The Maestro Attack: Orchestrating Malicious Flows with BGP

Tyler McDaniel(✉), Jared M. Smith, and Max Schuchard

University of Tennessee, Knoxville, USA
{bmcdan16,jms,mschucha}@utk.edu

Abstract. We present **Maestro**, a novel Distributed Denial of Service (DDoS) attack that leverages control plane traffic engineering techniques to concentrate botnet flows on transit links. Executed from a compromised or malicious Autonomous System (AS), Maestro advertises routes poisoned for selected ASes to collapse inbound traffic paths onto a single target link. A greedy heuristic fed by bot traceroute data iteratively builds the set of ASes to poison. Given a compromised router with advantageous positioning in the AS-level Internet topology, an adversary can expect to bring an additional 30% of the entire botnet against vulnerable links. Interestingly, the size of the adversary-controlled AS plays little role in this amplification effect; core links can be degraded by small, resource-limited ASes. To understand the scope of the attack, we evaluate widespread Internet link vulnerability via simulation across several metrics, including BGP betweenness and botnet flow density, and assess the topological requirements for successful attacks. We supplement simulation results with ethically conducted "attacks" on real Internet links. Finally, we present effective defenses for network operators seeking to mitigate this attack.

Keywords: DDoS · Link Flooding Attack · Interdomain routing

1 Introduction

Distributed denial of service (DDoS) attacks direct traffic from many distinct sources on the Internet to overwhelm the capacity of links or end hosts. These attacks proliferate despite extensive academic and economic investment in mitigation, and intensify with the Internet's expansion to new devices and services. A reflection attack fueled by unprotected memcached servers, for example, temporarily disabled Github [35]. *Link Flooding Attacks* or *LFAs* are DDoS attacks against infrastructure links rather than end hosts. Prominent examples in the literature include the Coremelt [52], Crossfire [26], and CXPST [46] attacks. Worryingly, LFAs could be moving from proposed attacks to present threat. In 2016, a Mirai botnet-sourced attack directed over 500 Gbps of traffic to a Liberian infrastructure provider in a real-world LFA [47].

Study supported by the National Science Foundation under Grant No. 1850379.

N. Park et al. (Eds.): SecureComm 2020, LNICST 335, pp. 97–117, 2020.
https://doi.org/10.1007/978-3-030-63086-7_7

In order for an LFA adversary to successfully target a link, the adversary must be able to direct traffic from bots to destinations such that the traffic traverses the target link. However, while end hosts control their traffic's destination, they cannot control the path taken by that traffic. Network operators for the Autonomous System (AS) that bots reside within chose outbound paths for their flows. This means that links have varied exposure to LFAs due to routing choices. For some targets, an attacker may find destinations for individual bots such that the entire botnet sends traffic over some target link; for others, bots may have no such paths.

Prior academic work fails to deeply address this topological constraint. For example, Coremelt [52] only quantified if an adversary could flood *at least one* link of the ten largest ASes by degree, but did not quantify how many links were impacted for those ASes. CXPST [46] targeted links based on control plane properties, and recognized in their results that many likely target links are out of the attacker's reach. Most recently, Crossfire [26] explored the ability of LFAs to disconnect geographic regions. However, their experiments used paths from PlanetLab nodes both as their model for botnet traffic propagation and *overall* topology, meaning that *by definition the only links considered in their model were those their "botnet" could reach.*

In Sect. 3, we quantify the vulnerability of Internet links to LFAs with simulations that join CAIDA's inferred Internet topology with botnet models backed by real-world bot distributions. For one of our link sample sets, we found that just 18% to 23% of links are traversable by the majority of bots in three major botnets, and fewer than 10% of sampled links were vulnerable to $>75\%$ of bots. Even if we restrict our set of targets to the 10% most traversed links, the majority are reachable by $<20\%$ of the botnet.

In Sect. 4, we investigate how routing-capable adversaries can expose potential LFA targets inadvertently shielded by routing choices. The result is a novel attack, Maestro, that orchestrates remote AS path selection and malicious bot flows to degrade links outside the reach of traditional LFAs. Our attack requires an adversary to have two tools: a botnet and limited control of a Border Gateway Protocol (BGP) speaker. The Maestro attack utilizes a traffic engineering technique called *BGP poisoning* to adjust bots' inbound path to the compromised BGP speaker's network so they include a target link. The attacker then launches a traditional DDoS attack against the compromised BGP speaker's network, resulting in attack flows funneled over the target link.

We demonstrate Maestro's ability to both amplify LFAs for already-vulnerable links and extend a botmaster's reach to previously unexposed targets in simulation. After executing the Maestro attack from a well-positioned adversary, more than 90% of sampled links are exposed to a majority of bots across each botnet, and 85% to 87% of links are exposed to 75% or more bots. Our analysis explores a number of different target link/compromised BGP speaker selection methods to discover how these properties factor into attack success. The results of these evaluations are explored in depth in Sect. 6, where we analyze two Maestro adversary types.

We validate Maestro's underlying mechanisms with experimental "attacks" on the live Internet in Sect. 5. From two different "compromised" routers, we achieved 2x or greater flow density improvement against target Internet links. Following our experiments, we consider defenses to mitigate Maestro, exploring the relative effectiveness of each via simulation, with in-depth discussion of results. Our goal is to give a first look into mitigation techniques network operators can individually deploy to protect their own links from the Maestro attack. Feedback from outreach to the network operator community is also presented.

2 Background

2.1 Border Gateway Protocol

The *Border Gateway Protocol* (BGP) [43] is the Internet's de facto routing protocol. BGP enables 68,000 Autonomous Systems (ASes) to exchange routing information and connect disparate parts of the Internet's infrastructure. BGP routes are defined by a destination IP prefix and a collection of attributes, including the AS PATH, or list of AS-level hops to the route destination. ASes originate routes to hosted IP prefixes via BGP advertisements to neighboring ASes. Each AS chooses its path to a prefix based on attributes of stored paths, most notably LOCAL PREF and AS PATH length. LOCAL PREF represents the AS operator's local policy choices regarding path qualities, and holds precedence over AS PATH length in the decision process. The *longest prefix matching* rule dictates that the stored path with the longest (most specific) IP prefix match is used to forward received packets.

Because the BGP decision process draws on path and policy attributes in route selection, BGP is a path-vector algorithm *with policies*. These policies are informed by *business relationships* between ASes. ASes can have peers, customers, and providers. Peers exchange traffic for free, while customers pay providers to transit traffic. These economic partnerships shape the Internet topology according to the valley-free routing model [18]. In simple terms, the model states that BGP routes will not transit from a customer to a provider after transiting from a provider to a customer, which ensures ASes do not incur monetary costs whenever possible. One important abstraction related to this model is the *customer cone*, defined as an AS and all of its direct/indirect customers. More formally, an AS's customer cone is the set of all ASes that can be reached from the AS transiting only provider to customer links [33].

