM*, 2002.

[47] H. Sun, J. C. Lui, and D. K. Yau. Defending Against Low-rate TCP Attacks: Dynamic Detection and Protection. In *Proc. International Conference on Network Protocols*, 2004.

[48] F. Wang, L. Gao, J. Wang, and J. Qiu. On Understanding of Transient Interdomain Routing Failures. In *Proc. International Conference on Network Protocols*, 2006.

[49] L. Wang, D. Massey, K. Patel, and L. Zhang. FRTR: A Scalable Mechanism for Global Routing Table Consistency. In *Proc. International Conference on Dependable Systems and Networks* , 2004.

[50] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and Analysis of BGP Behavior under Stress. In *Proc. ACM SIGCOMM Internet Measurement Workshop*, 2002.