5-2019

The Maestro Attack: Orchestrating Malicious Flows with BGP

Benjamin Tyler McDaniel
University of Tennessee

To the Graduate Council:

I am submitting herewith a thesis written by Benjamin Tyler McDaniel entitled "The Maestro Attack: Orchestrating Malicious Flows with BGP." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Maxfield Schuchard, Major Professor

We have read this thesis and recommend its acceptance:

Scott Routi, Bruce Maclennan

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

The Maestro Attack: Orchestrating Malicious Flows with BGP

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Benjamin Tyler McDaniel

May 2019

Dedicated to Burt and Tammy McDaniel, my father and mother, who have given me a lifetime of enduring love, inexplicable patience, and deep insights.
I love you dearly.

Acknowledgments

I would like to recognize my colleagues at the UT Computer Security Laboratory for their assistance on this work. Jared Smith's Internet routing knowledge and data science acumen were critical in experimental design and the analysis of results, respectively. He has been a key partner throughout this project. Joseph Connor's analytical insight was key in reformulating the optimal poison choice as MAX SAT. Kyle Birkeland's subject matter expertise as a network operator was a helpful source of intuition in algorithm development and understanding experimental results.

Finally, I would like to thank my advisor, Dr. Max Schuchard. Dr Schuchard's direction and instruction has been invaluable throughout this thesis project.

Abstract

We present the **Maestro Attack**, a Link Flooding Attack (LFA) that leverages Border Gateway Protocol (BGP) traffic engineering techniques to improve the flow density of botnet-sourced Distributed Denial of Service (DDoS) on transit links. Specific-prefix routes poisoned for certain Autonomous Systems (ASes) are advertised by a compromised network operator to channel bot-to-bot flows over a target link. Publicly available AS relationship data feeds a greedy heuristic that iteratively builds a poison set of ASes to perform the attack.

Given a compromised BGP speaker with advantageous positioning relative to the target link in the Internet topology, an adversary can expect to enhance flow density by *more than 30 percent*. For a large botnet (e.g., Mirai), the bottom line result is augmenting the DDoS by more than a million additional infected hosts. Interestingly, the size of the adversary-controlled AS plays little role in this effect; attacks on large core links can be effected by small, resource-limited ASes.

Link vulnerability is evaluated across several metrics, including BGP betweenness and botnet flow density, and we assess where an adversary must be positioned to execute the attack most successfully. Mitigations are presented for network operators seeking to insulate themselves from this attack.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Adversaries are exploiting long-known vulnerabilities in the Internet's routing architecture to launch increasingly sophisticated control-plane attacks. In 2014, security researchers discovered that a Canadian ISP surreptitiously hijacked bitcoin mining related traffic to steal victim miners' computational work, netting over $80,000 [27]. On an even larger scale, fraudulent networks designed to deceive advertisers into paying for automated ad views have raked in multimillion dollar hauls [45]. One such opeation, 3ve, persisted for years and raked in nearly $30 million [14].

The security industry partnership that eventually unravelled 3ve marvelled at its technical difficulty and professional execution - at its height, the operators were concurrently managing three distinct fraud operations. It is relevant to note that 3ve's operators registered their own Internet-level networks, or *Autonomous Systems* (ASes), and demonstrated a thorough technical knowledge of how to exploit this privileged position on the Internet. While a detailed analysis of 3ve is beyond the scope of this work, it is sufficient to note that steps taken to disguise 3ve - e.g. the seizure of derelict ASes to serve as pretend customers for the operators' AS - were effective in slowing its detection.

Distributed denial of service (DDoS) attacks are another scourge of the Internet. In short, these attacks direct traffic from many points on the Internet to a target or targets, in an effort to overwhelm the capacity of links or end hosts. As shown in Fig 1.1, hundreds of these attacks are launched every day. Sources for these flows are only growing more plentiful over time as the number of devices and services on the Internet expands. The

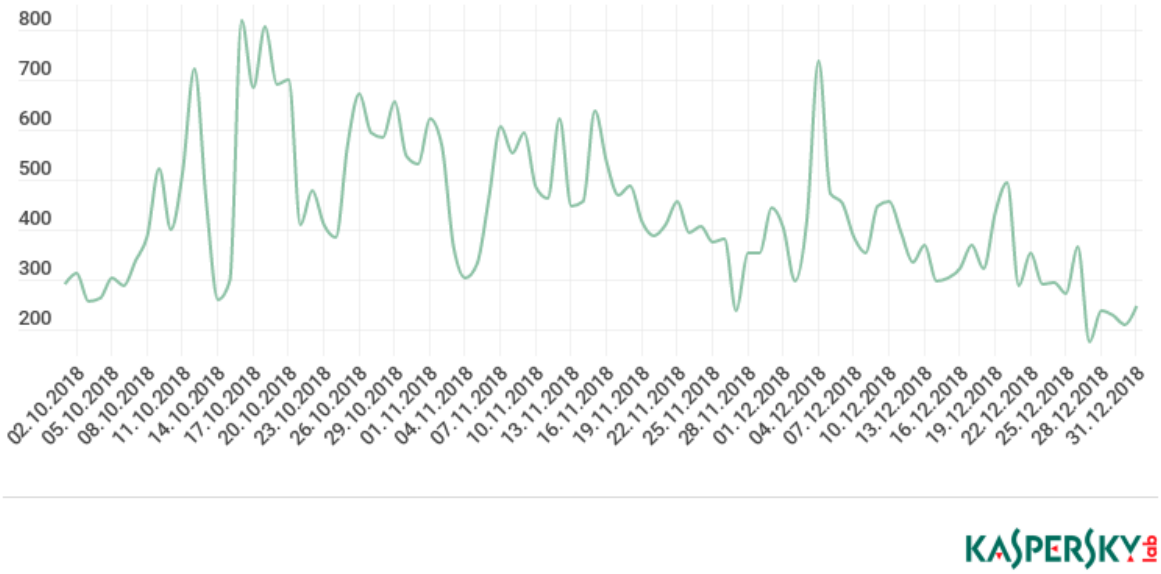


Figure 1.1: DDoS attacks in Q4 2018 [21]

billions of devices connected by the Internet of Things, for example, are already fueling DDoS attacks [23]. The development and adoption of novel Internet services is another source of potential attack flows. Unprotected memcached servers were recently used in a reflection attack that temporarily took Github offline [29].

Unfortunately, the Internet is not well-positioned to respond to this growing threat. The simplest and perhaps most effective response to volumetric DDoS is paying for mitigation services that, in general, divert traffic into a robust infrastructure to maintain availability for the purchaser during attack analysis/response [17]. This and other currently deployed solutions are ineffective in the face of more sophisticated methods that target infrastructure links rather than an end host, called *Link Flooding Attacks* (LFAs) [20, 41]. These novel attacks may have crossed from academic possibility to present threat: a 2016 Mirai attack directed over 500 Gbps attack to a provider in Liberia in what may have been an early attempt to execute an LFA [37].

We will demonstrate that Link Flooding Attacks (LFAs) are limited by Internet routing characteristics, and that these limitations can be partially defeated by a routing capable adversary. Our novel attack, **Maestro**, arises from the

confluence of the routing exploits and DDoS techniques presented in this section. Maestro allows an adversary with large traffic flows (from botnets or other sources) to channel them onto a target link with unprecedented control. If critical transit links in the dense core of the Internet are targeted, a strategy introduced in prior work [41], this attack could create broad disruption affecting thousands of peripheral networks served by the link. We will propose effective mitigations to prevent such an attack, and share insight into promising avenues for future work.

Our major contributions are as follows:

- **Measure how Internet routing properties limit Link Flooding Attacks.** These initial experiments motivate our attack, and are presented below in Section 1.1.
- **Develop a technique to overcome these limitations: The Maestro Attack.** We will demonstrate how traffic engineering techniques can be employed to increase the portion of an adversary's botnet that can attack a target link in Section 3. This effectively amplifies Link Flooding Attacks for already-vulnerable links and extends a botmaster's reach to new targets.
- **Evaluate our new attack via simulated attacks on Internet links.** We extend the Chaos BGP simulator (see Section 3.4) to execute thousands of attacks on links in a simulated Internet topology, varying target link selection and the adversary's relative position and quantifying our level of success. The results of these attacks are presented in Section 4.
- **Explore the relationship between link vulnerability and adversary position.** We summarize in Section 6 the insights we have derived from our experiments regarding where adversaries should be positioned for maximum effect on a target link.