2.2 BGP Poisoning

The BGP decision process gives local operators control over outbound paths, but operators have little influence over inbound traffic paths. Techniques do exist for next-hop inbound path control, including the MULTI EXIT DISC (exit discriminator) attribute [37] and BGP communities [16], but both are subject to remote AS's policies. This means inbound path control cannot be exerted by a

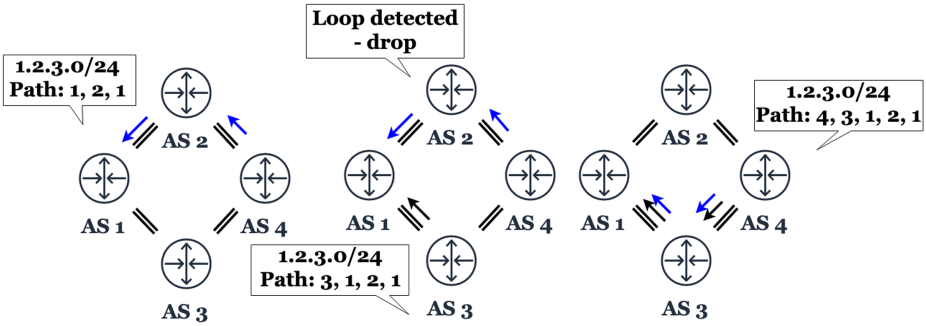


Fig. 1. BGP poisoning. AS 1 advertises a specific prefix (thicker arrow). AS 4's traffic to AS 1 (blue) is moved to the more specific route. AS 2 is said to have been *poisoned*. (Color figure online)

destination AS arbitrarily on the broader Internet. Fortunately, *BGP poisoning*, a traffic engineering technique found in prior work [1,28,48,49,54], allows for inbound route manipulation *without coordination* from other ASes. Recent work demonstrates poisons of non-Tier 1 networks are rarely filtered [48].

BGP poisoning relies on two characteristics of BGP: loop detection and longest-prefix matching. Loop detection is a specified BGP behavior that dictates ASes drop paths that already contain their own AS number (ASN). This prevents loops, but it also enables BGP poisoning (see Fig. 1 for an illustration). An AS that wishes to adjust inbound paths off certain links can falsely include the ASes to be avoided in an advertised path. As a result of being included on the path, the poisoned ASes will drop the incoming update, and routes will propagate around links involving the poisoned ASes. In practice, the poisoned ASes are sandwiched between copies of the originator's ASN to avoid issues with RPKI (see Sect. 2.2 below). Connectivity between poisoner and poisoned ASes can be maintained by using two sets of advertisements: a non-poisoned advertisement for the IP prefix, and poisoned advertisements for more specific (longer) prefixes that cover the original block of IP addresses. Since packets are forwarded on the most specific address match, any AS which has access to the poisoned path will always install it.

RPKI, Path Aggregation, and Poisoning: The Resource Public Key Infrastructure (RPKI) [34] is a feature in partial deployment [6,20] that ties ASNs to their allocated prefixes via Route Origin Authorizations (ROAs). ROAs are designed to prevent prefix hijacking [39] and do not affect BGP poisoning as described above. BGPsec [30] extends BGP with cryptographic hardening of AS PATHs. If fully deployed, BGPsec would prevent BGP poisoning entirely. However, no commercial BGPsec implementations exist [50], and BGPsec is largely ineffective unless widely implemented [21].

AS PATH aggregation as described in [43] allows ASes to aggregate downstream routes for multiple specific prefixes (e.g., many consecutive/24s) into a

single, more general prefix (e.g., a/16) originated by the *aggregator*. Operators can apply an AS SET attribute with ASNs of aggregated routes; this step preserves poisons on aggregated routes. The Potaroo project's 2020 routing table analysis found that nearly 40% of aggregate address spans are covered by more specific routes, suggesting that aggregators often allow more specifics to propagate alongside aggregates [19].

2.3 Link Flooding Attacks

Volumetric *Distributed Denial of Service* (DDoS) describes a coordinated availability attack on a target link or end host with overwhelming traffic from multiple sources. A common flow source for these attacks are botnets: networks of compromised end hosts (bots) under an attacker's control. *Link Flooding Attacks* (LFAs) refer to DDoS against network infrastructure rather than end hosts. One of the first such attacks in the literature is Coremelt [52]. Coremelt specifies that a botmaster 1) map which links are present on bot-to-bot paths, 2) target a specific link used between many bots, and 3) direct bot traffic *between bots* over the link. The resulting n^2 flows (for n bots with paths over the link) exhausts target link capacity. The Crossfire attack [26] strategically targets multiple Internet links to isolate an entire region (military installation, university, geographic region, etc.). Rather than directing traffic to one another, bots map paths to publicly available web services (decoys) that transit target links. Bots then send sustained, low-intensity flows to decoys to execute the attack.

3 Can Botnets Target Any Link?

Link Flooding Attacks (see Sect. 2.3 earlier) hinge on the botmaster's ability to drive traffic over a target link. First, the botmaster must find paths from bots to remote destinations that cross the target link. Next, they must direct flows over these paths to aggregate traffic at the target and exhaust its capacity. If the attacker's flows overwhelm the link, "drowning out" benign traffic, the attack is successful. Bots without a path to some destination that includes the target link **cannot participate in the attack**.

This path requirement is largely unaddressed by prior work on LFAs. Researchers often 1) do not perform their measurements with distribution data from a real botnet, notably Kang et al.'s Crossfire [26], 2) assume botnets can direct significant flows over arbitrary links on the Internet without evidence, as in Tran et al.'s examination of the feasibility of re-routing based LFA defenses [54], and/or 3) target links most accessible to botnet flows, as in Coremelt [52], CXPST [46], and Crossfire. In fact, both target selection and attack success for Crossfire is measured via a degradation ratio that *only* considers links exposed to bots. These limitations raise questions about how well LFAs can target arbitrary links in the Internet topology. Here we explore those questions with Internet-scale BGP simulation.

3.1 Simulation Methodology

Our experiments are performed by extending the Chaos BGP simulator from prior work [45,46,49,54] with new modules. Chaos builds the Internet's BGP topology based on publicly available inferred AS relationship data from CAIDA [56]. In the simulator, ASes perform a simplified BGP decision process for path selection that includes longest prefix matching, path selection based on LOCAL PREF and AS PATH, and route export based on local policy. As true local AS policies are private, this is the most accurate simulation of AS behavior we can devise. We model our simulator's poison mechanics on the live Internet's treatment of BGP poisoning [1,28,48].

