Poster: Network Resiliency via Reactive Routing

Jared M. Smith
University of Tennessee, Knoxville
jms@vols.utk.edu

Max Schuchard
University of Tennessee, Knoxville
mschucha@utk.edu

Distributed Denial of Service (DDoS) attacks leveraging massive numbers of bots are becoming increasingly prevalent. In the past year DDoS attackers targeted major online gaming brands like Sony and Xbox, as well as core Internet services like DynDNS. Prior work has developed measures that attempt to protect against DDoS attacks by means of load-balancing or black-holing packets on the border of the victim. However, these defenses do not impact attack traffic until it reaches the victim Autonomous System (AS). Thus these defenses often fail to resist recent attacks due to links upstream of the victim AS becoming too saturated with traffic.

Our system allows victim networks to mitigate DDoS attacks by actively manipulating the routes utilized by *incoming* traffic in an effort to route traffic away from attacked links. We build a technique that allows the victim AS to originate routes with three special properties for their networks. First, the victim can exclude arbitrary edges in the AS topology from being utilized by the route. Second, the victim can ensure that if an AS has at least one of these special routes, it will utilize that route as the best path for traffic bound to the victim AS. Lastly, the victim can blacklist arbitrary ASes, preventing them from receiving these routes.

Upon detection of upstream links experiencing a DDoS attack, the victim AS identifies which ASes are the primary contributors of DDoS attack traffic and what critical upstream ASes are impacted by the congestion. The victim AS originates our special routes, ensuring that the routes do not utilize the congested link. The victim attempts to ensure the propagation of these routes to critical upstream ASes, while at the same time masking them from bot heavy ASes. The result is that critical ASes migrate their traffic to unsaturated links, while the majority of attack traffic remains on its original path. Our results show that we can move traffic off and onto arbitrary links with 78% success for links adjacent to the victim AS and greater than 80% success for links one or more hops distant from the victim, while still preventing a 79% of attack traffic from utilizing the alternative path. We also explore the amount of side-effects we impose on neighboring AS's paths as a result of our system. We find no instances of increased transit costs for affected ASes and only modest increases in path length.

Background. *Volumetric Distributed Denial of Service attacks* degrade the availability of a host by saturating links the host utilizes to send and receive network traffic. DDoS attack traffic is commonly generated by botnets, collections of compromised end hosts scattered across the Internet. Traditional DDoS attacks target the link directly connecting the victim to the Internet. As a result many DDoS defenses focus on protecting this link via mechanisms like filtering [4]. Recently, a new DDoS attack strategy has emerged, targeting core transit links that serve the victim host's *entire network* [2]. Filtering defenses do not mitigate these new attacks since the congestion happens on transit links outside of the victim and its ISP.

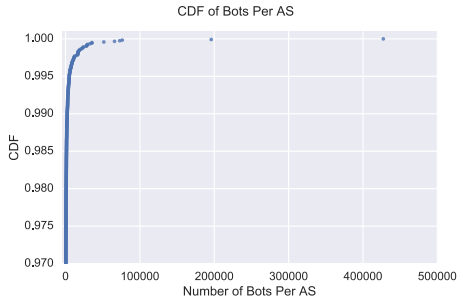


Fig. 1: CDF of the number of bots per AS in our Dataset.

The *Border Gateway Protocol*, or BGP, is the current defacto standard inter-domain routing protocol used on the Internet. BGP is a path vector routing algorithm augmented with policies. Route selection is first conducted based on business relationship with the next hop, followed by the path length. BGP is agnostic to a path's real time congestion, packet loss, round trip time, and other quality of service concerns. An important consequence of BGP's path vector nature is that networks have no *direct* control over what path *inbound* data takes.

Approach. Our goal is to protect the victim AS by moving critical traffic off saturated links while leaving routes used by the majority of DDoS traffic unchanged. We refer to victim ASes utilizing our system as *reactors*. As discussed earlier, BGP does not allow reactors to directly modify incoming routes. Instead, a reactor will take advantage of BGP's implicit behaviors to indirectly influence routes used by the outside world to reach it. BGP allows ASes to advertise sub-blocks of an already advertised block of IP addresses, called *hole punched routes*. These sub-blocks are treated by BGP as different destinations from the aggregate IP block. When a BGP router attempts to forward a packet, it will use the path for the most specific (i.e. smallest) IP block that contains the destination IP address. This means that if a BGP router has at least one hole punched path to a destination, it will always forward traffic along a hole punched path.

The reactor wants the hole-punched routes to propagate such that they do not traverse the attacked link. To achieve this objective, at least one AS connected to the attacked link must reject the hole-punched route. Since the AS rejected the route, they will not propagate it, which in turn means the hole-punched traffic will never traverse the attacked edge. The hole punched paths will propagate *around* these rejecting ASes normally, eventually reaching the critical ASes.