1.1 Betweeenness and Link Flooding Limitations

Our first experiment is designed to illustrate the critical nature of select core Internet links by examining their relative usage. For this purpose, we classify links by *betweenness*,

defined as the number of times a link appears on the currently-used (best) path between any pair of ASes. High betweenness indicates that a link is used for transit between many ASes; a low betwenness link, on the other hand, serves relatively few ASes. Figure 1.2 shows the distribution of Internet links by betweenness, based on CAIDA's AS relationship inference [3] and the Chaos BGP simulator (see Section 3.4). The majority of links appear on 10 or fewer paths, indicating they are little used or peripherally located. But select links have a betweenness of more than *1 million*, providing connectivity between more than 1,000 AS source/destination pairs. Attacks on these critical links would play havoc with upstream/downstream networks, and could potentially threaten entire regions (as in the Liberia attack).

We observe that prior work on LFAs often 1) do not perform their measurements with distribution data from a real botnet [20], 2) assume botnets can direct significant flows over arbitrary links on the Internet [43], or 3) choose specific links based on botnet flows [41, 36]. We quantify link vulnerability via *flow density*, defined for now as the percentage of a botnet's infected hosts with paths to another bot over the target link. This metric is based on a Coremelt-style attack, where n bots generate n^2 flows by sending traffic to one another, a technique that makes attack flows appear "wanted" by the receiver and therefore increases the difficulty in distinguishing them from normal traffic (see Section 2.3.1).

Figure 1.3 depicts the results of our second experiment, measuring botnet flow density as a function of betweenness for all links in the inferred Internet topology. Note that some low betweenness (peripheral) links are, not unexpectedly, wholly outside the reach of this kind of attack. **Critically, some moderate to high betweenness (core) links are also partially or completely absent from paths between bots.** 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 Figure 1.4. In Section 3 we will introduce the Maestro Attack, a novel method of combining traffic engineering techniques with Link Flooding Attacks in an attempt to increase the flow density a botmaster can bring on to target links. First, however, we must provide some essential background information on Internet routing and LFAs.

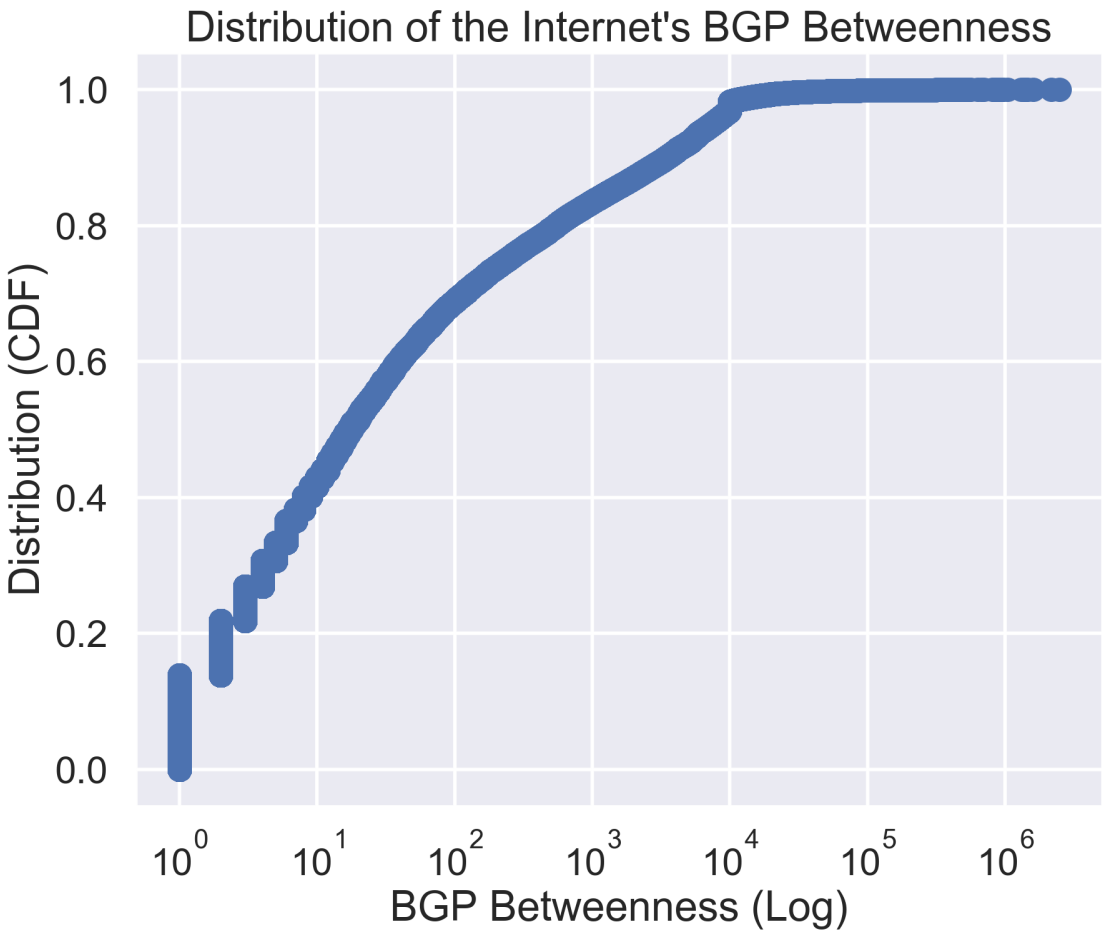


Figure 1.2: Betweenness of Internet links based on CAIDA inferred topology [3].