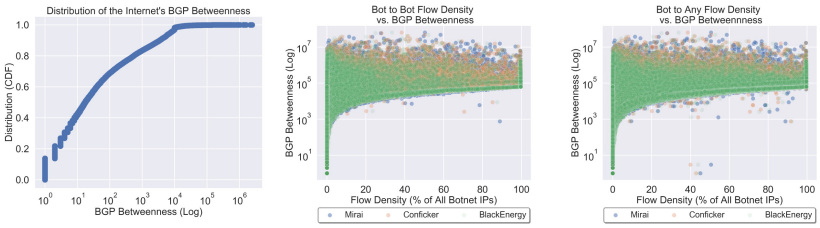
For each attack, we use three botnet models based on Mirai, Conficker, and Blackenergy botnet IP measurements. Our botnet models are built from passive and active measurements of infected hosts from a variety of sources. The *Mirai* botnet model includes 2.29 million IP addresses in 11,633 ASes. These addresses were recorded by a Chinese CDN as they spread the malware, a process with a unique signature [38]. Our *Conficker* model contains 2.28 million bot IPs from 12,095 ASes found by detecting bot rendezvous points and monitoring traffic to these points [53]. The *Blackenergy* model is a SCADA-focused botnet developed from similar techniques as presented in [5] with a total of 310,943 bot IPs across 4,291 ASes. Note these IP counts may not represent true botnet size given DHCP churn and other factors; in this work, we are primarily concerned with AS distribution of bots rather than counting infected hosts.

3.2 Vulnerability Experiments

Our initial experiments in the simulator measure the vulnerability of Internet links to LFAs. We build the AS-level topology as described above and classify links in two ways - relative usage and vulnerability. We quantify link usage with *betweenness*, defined as the number of times a link appears on paths between any pair of ASes in the simulator. Figure 2a shows the cumulative distribution of betweenness for simulated Internet links. Most links appear on 100 or fewer AS paths. Select links, however, have a betweenness of more than *1 million*. Attacks on these critical links would wreak havoc with upstream and downstream networks, and cause widespread disruption in Internet services. We quantify link vulnerability with *flow density*: the percentage of a botnet's infected hosts with simulated paths over the target link to either 1) another bot-hosting AS, called *bot-to-bot* flow density, or 2) any destination AS, called *bot-to-any* flow density. The bot-to-bot flow density models the effectiveness of a Coremelt-style attack; the bot-to-any flow density roughly follows the Crossfire attack.

To show how link vulnerability and usage are related, Fig. 2b plots bot-to-bot flow density on the x-axis against betweenness on the y-axis. Not unexpectedly, some low betweenness (peripheral) links are wholly outside an LFA attacker's reach. We note that relaxing our attack technique by allowing bots to send traffic to any AS destination does not significantly alleviate these limitations, as shown in Fig. 2c. **Critically, some moderate to high betweenness (core)**

links are also partially or completely devoid of paths between bots. This experiment provides evidence that botnet-sourced LFAs can be limited by topological factors. While low betweenness links are most difficult to reach from the botnet families in our study, even highly trafficked links are not always fully exposed to botnet flows.



(a) AS-level edge betweenness of Internet links

(b) Flow density (bot-to-bot) by betweenness

(c) Flow density (bot-to-any) by betweenness

Fig. 2. LFA vulnerability as function of betweenness

4 The Maestro Attack

The prior section demonstrates that most links, including many likely LFA targets, are shielded from the full force of major botnets. This condition arises from the lack of end host control over traffic routes; bots cannot always find a destination for their flows that cross a target link. In this section, we introduce the Maestro attack, the first combination of traffic engineering techniques with LFAs. Maestro alters the control plane to increase target flow density and extend a botmaster's reach to previously shielded links.

Threat Model: To execute the attack, an adversary requires 1) command of a botnet and 2) control of a BGP speaker, i.e., an AS's edge router. The first item is trivially obtainable, as botmasters routinely monetize their networks by renting them out in an attack-as-a-service model on the dark web [41]. **Recent events demonstrate that multiple feasible avenues exist for adversaries to gain routing capability**. The 3ve fraud operation [22] demonstrated the most straightforward route - simply registering a new AS. Insider attacks from disgruntled network operators, e.g. the Canadian bitcoin hijack [32], are another path to adversarial routing capability. Note that edge routers *themselves* need not be compromised to launch the attack - it is sufficient to compromise router *configuration* systems, as may have occurred in the XLHost ISP breach [17]. However, recent Cisco router vulnerability disclosures show that remote attacks on edge routers are also a realistic attack vector [7–10]. Finally, BGP has previously been weaponized for intelligence gathering [15] and censorship [14] by nation states.

Maestro Concept: Since bots are located in disparate ASes, an adversary who seeks to control bot traffic paths must adjust the best path at each bot-containing remote network. Obviously, if the adversary compromised all these ASes, they could change paths directly, but such an adversary is far more powerful than the one in our threat model. Maestro's central insight is that an adversary who controls one edge router in a single AS can influence remote networks' paths *to that AS*. If an adversary first directs bot traffic to an AS/prefix they control (the *compromised AS* or *adversary AS*), the adversary can orchestrate those flows onto a target link with poisoned BGP advertisements (like a conductor, or *maestro*). We call the origin endpoint of the target link the *From AS* and destination endpoint of the target link the *To AS*. In effect, Maestro also executes a traditional DDoS against the adversary AS. This is of little concern to an adversary who compromises another entity's AS. Figure 3 shows the attack in abstract, with link $5 \mapsto 8$ as the target. Before the attack, traffic from bot-hosting ASes (ASes 0–3) to the adversary (AS 10) flows around (and not over) the target link. Our adversary compromises AS 10, and issues specific prefix advertisements with ASes 4 and 6 poisoned. This causes inbound flows from the bot-hosting ASes to the adversary to concentrate over the target link. After altering these paths, the adversary AS (AS 10) directs bot traffic to itself. The result is a channeled DDoS flowing over $5 \mapsto 8$.

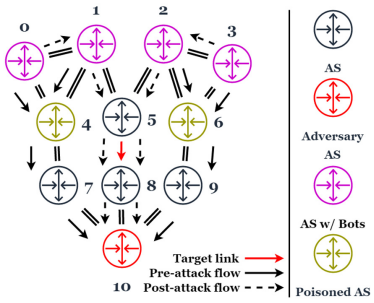


Fig. 3. Maestro: BGP poisons collapse botnet traffic onto a target link.

4.1 Poison Selection Algorithm

BGP poisoning is a primitive that moves inbound traffic *off* a chosen link as seen in prior related work [28,49]. Our adversary's core challenge is different - they want to find a set of poisons (the *poison set*) that focuses traffic *onto* a target link. Finding a poison set that successfully steers bot traffic is non-trivial, because poisons can conflict - successfully steering one AS can block others from traversing the target link. Additionally, each poison extends the poisoning advertisement's AS PATH, and excessively long paths are often filtered [54]. The adversary uses traceroutes [57] to determine bot paths as in [26,52].