To force the rejection of the hole punched route by an AS, the reactor utilizes BGP's lack of path integrity checks combined with BGP's loop detection mechanism. The reactor falsely adds the AS number of any AS which they wish to force to reject the hole punched route to the path prior to originating the route. After doing so, the reactor then adds their own AS

(a) Success Percentage (b) Average Disturbed ASes (c) CDF of Disturbed ASes (d) Average Path Change

Fig. 2: Success Percentages, Disturbed ASes, and Path Length Change

number to the path and advertises the route. When the hole punched route propagates to one of the blacklisted ASes, their first step will be to scan the path for their own AS number. Since their AS number appears in the path, the blacklisted AS will reject the path because it appears to form a loop. We call this technique for restricting path propagation *Fraudulent Route Reverse Poisoning*, or FRRP. The ASes surrounding the link that we want to avoid appear in the path; however, their presence is irrelevant for actual packet forwarding, because they appear after the destination AS.

FRRP can also be used to force ASes which contain high numbers of bots to ignore the hole punched paths, resulting in attack traffic remaining on the original path. Excessive blacklisting of bot ASes could result in the hole punched routes not propagating to the critical ASes. As shown in Figure 1, the majority of bots on the Internet live within a small number of ASes. Thus the bulk of attack traffic can be left in place by blacklisting only these bot heavy ASes.

Results. To evaluate the effectiveness of our approach, we built upon a BGP simulator used in our prior work [3]. The simulator is essentially a collection of software routers who speak BGP configured in a realistic topology. The topology used in the simulation is from Caida's inferred AS relationships dataset taken from December of 2016 [1]. The BGP policies used by the simulated routes matched the current best practices used by operators. We also have a dataset of of 23 botnet families which were observed launching DDoS attacks between late August 2012 and March 2013 with a total of 2.2 million unique hosts, which we use to measure the affects of bot location on our experiment.

Using our simulator, we can examine both the effectiveness and cost of a reactor using our system to migrate critical traffic off of links suffering under DDoS. Our experiment repeatedly picks two random ASes from the Internet's default free zone, that is ASes that are not stub ASes, fixing one of the ASes as the reactor and the other as an AS generating critical traffic. For each link on the original path between the reactor and critical AS, we simulate the reactor attempting to respond to a DDoS attack impacting that link. In each attack scenario, we determine the following: if we succeed at moving traffic off the attacked links, how many additional ASes see a change in their best path to the reactor AS (termed *disturbance*), the average increase in path length among the disturbed ASes, and if any disturbed ASes switch to using a next-hop that is less preferable economically.

Figure 2a shows the percentage of success in avoiding links along the path between reactor-critical AS pair. As can been seen, success is inversely proportional to the number of bot heavy ASes that need to be blacklisted. When preventing propagation to the 100 largest ASes by bot population we have success rates no less than 78%. Increasing blacklisting to the 1000 largest ASes by bot population results in success rates around 40%. Thankfully, blacklisting only the 100 largest ASes covers 79% of all attack traffic. Figure 2b shows the average number of disturbed ASes as a function of how far away from the reactor the attacked link is. For all distances, on average less than 5000 ASes among a total of over 55,000 ASes in the Internet are disturbed. Large bot AS blacklisting results in fewer disturbed ASes due to poorer propagation of hole punched roues. It should be noted that the current method of avoiding links does not perform any actions to reduce the amount of disturbed ASes outside of bot heavy ASes. No instances of an AS switching to a less economically preferable route, defined by switching from a customer learned route to a peer or provider learned route, were observed as a result of reactor actions. This means our current method incurs no additional monetary costs on ASes outside of the reactor. Figure 2d shows the average path length increase seen by disturbed ASes as a function of distance between the attacked link and reactor. Though the path length change is greater on links closer to the reactor, with about a 2.5 hop increase for links less than 2 hops out from the reactor and decreasing for links further out.

Future Work. Looking ahead, we will be exploring more advanced heuristics to minimize the side-effects of utilizing FRRP to mitigate the effects of DDoS attacks. We will also be developing a realistic link capacity model for our simulation, which will allow us to move traffic off links based on the their actual bandwidth limits.

Acknowledgments. We would like to acknowledge Aziz Mohaisen from SUNY Buffalo for providing the data describing bot locations across the Internet.

References

[1] CAIDA AS relationship dataset. http://www.caida.org/data/active/as-relationships/index.xml.

[2] M. S. Kang, S. B. Lee, and V. D. Gligor. The crossfire attack. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 127–141. IEEE, 2013.

[3] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper. Routing around decoys. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 85–96, New York, NY, USA, 2012. ACM.

[4] A. Yaar, A. Perrig, and D. Song. Siff: A stateless internet flow filter to mitigate ddos flooding attacks. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 130–143. IEEE, 2004.