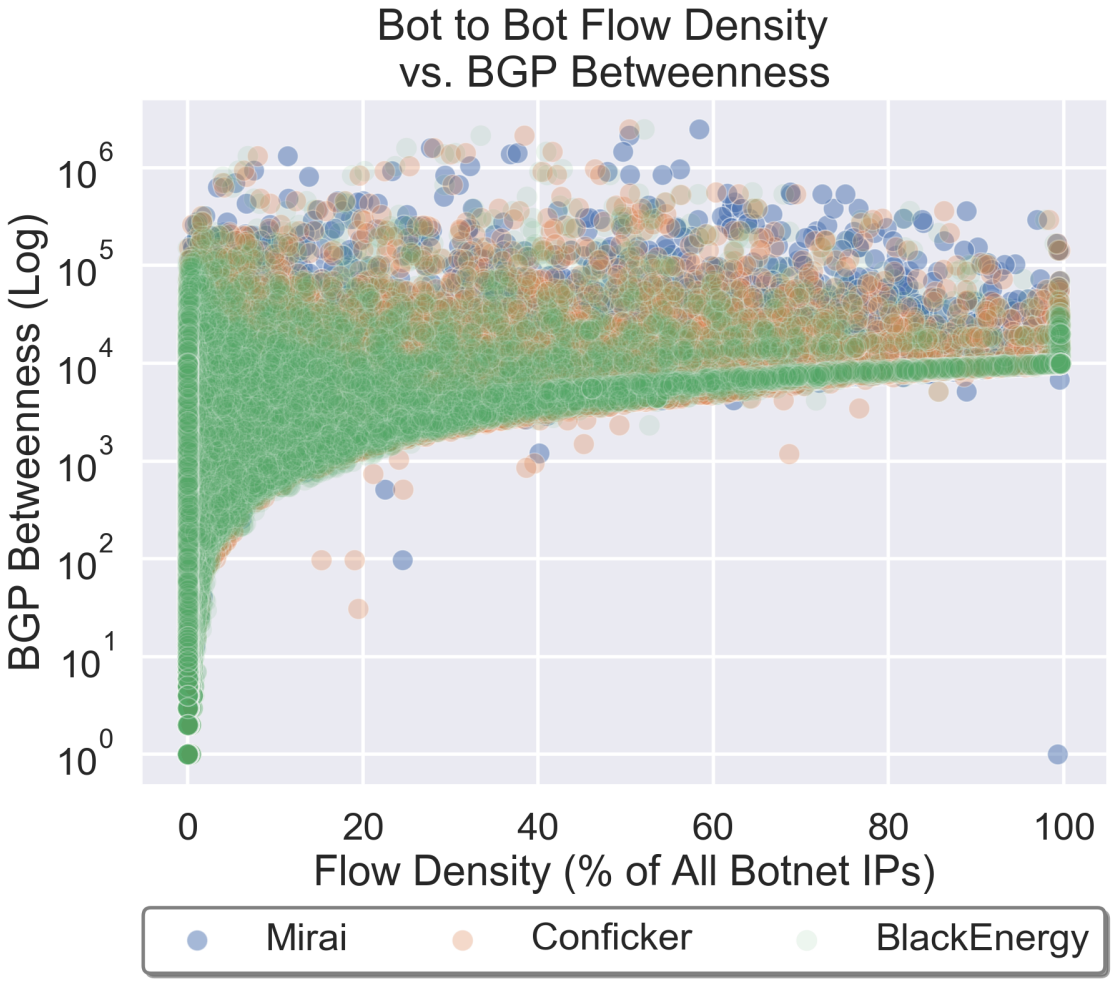


Figure 1.3: Flow density by betweenness.

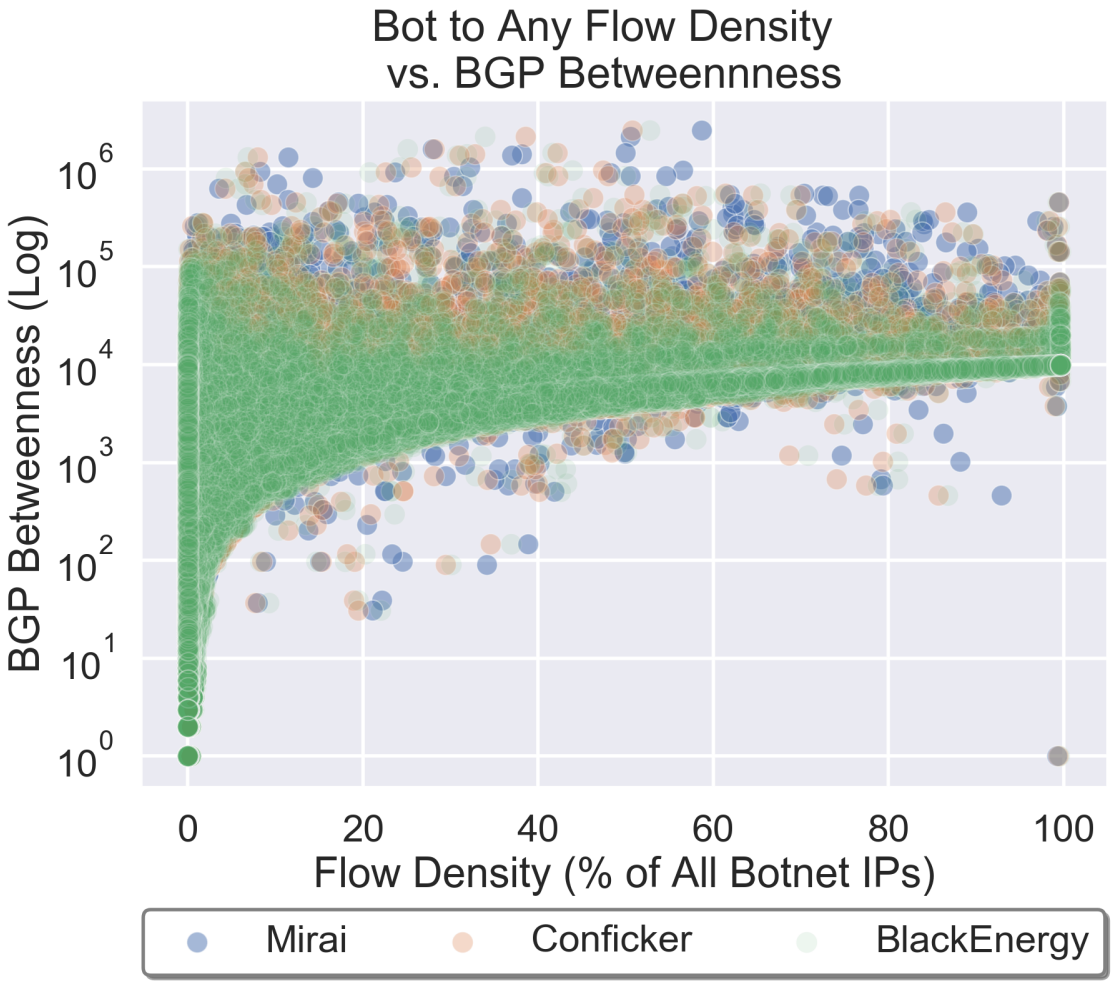


Figure 1.4: Extended flow density (bot to any destination) by betweenness.

Chapter 2

Background

Here we briefly describe the Internet's routing architecture and traffic engineering techniques to provide necessary context for our attack.

2.1 The Border Gateway Protocol (BGP)

While the Internet is most precisely described as a global network of interconnected routers, we can view it more abstractly as the composition of about 60,000 *Autonomous Systems* - or ASes - and their connections to one another. Each AS is a network of routers under singular administrative control [15] with a unique assigned identifier (an *ASN*). ASes exist to route traffic (in the form of IP packets) *internally* between hosts within the network and *externally* to other ASes. Each AS is directly connected to some number of other ASes as a *peer*, *customer*, or *provider*. A *customer-provider* relationship exists when one AS (the customer) compensates the other (the provider) to transit its traffic to/from the rest of the Internet. ASes in a *peering* relationship have agreed to a mutually beneficial relationship where traffic can be exchanged between them without compensation. To provide connectivity to their hosts, ASes assign IP addresses from their allocated blocks of IP addresses, called *prefixes*.

The *Border Gateway Protocol* (BGP) is the common language ASes use to communicate. BGP routes are defined by a destination IP prefix and a collection of attributes. Most notable among these attributes is the AS PATH, a sequence of ASNs describing the AS-level

hops along the path to the destination prefix. ASes originate routes to IP prefixes under their control and advertise them to the rest of the Internet via their neighboring ASes (see Figure 2.1). An AS's routers store all paths they learn about for one of the most important functions of BGP, the *decision process*. The decision process guides how routers select a best path to a destination prefix from all the routes they have been advertised. Importantly, routers will first filter out infeasible paths, including any route that contains their own ASN in the AS PATH. This provides BGP with a *loop-detection* mechanism. We will discuss later how this mechanism can be leveraged to selectively prevent route installation.

Feasible routes to the prefix are scored on their attributes, most notably AS PATH length and LOCAL PREF. LOCAL PREF is used to indicate the AS operator's level of preference for a path, informed by local policy choices regarding path qualities like desired next hops, and holds precedence over AS PATH length in the decision process. Shorter AS PATH length is used to break ties for paths with equal LOCAL PREF. Because the BGP decision process draws on path and policy attributes in route selection, it is categorized as a path-vector algorithm *with policies*. Upon receiving a packet, an AS's routers will compare the packet's destination IP to the prefixes for which it has installed a best path. The *longest prefix matching* rule dictates that the stored path with the longest (most specific) prefix match will be used to forward the packet to its next hop.