The following **iterative poison choice heuristic** represents the core of the Maestro attack. This algorithm finds a poison set for some adversary/target pairing, after which the adversary needs only to issue the poisoned advertisement and direct bot traffic to the poisoned prefix to complete the attack. Our algorithm works by iteratively partitioning ASes into four sets:

Algorithm: Poison Choice Heuristic

function ChoosePoisons $(From, To, Adv, Sources, n)$
Input : From AS $From$, To AS To, adversary AS Adv, source ASes $Sources$, poison limit n
Output: poison set $Poisons$
$Poisons = \emptyset$
while $Sources \neq \emptyset$ *and* $|Poisons| < n$ **do**
  $B = \{b \mid b$ is a bgp path $To \mapsto Adv\}$
  $Sacred = \{From, To, Adv\} + \bigcap_{i=1}^{|B|} B_i$
  $Success \mathrel{+}= \{s \mid s \in Sources$ and $\{From, To\} \in s \mapsto Adv\}$
  $Disconn \mathrel{+}= \{s \mid s \in Sources$ and no specific-prefix path $s \mapsto Adv\}$)
  $Sources \mathrel{-}= \{Sacred \cup Success \cup Disconn\}$
  $Score = [0] * |Sources|$
  foreach $s_i \in Sources$ **do**
    foreach $s_j \in s_i \mapsto Adv$ **do**
      $Score_j \mathrel{+}= 1$
    end
  end
  $Poisons \mathrel{+}= \{max(Score)\}$
  Adv sends advertisement to poison $Poisons$
end

Sacred ASes are those the adversary cannot poison, as doing so would disconnect the target link from the adversary AS. It is initialized with the From AS, the To AS, and the adversary AS. It is updated at each iteration with every AS that appears on all paths from the To AS to the adversary AS, as determined by traceroutes from bots to the adversary AS. Naturally, we must have a path for traffic from the target link to the poisoning prefix, so these ASes should never be poisoned. *Disconnected* ASes include poisoned ASes and those without a route to the advertising prefix that does not transit a poisoned AS. *Successful* ASes are those hosting bots whose traceroutes transit the target link to the adversary. Lastly, *Source* ASes are those hosting bots that are not yet assigned to another set.

After these sets are updated, we select an AS to poison from the source ASes and add it to the poison set. To accomplish this, we select the AS with the highest vertex betweenness on the directed graph formed by traceroutes from remaining source ASes to the adversary. This is the poison that invalidates the maximum number of source paths that avoid the target link. While no guarantee exists that source ASes will then select a path that *contains* the target link, we at least

remove a common hop that *avoids* the link. We update ASes' set membership based on their current traceroutes to the adversary after receiving the update poisoned for all poison set members. Finally, we move to the next iteration. We will terminate iteration once the source set is empty, or if the poison set (which is included in the AS PATH as described in Sect. 2.2) causes the AS PATH to exceed the size AS operators will almost certainly filter in practice: around 254 hops [23,54]. We show in Sect. 7 that this condition is rarely met.

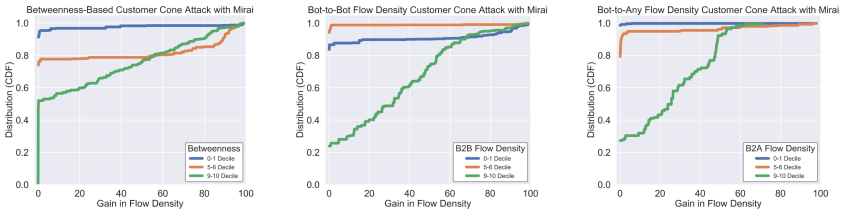
4.2 Evaluation

To evaluate Maestro's impact, we choose thousands of target link/adversary AS pairings for simulated botnet attacks in the framework described in Sect. 3.1. We aim to understand link vulnerability characteristics, and to show how target/adversary topological positions affect flow density. For each attack, we measure pre-attack *flow density* for the target, which represents the target's present vulnerability to LFAs. Next, we execute the Maestro attack to concentrate bot traffic. Finally, we measure post-attack flow density to quantify our success in steering bot-hosting ASes onto the target link. For most experiments, we make bot-to-bot (Coremelt-style) flow density measurements; when using bot-to-any based target link sampling, we will instead measure bot-to-any flow density.

Attacking From the Customer Cone: We initially simulated Maestro on 2,000 randomly selected links with adversaries randomly selected from nearby (1–3 AS-level hops) ASes to illuminate our attack's limits. The attack was generally unsuccessful, but two common success conditions emerged. First, successful attackers were almost universally located in the *customer cone* of the target link destination (the To AS). This is because path export rules are most generous for customers, as ASes provide their customers with all known best paths in hopes of transiting their traffic. So, we expect customer ASes will find the maximum number of *attack paths* - valley-free paths from flow sources to adversary AS that cross the target - among all potential attackers. Second, we observe that flow density improvement for successful attacks varied inversely with attacker distance from the target link. This is an intuitive result; distance increases the number of alternate inbound paths that avoid the target link.

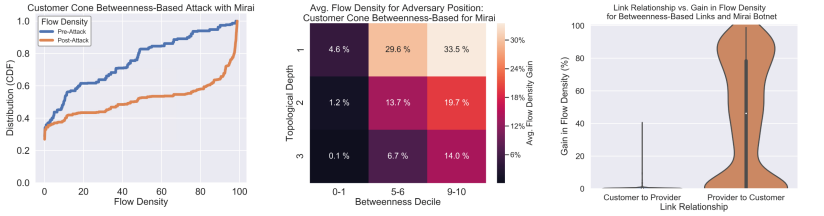
With this knowledge, we next sample 100 links each with relatively low, intermediate, and high betweenness/flow-density. For each target link, we sample three adversaries at each depth from 1–3 in the To AS customer cone. This results in about 1800 adversary/link pairings per link sample set, and about 5400 total simulated attacks. The results are shown in Figs. 5a and 5b. Because we observed similar patterns in success across botnet models, we present only Mirai results here. For direct customers of high betweenness links, on average an additional 30% of the **total bots** in the botnet can target the link (Fig. 5b). For low betweenness links, attack impact is negligible, but these links are not likely targets for LFAs.

A deeper examination of the data yields insights on successful Maestro scenarios. First, target link relationship is critical to attack success. For an adversary



(a) Betweenness-based link selection (b) Bot-to-bot flow density selection (c) Bot-to-any flow density selection

Fig. 4. Flow density gain results (post-attack density - pre-attack density) by link selection strategy, **Mirai** botnet model



(a) Flow density CDF, provider to customer. (b) Success at varying customer cone depths. (c) Success by link relationship.