Nyx: Network Resiliency via Reactive Routing Decisions

Jared M. Smith and Max Schuchard
University of Tennessee, Knoxville

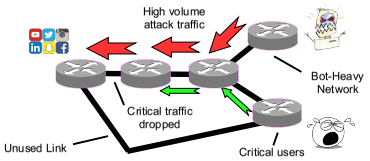
TICKLE COLLEGE OF ENGINEERING
THE UNIVERSITY OF TENNESSEE, KNOXVILLE

Abstract

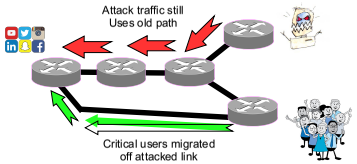
- DDoS attacks are on the rise and botnets have never been larger than they are today, yet current countermeasures can't stand up to the threat
- Links upstream of the victim Autonomous System (AS) become saturated by adversarial traffic and drop legitimate traffic
- We plan to mitigate this issue in two ways:
  - Load balance adversarial traffic across links inbound to the victim AS
  - Move legitimate traffic off saturated links around links impacted by DDoS
- In this work, we show we can accomplish the primitives required to do this by showing that an AS can manipulate traffic that is coming into it from an arbitrary AS on an arbitrary link

The Effects of DDoS Attacks and the Flexibility of the Routing Topology

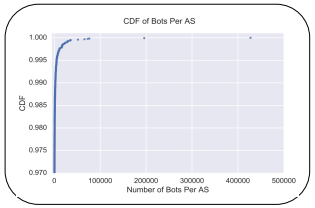
DDoS attacks saturate links and cause critical traffic to be dropped, but BGP is not aware of attacked links and doesn't attempt to preserve Quality of Service



Our system can manipulate routes so that critical traffic stays off attacked links while other links are under stress due to an attack



Bot Distribution in the Internet



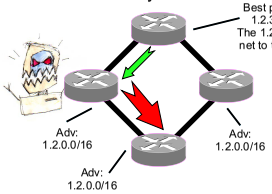
If we can move legitimate, non-bot traffic off the links that are along the best paths for those ASes, then we can halt the affects of DDoS attacks since there are a small number of ASes with a large amounts of bots

Impacting Incoming Traffic

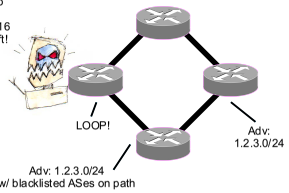
- Path selection decisions only control ***outbound*** traffic, not ***inbound***
- ***Fraudulent Route Reverse Poisoning*** (FRRP) uses BGP hole punching to reroute incoming traffic
  - BGP allows for sub-blocks of existing IP blocks to be advertised
  - Packets are forwarded along the best path to the *most specific* prefix known
  - Victim ASes, which we call reactor ASes, can falsely add all bot ASes to the BGP path of advertised routes as well as ASes along links they want incoming traffic to avoid, which we call moved ASes
  - Moved ASes will ignore these routes because of loop detection, and not propagate them

Fraudulent Route Reverse Poisoning

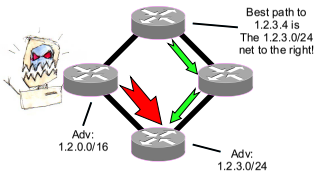
Non-reactors will send traffic bound to reactors across attacked links and links around bot heavy ASes



Reactors hole-punch routes adding blacklisted ASes, loop detection prevents route propagation



BGP will forward using the hole-punched route if available, which will not traverse over attacked links

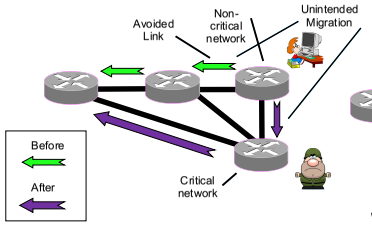


Percentage Success

Reverse Poisoning Side-Effects

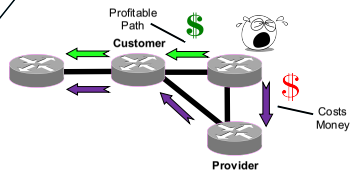
- By manipulating paths outside of the reactor AS, we cause several side effects to varying degrees:
- Disturbing other ASes best paths
  - **Problem**: We don't want to saturate additional links on critical paths to the reactor AS
- Changing the BGP Local Preference for routes
  - **Problem**: We don't want to incur additional monetary costs on ASes
  - Note: we don't observe this in our results
- Increasing the path length from reactor AS to other ASes
  - **Problem**: We don't want to lower performance of the network

We sometimes change the best paths to the reactor of ASes in the topology, which will be mitigated in future work

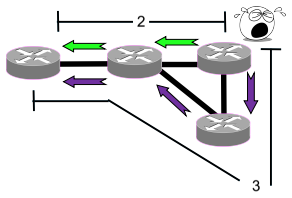


Side-Effects

We could cause inter-AS business relationships to change for routes, incurring financial costs on affected ASes



We sometimes cause path length increases from critical ASes to the victim AS, though modest at the most



We did not see any monetary cost increases!

Side-Effects of Path Manipulation