Once an AS selects a best path, it makes a propagation choice driven by economic incentive. If the path was learned about (and therefore leads through) a customer, the AS will propagate the routes onward to all direct connections to facilitate their customers' connectivity and increase their own compensation. Before advertising the route, the provider *prepends* their own ASN to the AS PATH, effectively extending the route to include themselves. Peers who receive a route advertisement will likewise prepend their ASN and forward it onward, but *only* to their customers, as they are not incentivized to provide transit to their peers or providers to the newly learned route. These route propagation behaviors shape the way paths are formed as they spread through the AS topology; the term [12]. Valley-free routing means that, in general, we expect that routes will never transit from a customer to a provider after transiting from a provider to a customer. An AS's *customer*

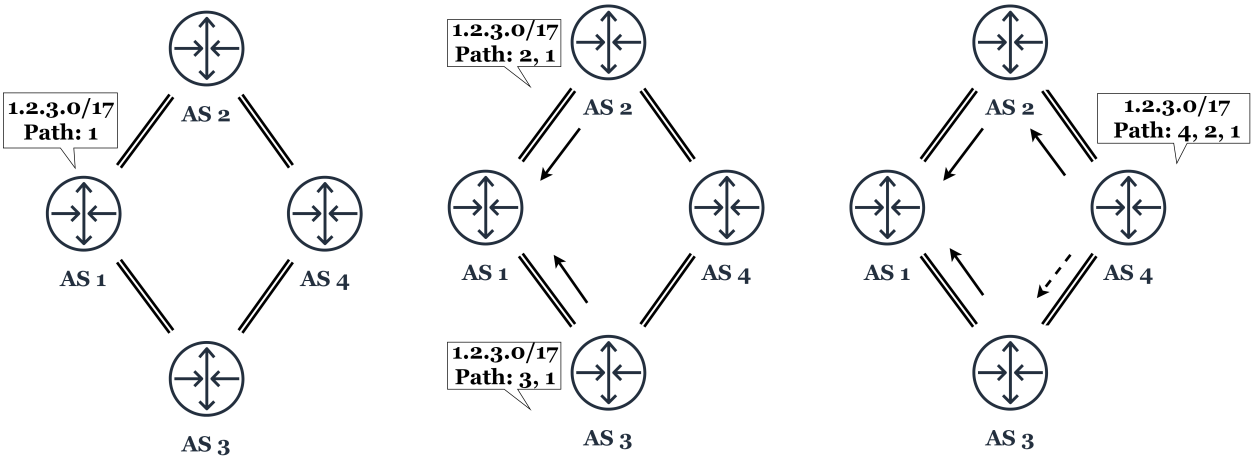


Figure 2.1: BGP routes built iteratively as they are propagated by neighboring ASes. Since 4 chooses path {2, 1} to reach 1, path {1, 3, 4} is not exported.

cone, defined as the set of all ASes reachable from an AS via only customer links, therefore have the highest visibility of routes advertised by the AS.

2.2 BGP Poisoning

As discussed in the previous section, the BGP decision process gives local operators control over (at minimum) the next hop outbound packets will take for any given prefix destination. Additionally, assuming compliant routing behavior, and ignoring short term disruptions caused by path changes along routes, the entire preferred outbound path can be selected. Unfortunately, BGP allows for relatively little control over the paths of inbound traffic. Some techniques do exist for engineering inbound flows, including the MULTI EXIT DISC (exit discriminator) attribute [30], BGP communities [11], and AS PATH or destination prefix manipulation, but all are subject to the traffic source AS's policy.

BGP poisoning is a traffic engineering primitive that allows for the manipulation of an AS's inbound traffic routes *without coordination* from other ASes [40]. BGP poisoning relies on two characteristics of BGP: loop detection and longest-prefix matching. An illustration of BGP poisoning is shown in Figure 2.3. The advertising or poisoning AS advertises a more

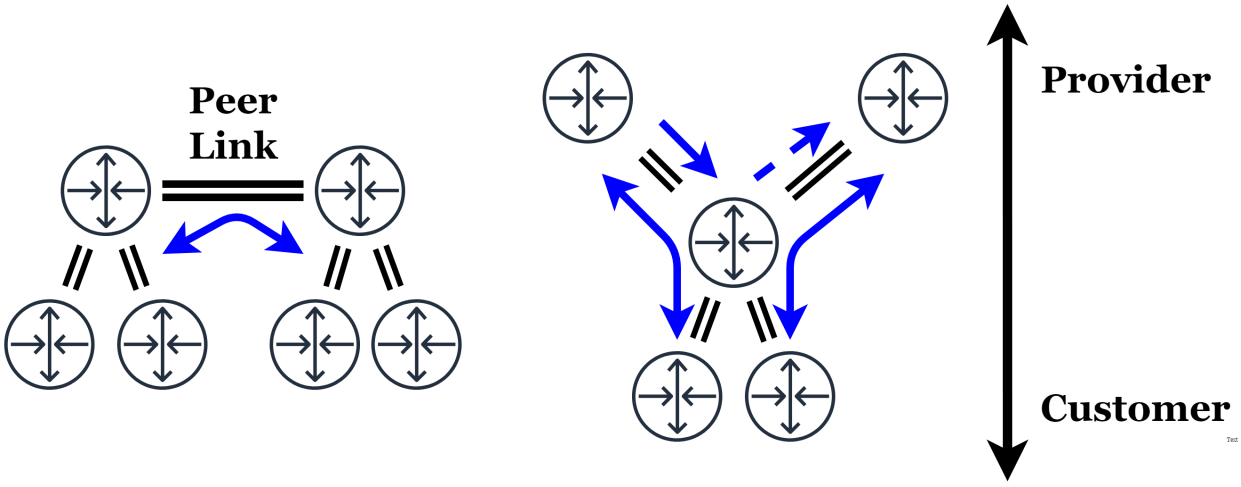


Figure 2.2: Valley free routing: ASes inform customers of all paths, but do not transit traffic for providers.

specific (longer) prefix for the traffic it wishes to move. As an example, Fig 2.3 depicts AS 1 advertising a longer /24 prefix compared to the /17 in Figure 2.1. Longest-prefix matching means that ASes directing traffic to the IPs within the prefix will switch on to the new route (see AS 2). However, an AS or set of ASes are included in the AS PATH for the advertisement, sandwitched between copies of the originator's ASN (in this case, 1). Because they are on the AS PATH, these ASes are *poisoned*; that is, they will detect a loop and drop the advertisement. Note that these ASes still have connectivity to the advertising AS's other prefixes. However, they do not have a path to the more specific prefix, and their traffic flows are unchanged by the advertisement.

2.3 Botnets and Distributed Denial of Service (DDoS)

Distributed denial of service (DDoS) is the term used to describe a network attack sourced from multiple, coordinated hosts. In this work, we will only discuss the most brutally straightforward form of DDoS: volumetric DDoS (Fig 2.4). Traditionally, a volumetric DDoS simply requires that the attacker pour more traffic into a host or service than the target can withstand. The result is a partial or total degradation of the network service until the attack can be mitigated. These attacks are more than a nuisance - they are employed in

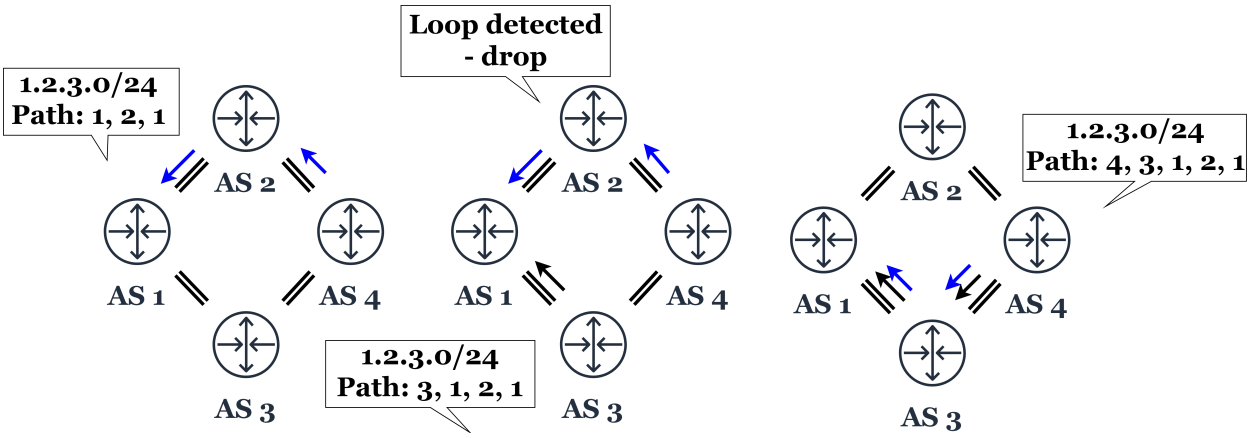


Figure 2.3: BGP poisoning. AS 1 advertises a specific prefix (thicker arrow). AS 4's traffic to AS 1 (blue) is moved to the more specific route. AS 2 is said to have been *poisoned*.

nation-state level attacks [5, 16] and can isolate or degrade Internet performance for large geographic regions [1].