Fig. 5. Deeper look at the customer attack success, betweenness link sample

in the To AS customer cone, attack paths are most prevalent when the To AS is a customer of the From AS; that is, the target link is a *provider to customer* link. This is an intuitive finding; any bot AS that must transit a provider to customer link to reach the target link *cannot* then transit a customer to provider link and remain valley-free. Like locating the adversary in the To AS customer cone, targeting a provider to customer link removes a potential valley from attack paths, but *at* rather than *after* the target link. The importance of this dynamic is shown in the relative distribution of flow density gains by link relationship in Fig. 5c. Virtually all successful cases for this experiment were on provider to customer links.

Figure 5a displays pre vs. post attack flow density for the same betweenness link sample in Fig. 4a, filtered to include only provider to customer links. Here we see results for the ideal case for the attack: an adversary AS located in the customer cone of the To AS, when the target link is a provider to customer link. The region between the curves in this figure represents the attacker's gain from executing Maestro. Before the attack, most sampled links have flow densities below 10% - that is, most link targets are vulnerable to 10% or fewer bots in a Coremelt LFA. **After Maestro execution, roughly half of sampled links have >50% flow density**.

Customer to Provider Link Attacks: Our previous experiment highlights the roles that target link relationship and adversary position play in attack path prevalence. Since most Internet services require bidirectional communication, the simplest method for attacking customer to provider links is to attack the opposite direction; i.e., target the associated provider to customer link. To confirm this method's viability, we reverse the target link direction for all customer to provider links from our betweenness link sample set with <1% post-attack flow density in the prior experiment. We then sample adversaries from the new To AS customer cones and simulate attacks. Figure 6 shows the results of these reversed attacks. Clearly, link relationship was the primary culprit preventing attack success, as attacking the reversed direction yields drastically improved flow density. For most links, we see 50% or greater post-attack flow density, meaning that we expect to engineer most bot traffic onto the target.

However, if an adversary has already compromised an AS in the To AS customer cone of a customer to provider link, the adversary cannot always attack the reversed direction; this requires compromising a different AS. So, we ask now under what conditions the customer to provider direction can be successfully attacked. The only attack paths available in this case originate from within the From AS customer cone, because any flow sources *not* located there must transit a peering or provider to customer link before reaching the target link. This means that potential targets are limited to those with significantly bot-infected From AS customer cones. To test our ability to steer bot traffic in these scenarios, we randomly sample 300 links from above the 9th deciles for From AS customer cone infection rate from the set of all customer to provider links, simulate attacks, and measured flow density improvement. We find that we can exert significant steering influence on bots in the From AS customer cone as shown in Fig. 7.

While the Maestro attack is most effective for attacking links between customers and providers, the concept of leveraging routing capability to expose links to attack could be extended to peer links, as well. While the details of such an attack fall outside the scope of our paper, we will briefly sketch one possible technique. Peer links (like customer to provider links) are generally only exported within the peers' customer cones, so an adversary wishing to expose such a link would require routing capability at an AS located within the link endpoints' cones. From that position, an adversary could trigger an intentional route leak to a destination that crosses the peering link to expose it to attack; this is analogous to the role of poisoned advertisements in the Maestro attack. Further work could explore the effectiveness of this technique.

5 Internet Experiments

We built an experimental Maestro implementation with the PEERING BGP testbed [44] and the RIPE Atlas framework [51] to 1) validate the basic steering mechanisms behind Maestro and 2) explore what real-world dynamics could affect the attack. PEERING allows us to originate routes from its points of presence (PoPs) worldwide hosting BGP edge routers. RIPE Atlas provides a distributed probe network for data plane measurements including ping, traceroute,

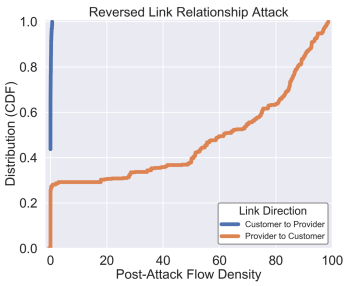


Fig. 6. Reversing relationship to attack cust. to prov. links.

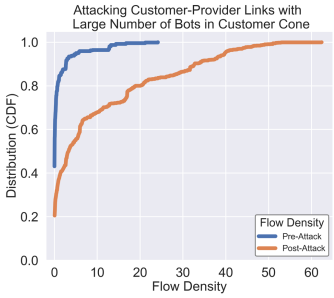


Fig. 7. Attacking cust. to prov. links.

DNS, SSL/TLS and NTP. A subset of these probes, called *anchors*, periodically form a publicly accessible "mesh" measurement of all-to-all traceroutes.

A single PEERING PoP serves as the adversary's compromised AS. All responsive RIPE Atlas anchors ($n \approx 580$) across roughly 470 ASes function as the experimental "botnet". To measure flow density, we combine anchor mesh measurements with traceroutes from anchors to the compromised AS, mapping IP-level hops to AS-level ones using Team Cymru's mapping service [13]. Total flow density includes all anchors with paths to one another or the compromised AS that transit the target.

We conduct the "attack" by first selecting a target from links within three AS-level hops of the compromised AS. Next, we originate an unpoisoned/24 advertisement from the compromised AS and measure pre-attack flow density. Finally, we iteratively select poisons and issue poisoned advertisements for the/24 as described in Sect. 4 to steer traffic over the target link. After the attack, we issue new traceroutes to compare achieved flow density with the original measurement. We perform this procedure on two target links/PEERING PoP pairings.

Ethics: All control plane advertisements are protocol compliant, and all data plane measurements target PEERING prefixes that host only our experimental workstation, so no Internet traffic was affected except our own. Measurements were made with Paris traceroutes [2] using 3 packets of 48 bytes each originating from fewer than 600 RIPE Atlas anchors. The total additional network traffic on any affected link is less than 100KB per measurement. Measurements are spaced by at least one minute. We assess that the potential for our experiments to disrupt normal network operation is minimal, and we did not receive any complaints from operators.

Clemson Results: The Clemson University PoP is the compromised AS for our first experiment. We select 209 ↦ 2722, a link from the Tier 1 clique to the compromised AS's upstream provider AS2722, as our target. This link is two topological hops from the compromised AS. Zero bots transit this link to one another or to the compromised AS prior to the attack; we identified the link from CAIDA's topology [56]. So, the pre-attack bot-to-bot flow density was 0%. This does not indicate the link is unused, only that it is not used to transit traffic between bots and the compromised AS.