Often the traffic source for these attacks are *botnets*, which are networks of compromised end hosts (bots) under an attacker's control. Because these networks are often large and well dispersed among many ASes, their small per-bot flows are difficult to filter/classify, but their aggregate traffic volume can be devastating [1]. We analyze our attack using three different botnet families, classified by the malware used to infect/control bots, each with distinct characteristics. The Mirai worm was one of the first to infect Internet of Things devices [23]. These small, resource-constrained hosts are generally poorly secured and plentiful, allowing Mirai-based networks to generate flows greater than 1Tpbs [1]. Conficker is an older, more traditional worm targeting Windows machines with advanced self-propagation mechanisms that has infected millions of victims [38]. Blackenergy botnets are based on malware developed and primarily distributed in Eastern Europe, often spread via infected Microsoft Office documents. In 2015, a Blackenergy botnet was used to launch a power grid attack in Ukraine that resulted in outages for over 200,000 consumers [5].

2.3.1 Link Flooding Attacks

A more recent class of DDoS attacks, *Link Flooding Attacks* (LFAs), targets network infrastructure rather than end hosts. One of the first such attacks in the literature is

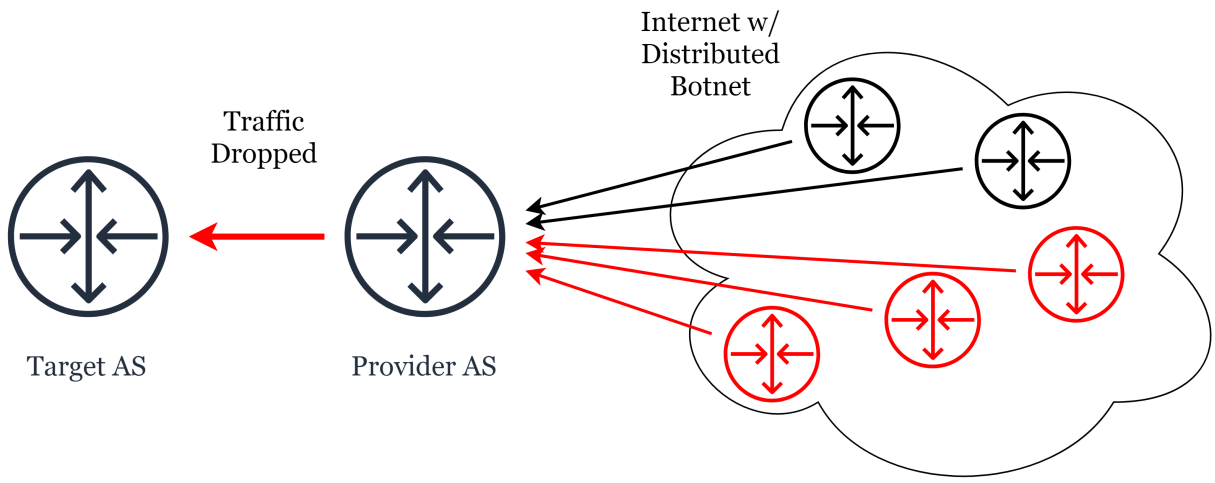
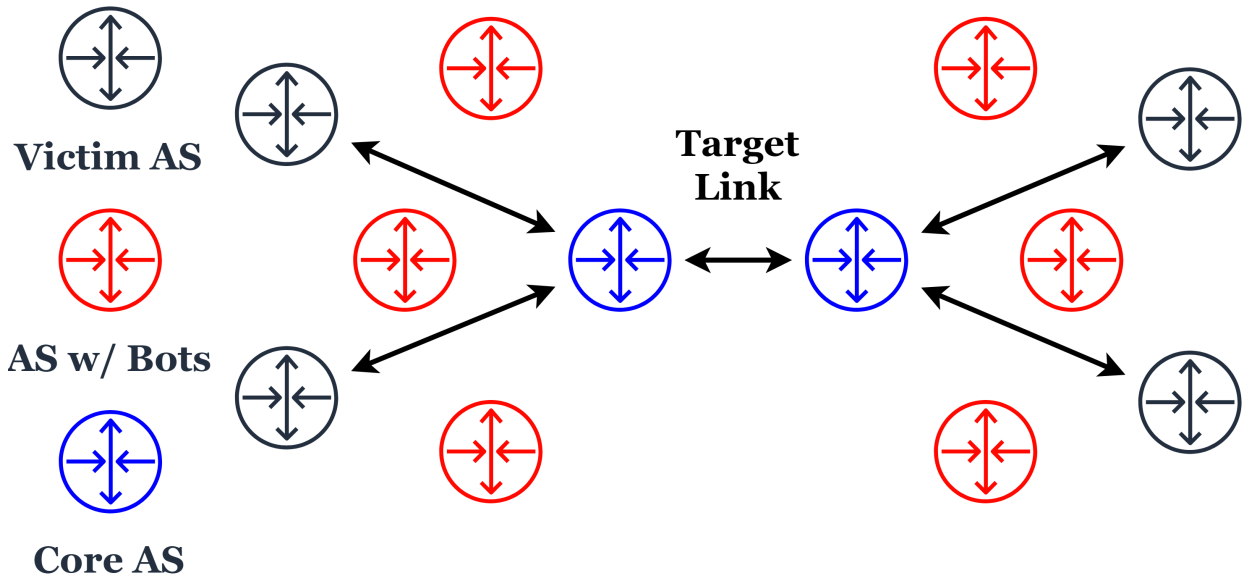


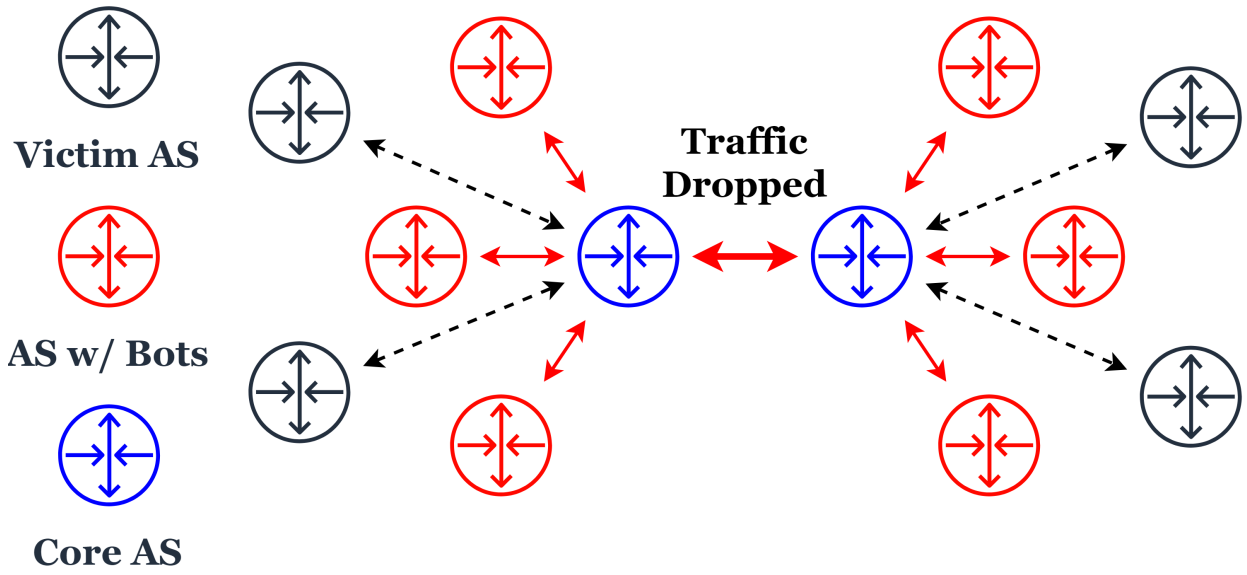
Figure 2.4: DDoS: a botnet with infected hosts in multiple ASes launches an attack against the Target AS. Bot traffic (red) overwhelms normal traffic to the target, which is lost.