AS174 is the most common AS on inbound bot paths and would be our first poison choice, but poisoning this AS disconnects the majority of bot ASes. We observe this behavior in both Internet experiments when attempting to poison Tier 1 ASes, so we suspect that defensive filtering of Tier 1 poisons [24] is to blame. Because AS2722's direct connections are still able to reach the compromised AS, we suspect that filtering occurred at Tier 1 providers. To avoid such disconnections, we skip AS174 and continue the algorithm. After finding 2 poisons, nearly all (533/576 bots) transit the target 209 ↦ 2722. The remaining bots either directly connect to AS2722 (20 bots in AS15169 Google) or were disconnected from the compromised AS (13 bots). **Maestro execution enhances flow density from 0% pre-attack to over 90% after.**

Utah Results: The PEERING PoP behind Utah Education Network (AS210) is our second compromised AS. We selected a Tier 1 to Tier 1 link 3 hops from the compromised AS, 174 ↦ 209, as the target. The original flow density on this link is about 7% - 45/576 bots had bot-to-bot paths that transited this target link before the attack, and none used it to reach the compromised AS.

Our first poison choice would be AS1299, but poisoning this Tier 1 AS triggers the disconnection of most bot ASes as in the Clemson experiment. We again suspect Tier 1 ASes filtering Tier 1 poisons is to blame, so we continue the algorithm without poisoning AS1299 or other Tier 1 providers. The attack finds 3 poisons that steer roughly 16% (91/576) of bots over the target link to the compromised AS. 106 bots transited the target link to reach either another probe or the compromised AS. **Overall, the attack modifies probe flow density from 7% to above 18%, a greater than 2x improvement in flow density.**

Discussion: We find substantial evidence of Tier 1 filtering of Tier 1 poisons during our experiments consistent with a recent BGP interception study, SICO [4]. We note that SICO's path export communities could provide a mechanism to further concentrate traffic when Maestro is limited by Tier 1 poison filtering. We explore filtering in greater detail in Sects. 7 and 9.

6 Attack Scope and Vulnerability

As shown in Sect. 4.2, Maestro adversary is most successful when they 1) target a provider to customer link and 2) compromise a direct/indirect customer of the

To AS. In this section, we explore the Internet's vulnerability to two Maestro attacker types: strategic and opportunistic. The *opportunistic adversary* achieves routing capability at some arbitrary AS, and exploits their position to attack upstream links, e.g. as part of a ransom DDoS. For this attacker, a key question is how many more critical links the adversary can attack with Maestro. The *strategic adversary* targets a specific link or links for broader strategic purposes, e.g. disrupting key services. For this attacker, we want to quantify the pool size of potential attacking ASes given some target link.

Opportunistic Adversary: To answer how Maestro enables the opportunistic adversary, we compare the number of links the adversary can attack with Maestro to the number that can be attacked with a Crossfire LFA. For every AS x in CAIDA's inferred topology [56], we classified upstream links as vulnerable to Maestro if they were provider to customer links where the To AS is 1) a direct/indirect provider of AS x and 2) within 3 topological hops of AS x. We counted links as vulnerable to Crossfire if their simulated pre-attack bot-to-any flow density was over 10%. We call this targeted flow density level the *vulnerability threshold*. We averaged the results by the compromised AS's UCLA classification [40]: stub AS, small ISP, large ISP. Note Tier 1 networks are excluded from these results, as they are not customers of any AS and thus cannot meaningfully execute Maestro. For this analysis, we restrict our focus to core (above 7th decile betweenness) vulnerable links.

The results are displayed in Table 1. In the Maestro columns, we display the average and standard deviation number of Maestro-vulnerable links for each AS classification. The Crossfire columns show the same metrics for links with 10% flow densities without Maestro. **We find that in all AS classes, Maestro increases the pool of potential LFA targets by at least 2x on average**. Small ISPs have the greatest average number of Maestro-vulnerable upstream links (40). This is consistent with our understanding of position-driven vulnerability - small ISPs have, on average, more direct providers than stub ASes and a greater number of provider hops to the Tier 1 clique than large ISPs. This means they have the highest number of provider to customer links within 3 upstream hops, and thus the largest pool of potential Maestro targets.

Table 1. Opportunistic Maestro: key links exposed to Maestro vs. Crossfire.

	Maestro Vuln.		**Crossfire Vuln.**	
	Avg	**Stddev**	**Avg**	**Stddev**
Stub AS	29	35	14	16
Small ISP	40	49	19	23
Large ISP	35	41	16	18

Strategic Adversary: The strategic adversary is motivated to attack specific links in the topology. For this attacker, Maestro's key capability is its ability to focus traffic on specific upstream targets, including those isolated from LFA flows by topological constraints. Because this attacker seeks to attack a specific link, we want to understand both the number of ASes positioned to attack the target and the attack's potency relative to existing LFAs.

For this analysis, we consider all links in the betweenness link sample set described in Sect. 4.2, which contains randomly selected high, intermediate, and low betweenness links. We analyze the results of the Maestro attack simulations from Sect. 4.2 and determine the proportion of links over a given vulnerability threshold before and after attack execution. The results are presented in Table 2. **Less than 50% of these links have 25% or greater pre-attack flow density across all botnet models; after the attack, greater than 95% of these links are above this threshold**.

In addition to the attack effect, we also want to know how many ASes are positioned to attack each link. The adversary sample set columns in Table 2 display, for each vulnerability threshold, the average set size of potential ASes that an adversary could compromise to perform the attack. To calculate this number, we first note that adversary AS size is not a significant determinant of attack success - the absolute value of the correlation coefficient of CAIDA AS rank [55] and flow density gain for these attacks was always smaller than .01. Success for adversaries in the customer cone is instead dominated by relative topological position. For this reason, if an AS two AS-level hops into the To AS customer cone successfully executes Maestro at some threshold, we expect every other AS at the same or closer depth to likewise succeed at that threshold.

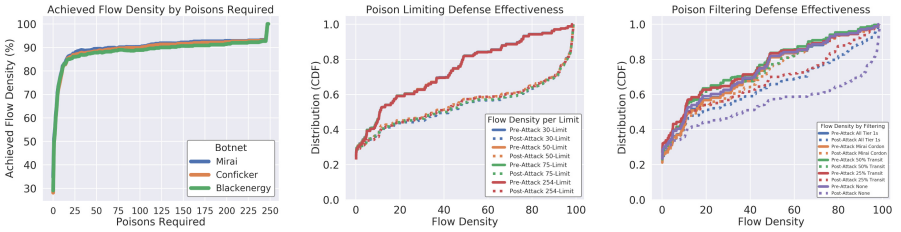
Out of about 600 provider to customer attacks in this sample set, we found that fewer than 3% violated this expectation for any botnet model. So, we estimate the potential adversary pool as the To AS customer cone to the depth of the deepest successful adversary. The standard deviation is also presented for these adversarial sets. Note that the high variance in set size expressed by these statistics illustrates that the potential adversary pool size varies greatly. For target links with large ISPs as the To AS (like core Internet links), the To AS customer cone can include thousands of potential adversaries. For many smaller targets, only a handful of ASes are well-positioned for the attack.