Coremelt [41]. To execute a Coremelt attack, bots in a botnet 1) map which links are present on paths on routes between them, 2) target a specific link used on paths between many bots, and 3) direct bot traffic *to other bots* over the link. The resulting n^2 flows (for n bots with paths over the link) overwhelms benign traffic on the target link. The bot traffic is especially difficult to classify/filter as it is "wanted" by the destination host and therefore appears legitimate.

The Crossfire attack, like Coremelt, targets links in the Internet topology, but has the more ambitious goal of isolating an entire region (military installation, university, geographic region, etc) [20]. Rather than directing traffic to one another, bots map paths to publicly available web services (decoys) that result nevertheless in flows transiting target links. Bots use sustained, low-intensity flows to these services to execute the attack, a pattern that makes Crossfire extremely difficult to detect and counter.



(a) Core transit link carries traffic between ASes



(b) Botnet flows congest link

Figure 2.5: The Coremelt attack: bots direct traffic to one another over the target link.

Chapter 3

Methodology

3.1 The Maestro Attack

As discussed in Section 2.1, BGP allows network operators to apply their own policies to select outbound routes for any given destination prefix. Hosts within an AS (including bots) have no such control; their traffic follows routes chosen by the network operator. This limits choice of targets for a Link Flooding Attack (LFA), because bots cannot always find a destination for their traffic that crosses an arbitrary link on the Internet (see Section 1.1). The result is that very few links can be hit with the full force of a distributed botnet, and many cannot be affected at all.

The central insight of the Maestro attack is that while an adversary cannot fully dictate *outbound* bot traffic paths, a routing-capable adversary can use BGP poisoning to alter *inbound* paths to themselves. If an adversary first directs bot traffic to the AS/prefix under adversarial control (the *compromised AS* or *adversary AS*), they can then orchestrate those flows onto a target link using BGP (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*. Note that the adversary cannot influence the route selection *process* in the ASes housing the bots; rather, BGP poisoning essentially *bypasses* route selection by presenting a more specific prefix than infected ASes have previously seen.

In effect, this also executes a traditional DDoS against the adversary AS. As discussed in Section 3.2, however, this may be of little concern to a motivated attacker. They may

have compromised or registered an AS for this very purpose, and/or have calculated that the temporary disruption to their own AS is worth degrading the target link. We will show that, given certain topological relationships between the compromised AS and target link, the adversary can expect a significant improvement in flow density on the target.

3.2 Threat Model

To execute this attack, an adversary requires 1) command of a botnet and 2) control of a BGP speaker, i.e., a router on the edge of an AS. 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 [32]. Recent events demonstrate that there are, unfortunately, multiple feasible avenues for malicious parties to achieve routing capability. The 3ve fraud operation [14], discussed in Section 1, demonstrates the most straightforward route - simply registering an AS. Network operators could also be hacked or compromised by an insider, as may have been in the case in the Canadian bitcoin hijack [27]. Finally, BGP has already been weaponized for intelligence gathering [10] and censorship [9] by nation states. While these more powerful adversaries have many tools at their disposal, they certainly have the leverage to execute the Maestro attack.

3.3 Algorithm

One core capability for the attacker is an algorithm to determine which ASes to poison to maximize inbound bot traffic over the target link. We call this set of ASes the *poison set*. These are ASes that will be sandwitched between the compromised ASN in the poisoned advertisement (see Section 2.2). Finding a poison set that successfully steers bot traffic is no trivial task, because poison sets are *conflicting*; that is, the poisons required to steer one bot-containing AS (or *source AS* onto the target link will disconnect 1) the poison set and 2) all ASes without a path to the poisoning prefix that does not transit the poison set. Also, we cannot precisely predict AS behavior; our expectations for how ASes will respond to poisons

are therefore based on inferred AS relationships from CAIDA [3] and the Chaos simulator (see 3.4.

3.3.1 Optimal Poison Choice

We can solve for the optimal poison set by re-framing the problem as MAX-SAT, a generalization of boolean satisfiability (SAT) where we seek to assign truth values to variables in order to maximize the number of satisfied clauses rather than achieve complete boolean formula satisfaction [44]. Consider that each source AS has a set of poison sets S that map to resulting paths P over the target link to the adversary AS, where each set $s \in S$ corresponds to a resulting path to the poisoning prefix $p \in P$ (that is, $s \mapsto p$ for that source AS). This signifies that if the adversary chooses a poison set that contains all of the ASes in s and none of the ASes in p, the source AS will shift onto path p. Note that, depending on the adversary AS and target link position relative to the source AS, S and P may be empty; in that case, there is no way to steer the source AS onto the target link.

In our boolean formula, the variables will be the ASes in the topology. We can define the structure of the boolean formula by building a clause for each source AS thusly: an AS appearing in a poison s is represented by its AS variable, and the ASes in the resulting path p are represented by the inversion of their AS variables. These variables are joined conjunctively, along with the source AS itself; if it is poisoned, it will of course not have a path to the poisoned prefix. We disjunctively join each source AS's conjunctive clauses, one for each $s \mapsto p$, to form a clause in disjunctively normal form for that source AS. This clause is the boolean representation of the poison choices we must make to bring the source AS onto the target link.

Finally, we join the all source AS clauses by conjunction, and we have defined a boolean formula that describes how our poison choices will affect the paths of ASes containing bots to the adversary. While we do not reformulate the problem in conjunctive normal form, it is always possible to do so [18].

Unfortunately, MAX SAT is APX-complete; no efficient algorithm can solve it, and no polynomial time approximation scheme can be devised (unless $P = NP$) [7]. This is problematic because exploring how the relative topological position of the adversary, target,

and flow sources requires the simulation of thousands of attacks. To enable this, we designed an efficient heuristic that exploits the specific structure of our problem.

3.3.2 Iterative Poison Choice Heuristic

We begin from the observation that despite the high runtime complexity of the problem, the adversary's goal is simple: selectively poison ASes on source AS paths to the adversary that *do not* cross the target link in an attempt to force source ASes to switch onto paths that *do* contain the target link. Intuitively, the adversary wants to form a bottleneck to the poisoning prefix over the target link.

Our poison choice heuristic (Algorithm 3.3.2) represents one of our major contributions. The algorithm works by first establishing a set of *sacred* ASes that should never be poisoned. This set is initialized with the from AS, the to AS, the compromised AS, and all ASes that appear on every path from the to AS to the compromised AS (naturally, we must have a path for traffic from the target link to the compromised AS). We will then iteratively build the poison set.

At each iteration, we 1) select an AS to poison from the source ASes that remain (i.e. those not already poisoned, disconnected by poisons, or marked sacred), 2) add it to the poison set, 3) measure the simulated response of source ASes to the new poison set, and 4) update the sacred set. We will terminate iteration when all ASes are either poisoned/disconnected from the poisoned prefix, successfully transiting the target link to the poisoned prefix, or marked sacred. An additional termination condition is reached if the poison set (which is included in the AS PATH as described in Section 2.2) causes the AS PATH to exceed the size AS operators will almost certainly filter in practice: around 254 hops [43, 40].

3.3.3 Example Attack

To further elucidate our heuristic, we present an example attack on a small toy topology in Fig 3.1. ASes above (closer to the top) of the figure provide for the linked ASes below them. Each subfigure displays the results at one iteration.