Table 2. Strategic Adversary: Maestro result summary, provider to customer betweenness link selection, customer-only attack

	Vuln. Before			Vuln. After			Avg Adversaries			Stddev Adversaries		
Vuln. Threshold	25%	50%	75%	25%	50%	75%	25%	50%	75%	25%	50%	75%
Mirai	0.44	0.18	0.09	0.97	0.91	0.87	135.45	114.64	91.22	349.69	312.94	215.85
Conficker	0.44	0.21	0.08	0.96	0.91	0.85	151.25	100.01	92.35	360.26	226.39	216.06
Blackenergy	0.38	0.23	0.10	1.00	0.94	0.87	130.61	103.98	92.48	335.06	273.92	217.14

7 Towards Defenses

Two broad categories exist for defense against this attack: general LFA defense solutions and solutions targeting poisoned announcements. Unfortunately, many state-of-the-art LFA defense options are not widely available to network operators, as they require collaboration across ASes [29], deployment of next-generation architectures [12], or additional hardware [25,31,42]. Nyx, a re-routing system [49] could only partially mitigate Maestro by moving traffic between some critical AS and the Nyx deployer AS off the congested target. Here we consider the more relevant mitigations that target Maestro's poisoned advertisements.



(a) Flow density CDF by number of poisons

(b) Flow density CDF against path filtering

(c) Flow density CDF by various AS filtering sets

Fig. 8. Evaluation of Maestro defenses

The path length defense - rejecting advertisements above some limit - is one easy-to-implement response to this attack. Unfortunately, as shown in Fig. 8a, nearly all of the attack effect is achieved with 5 or fewer poisons; this is partially an artifact of relatively dense botnet distributions as discussed in the next mitigation. Tran et al. [54] observed AS PATH lengths commonly reach 30 hops in legitimate advertisements, indicating that adding 5 hops to the AS PATH is likely insufficient to distinguish attack advertisements. Figure 8b shows the impact if all ASes limited advertisement AS PATH length at various levels, including those observed by prior work (30 and 75) [54] and implemented in common routing hardware (254) [23]. These limits *do not decrease achieved flow density.*

An alternative defense involves detecting and suppressing poisoned BGP advertisements. The current and potential effects of RPKI and BGPSec filtering is discussed in Sect. 2.2. Alternatively, ASes could prevent Meastro with by filtering all poisoned advertisements. This is a feasible approach, because poisoned advertisements have a clear signature. However, as previously discussed, this would prevent benign traffic engineering poisons. ASes are also under unique administrative control, so uniform filter deployment is a challenge.

Figure 8c shows how our betweenness-based link selection/customer only adversary selection experiment responds to different sets of filtering ASes. For this mitigation trial, we ran our attack against four filtering sets. The first two

sets place poison filters at 25% and 50% of all transit ASes (those with one or more customers) filtering all poisoned advertisements. The next two sets are smaller but more strategically targeted: filters at all 20 Tier 1 ASes (Hurricane Electric included) to explore how the best-provisioned networks can protect the Internet as a whole, and botnet-specific defenses. The botnet-specific defense springs from the observation that all three botnets in our study are highly concentrated. In the Mirai model, for example, 64% of bots are hosted in just 30 ASes. So, for our final filtering set, we include all providers (58) for these 30 ASes to *cordon* the botnet. While this filtering pattern is botnet-specific, it illustrates the efficient protection that few well-positioned filterers provide relative to many randomly distributed ones. **Both the 50% transit and Mirai cordon filtering sets eliminate virtually all Maestro effect**. Interestingly, filtering at Tier 1s (20 ASes) provides greater overall protection than filtering at 25% of transit providers (2441 ASes). While prior work suggests short (<50 poisons) poisoned announcement filtering is rare [48], Tier 1 filtering is encountered in [4].

8 Related Work

The Coremelt [52] and Crossfire [26] attacks are discussed in detail in the background, Sect. 2.3. Bellovin's link cutting attack [3] discussed strategically cutting topological links to route traffic through colluding ASes. While the purpose of that attack - interception in the presence of path security - is different, the concept of severing links to re-route traffic is similar. Classifying links by BGP betweenness is a technique employed in Schuchard et al.'s Coordinated Cross Plane Session Termination (CXPST) control plane attack [46]. Select LFA mitigation work is presented in Sect. 7. Other uses of BGP poisoning include LIFEGUARD from Katz-Bassett et al. [27,28] as well as Colitti et al. and Anwar et al.'s policy exploration studies [1,11]. Nyx [49] from Smith et al. employs BGP poisoning for DDoS mitigation. The propagation of poisoned advertisements throughout the Internet is actively measured in [48].

9 Conclusion

In this work we explored both LFA limitations and how adversaries can overcome those limitations. Our experiments show that contrary to assumptions in previous literature, botnet-sourced LFAs cannot target arbitrary links with full force in practice. In fact, many core Internet links can be reached by just a fraction of infected hosts in all three of our botnet models. Our simulations show the Maestro attack can partially overcome topological reach restrictions. Most troublingly, high betweenness links are most vulnerable to this attack, and the rank of AS adversaries plays little role in attack success. Provider to customer targets are most vulnerable to Maestro, but other links with significantly infected From AS customer cones can be affected by the attack. Our mitigation experiments suggest strategic poison filtering drastically reduces vulnerability.

Operator Engagement: We submitted a preprint of this work to the NANOG (North American Network Operators Group) mailing list to solicit feedback on the attack and disseminate mitigations. Responses indicated that operators had not seen any similar attack executed in practice. Some operators suggested that defensive filtering (a "peer lock" mechanism) could provide some protection from the attack [24]. Peer locking validates advertisements against known relationships, an intuitive step in averting path manipulations like those used in Maestro. However, this requires periodic out-of-band information exchange between ASes, and there is little evidence that this feature has penetrated beyond Tier 1 ASes [36]. We encountered some evidence of Tier 1 peer locking in Sect. 5, as did Smith et al. [48] in their measurement study. Our Tier 1 filtering examination from Fig. 8c provides a window into the effect of full Tier 1 peer locking.

References

1. Anwar, R., Niaz, H., Choffnes, D.R., Cunha, Í.S., Gill, P., Katz-Bassett, E.: Investigating interdomain routing policies in the wild. In: ACM IMC (2015)
2. Augustin, B., et al.: Avoiding traceroute anomalies with Paris traceroute. In: ACM SIGCOMM (2006)
3. Bellovin, S.M., Gansner, E.R.: Using Link Cuts to Attack Internet Routing (2003)
4. Birge-Lee, H., Wang, L., Rexford, J., Mittal, P.: SICO: surgical interception attacks by manipulating BGP communities. In: ACM CCS (2019)
5. Chang, W., Mohaisen, A., Wang, A., Chen, S.: Measuring botnets in the wild: some new trends (2015)
6. Chung, T., et al.: RPKI is coming of age: a longitudinal study of RPKI deployment and invalid route origins. In: ACM IMC (2019)
7. Cisco: Cisco IOS and IOS XE Software Cluster Management Protocol Remote Code Execution Vulnerability. https://bit.ly/3aFfFhN
8. Cisco: Cisco IOS XE Software AAA Login Authentication Remote Code Execution Vulnerability. https://bit.ly/2RmkB3o
9. Cisco: Cisco IOS XE Software Static Credential Vulnerability. https://bit.ly/2RnyjmA
10. Cisco: Cisco REST API Container for IOS XE Software Authentication Bypass Vulnerability. https://bit.ly/2NVkIB5
11. Colitti, L., et al.: Internet Topology Discovery Using Active Probing (2006)
12. Cristina, B., et al.: SIBRA: scalable internet bandwidth reservation architecture (2016)
13. Cymru, Team: The Team Cymru IP to ASN lookup page. https://www.team-cymru.com/IP-ASN-mapping.html
14. Dainotti, A., et al.: Analysis of country-wide internet outages caused by censorship. In: ACM SIGCOMM (2011)
15. Demchak, C.C., Shavitt, Y.: China's maxim - leave no access point unexploited: the hidden story of China telecom's BGP Hijacking. Mil. Cyber Aff. (2018)
16. Donnet, B., Bonaventure, O.: On BGP communities. In: ACM SIGCOMM (2008)
17. Madory, D.: BGP Hijack of Amazon DNS to Steal Crypto Currency (2018). https://bit.ly/37vW2Ha
18. Gao, L.: On inferring autonomous system relationships in the Internet. In: IEEE/ACM ToN (2001)

19. Huston, G.: AS65000 BGP Routing Table Analysis Report (2020). http://bgp.potaroo.net/as2.0/bgp-active.html
20. Gilad, Y., Cohen, A., Herzberg, A., Schapira, M., Shulman, H.: Are we there yet? On RPKI's deployment and security. In: NDSS (2017)
21. Goldberg, S.: Why is it taking so long to secure internet routing? CACM (2014)
22. Google Security and White Ops: The Hunt for 3ve (2016)
23. Pepelnjak, I.: Limit the maximum BGP path length (2009). http://wiki.nil.com/Limit_the_maximum_BGP_AS-path_length
24. Snijders, J.: NTT Peer Locking (2016). http://instituut.net/job/peerlock_manual.pdf
25. Kang, M.S., Gligor, V.D., Sekar, V.: SPIFFY: inducing cost-detectability tradeoffs for persistent link-flooding attacks. In: NDSS (2016)
26. Kang, M.S., Lee, S.B., Gligor, V.D.: The crossfire attack. In: IEEE S&P (2013)
27. Katz-Bassett, E., et al.: Reverse traceroute. In: Usenix NSDI (2010)
28. Katz-Bassett, E., et al.: LIFEGUARD: practical repair of persistent route failures. In: ACM SIGCOMM (2012)
29. Lee, S.B., Kang, M.S., Gligor, V.D.: CoDef: collaborative defense against large-scale link-flooding attacks. In: ACM CONEXT (2013)
30. Lepinski, M., Sriram, K.: RFC 8205 - BGPSEC protocol specification. IETF (2013)
31. Liaskos, C., Kotronis, V., Dimitropoulos, X.: A novel framework for modeling and mitigating distributed link flooding attacks. In: IEEE INFOCOM (2016)
32. Litke, P., Stewart, J.: BGP hijacking for cryptocurrency profit (2014)
33. Luckie, M., et al.: AS relationships, customer cones, and validation. In: ACM IMC (2013)
34. Lepinski, M., Kent, S.: An Infrastructure to Support Secure Internet Routing (2012). https://tools.ietf.org/html/rfc6480
35. Majkowski, M.: Memcrashed-Major amplification attacks from UDP port 11211 (2018)
36. McDaniel, T., Smith, J.M., Schuchard, M.: Flexsealing BGP against route leaks: peerlock active measurement and analysis. In: NDSS (2021, in press)
37. McPherson, D., Gill, V.: BGP MULTI_EXIT_DISC (MED) Considerations (2006)
38. Netlab360: Mirai Scanner (2017). http://data.netlab.360.com/mirai-scanner/
39. Nordström, O., Dovrolis, C.: Beware of BGP attacks. In: ACM SIGCOMM (2004)
40. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: The (in) completeness of the observed internet AS-level structure. In: IEEE/ACM ToN (2009)
41. Putman, C.G.J., et al.: Business model of a botnet
42. Ravi, N., Shalinie, S.M., Theres, D.D.J.: BALANCE: link flooding attack detection and mitigation via hybrid-SDN. IEEE Trans. Netw. Serv. Manag. **17**, 1715–1729 (2020)
43. Rekhter, Y., Li, T.: A Border Gateway Protocol 4 (BGP-4) (1995)
44. Schlinker, B., Arnold, T., Cunha, I., Katz-Bassett, E.: PEERING: virtualizing BGP at the edge for research. In: ACM CONEXT (2019)
45. Schuchard, M., Geddes, J., Thompson, C., Hopper, N.: Routing around decoys. In: ACM CCS (2012)
46. Schuchard, M., Mohaisen, A., Foo Kune, D., Hopper, N., Kim, Y., Vasserman, E.Y.: Losing control of the internet: using the data plane to attack the control plane. In: ACM CCS. ACM (2010)
47. Scott Sr, J., Winter Summit: Rise of the Machines: The Dyn Attack was Just a Practice Run, December 2016

48. Smith, J.M., Birkeland, K., McDaniel, T., Schuchard, M.: Withdrawing the BGP re-routing curtain: understanding and analyzing the security impact of BGP poisoning through real-world measurements. In: NDSS (2020)
49. Smith, J.M., Schuchard, M.: Routing around congestion: defeating DDoS attacks and adverse network conditions via reactive BGP routing. In: 2018 IEEE Symposium on Security and Privacy (SP) (2018)
50. Sriram, K., Montgomery, D.C.: Resilient Interdomain Traffic Exchange: BGP Security and DDoS Mitigation. NIST (2019)
51. RN Staff: Ripe atlas: a global internet measurement network. IP J. (2015)
52. Studer, A., Perrig, A.: The coremelt attack. In: ESORICS (2009)
53. Thomas, M., Mohaisen, A.: Kindred domains: detecting and clustering botnet domains using DNS traffic. In: WWW (2014)
54. Tran, M., Kang, M.S., Hsiao, H.-C., Chiang, W.-H., Tung, S.-P., Wang, Y.-S.: On the feasibility of rerouting-based DDoS defenses. In: IEEE S&P (2019)
55. UCSD-CAIDA: CAIDA AS Rank dataset (2019). http://as-rank.caida.org/
56. UCSD-CAIDA: CAIDA AS Relationship dataset (2019). https://bit.ly/2RpRWuv
57. Jacobson, V.: Traceroute Man Page. https://linux.die.net/man/8/traceroute