Algorithm: Poison Choice Heuristic

function Choose Poisons $(f, t, a, n, Sources)$

Input : from AS f, to AS t, adversary AS a, poison limit n, source ASes $Sources$

Output: poison set $Poisons$

$Poisons = \emptyset$

while $Sources \neq \emptyset \; and \; |Poisons| < n$ **do**

Setup:

$B = \{b \mid b \text{ is a bgp path } t \mapsto a\}$

$Sacred = \{f, t, a\} + \bigcup_{i=1}^{|B|} B_i$

$Success = \{s \in S \mid \{f, t\} \in s \mapsto a\}$

$Disconn = \{s \in S \mid \nexists \text{ a specific-prefix path } s \mapsto a\}$

$Sources \; -= Sacred \cup Success \cup Disconn$

Score poisons:

$Score = [0] * |Sources|$

foreach $s_i \in Sources$ **do**

  foreach $s_j \in s_i \mapsto a$ **do**

    $Score_j \; += 1$

  end

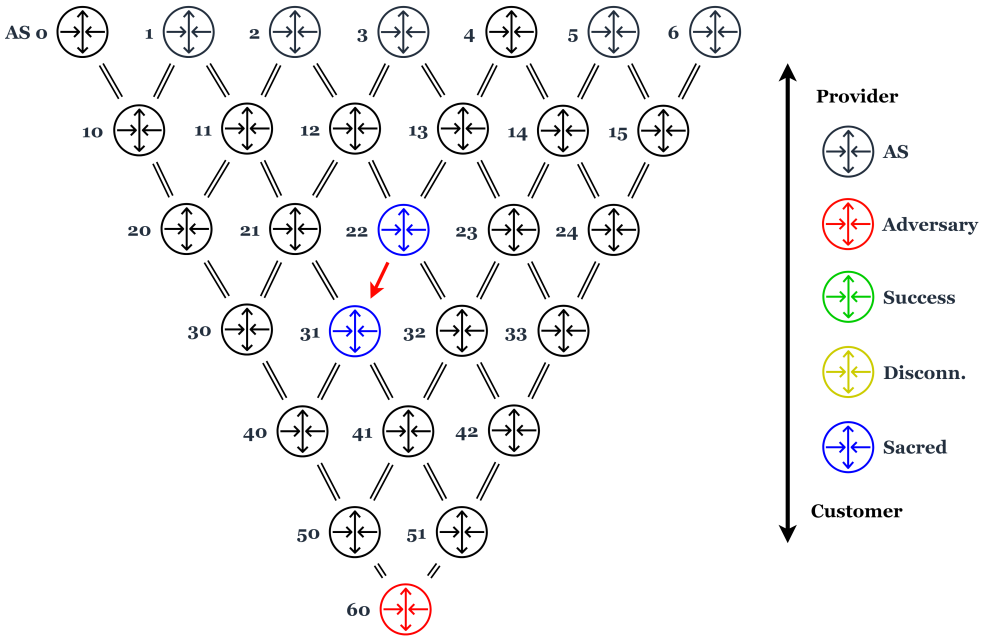
end

Poison:

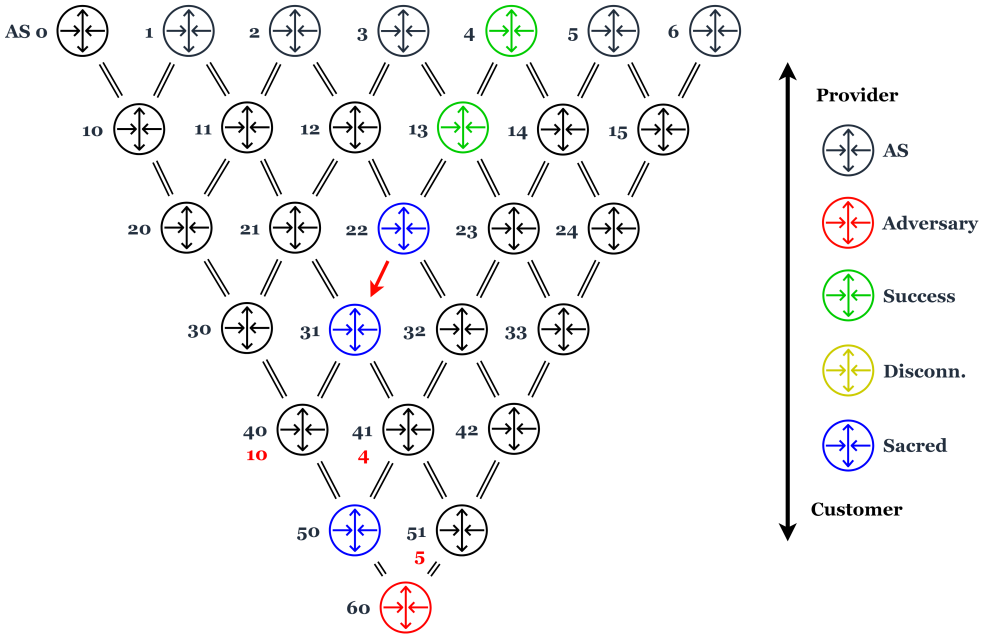
$Poisons \; += j \ni Score_j == max(Score)$

a sends advertisement to poison $Poisons$

end

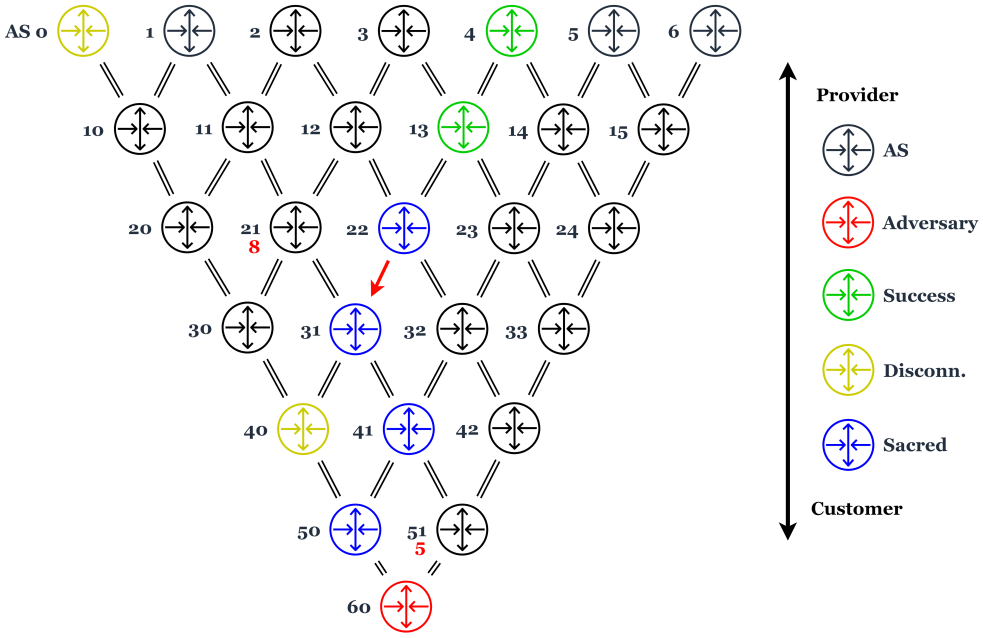


(a) Topology at start with target link 22 → 31 and adversary 60 in red.

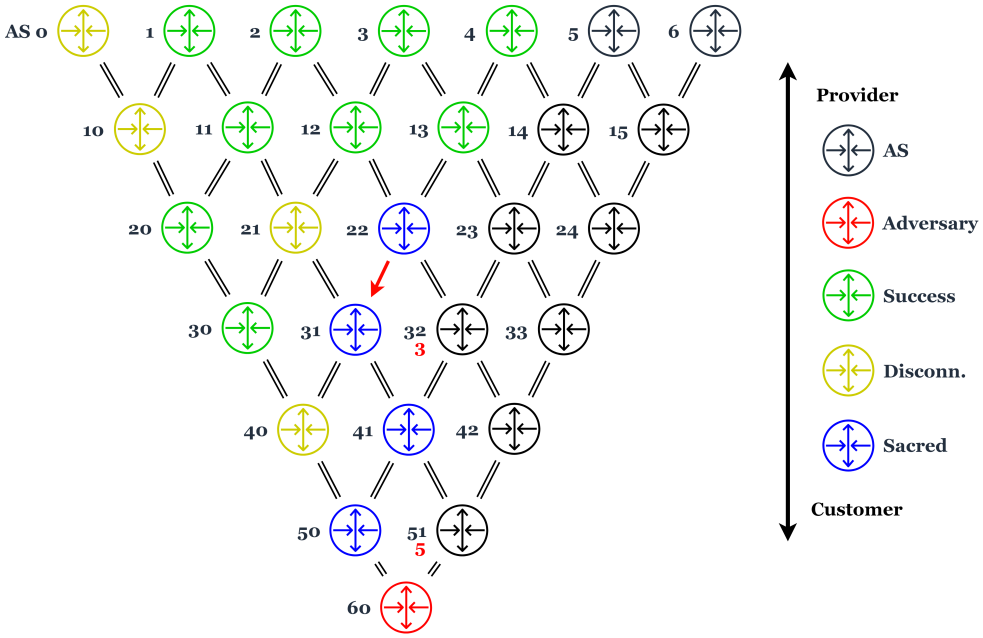


(b) ASes prefer customer routes, shorter paths, and lower ASNs. {4, 13, 22} on link at start; 50 marked sacred, 31 not advertised a path to 60 without it. Select **40** to poison; most left side ASes transit it to 60.

Figure 3.1: Example Poison Scoring for Attack (1/3)

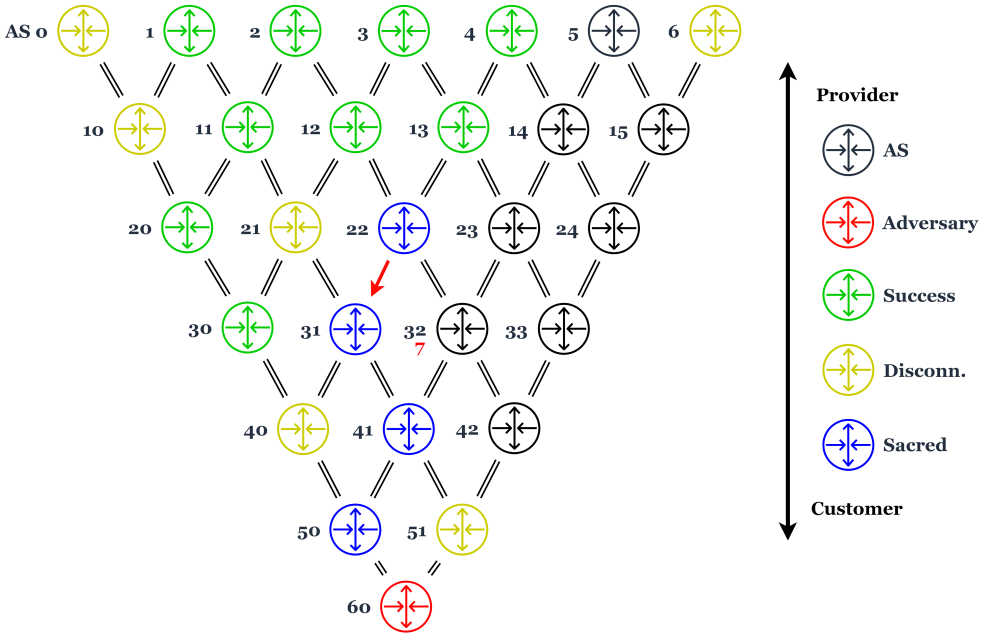


(c) This isolates 0; 0 has no valley free path to 60 without 40. No ASes move onto link after first poison. Most of top left now channeled through 21. Add **21** to poison set: **{40, 21}**.

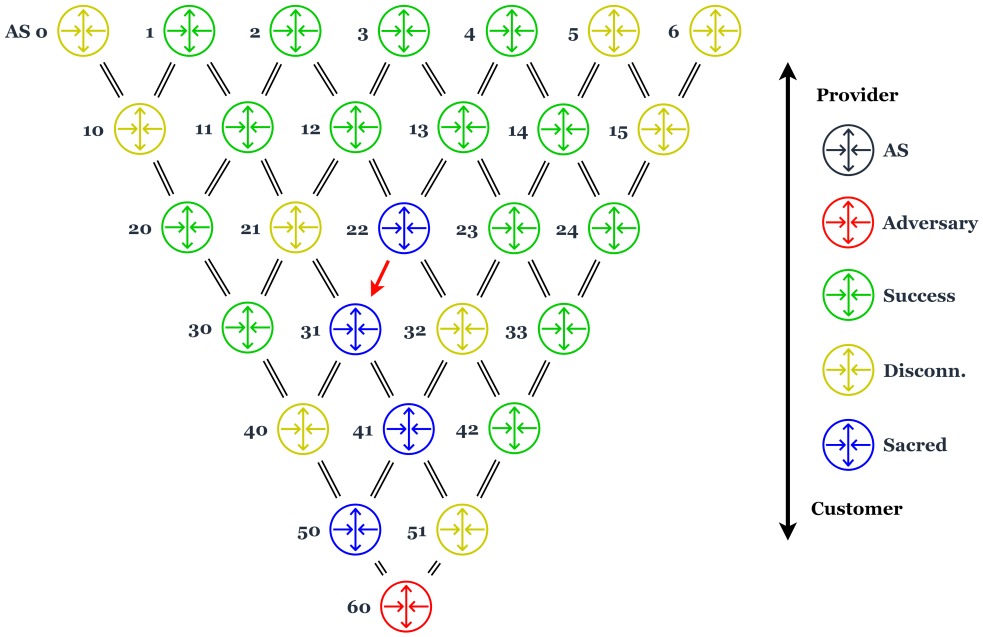


(d) Left side ASes on link or disconnected. Poison: **51**. Poison set: **{40, 21, 51}**.

Figure 3.1: Example Poison Scoring for Attack(2/3)



(e) 32 is the next poison choice. **{40, 21, 51, 32}**.



(f) We reach the termination condition: all ASes either transit link to adversary, or are disconnected/sacred. **14** ASes' traffic on target, compared to **3** initially.

Figure 3.1: Example Poison Scoring for Attack (3/3)

3.4 Simulation Details

To evaluate Maestro, we extended the Chaos BGP simulator used in previous related work [43, 40, 35] to run simulated attacks. This Internet-scale simulator builds a BGP topology based on publicly available, state-of-the-art inferred AS relationship data from CAIDA (20190201 data used) [3]. In the simulator, ASes perform a simplified BGP decision process for path selection that includes longest-prefix matching, shortest AS PATH, and simplified local policies. As true local AS policies are private, this is the most accurate simulation of AS behavior we can devise; soon-to-appear work shows that real-world AS responses to BGP poisoning generally track their simulated responses in Chaos [39].

For each attack, we use three botnet models (see Section 3.4.1) based on Mirai, Blackenergy, and Conficker botnet IP measurements. With these models, we can measure pre-attack *flow density* for a target, which we define as the percent of bot IPs with a path to another bot IP or the adversary over the link in the inferred Internet topology. This represents the present vulnerability of the link to a Coremelt-style Link Flooding Attack [41]. Next, we execute the Maestro Attack using the technique from the previous section in an attempt to bring additional bot traffic to bear on the target. Finally, we measure post-attack flow density to determine how well we steered bot-containing ASes onto the target link.

3.4.1 Botnet Models

Our botnet models are built from passive and active measurements of infected hosts from a variety of sources. The *Mirai* botnet model includes more than 2 million IP addresses. These addresses were recorded by a Chinese CDN as they attempted to spread the malware, a process with a unique signature [31]. Our *Conficker* model is based on prior work that presented a method for detecting command-and-control domain names (in essence, rendezvous points for infected hosts) for the Conficker botnet family; the IPs in the model were determined by monitoring bot traffic to those domains [42]. The *Blackenergy* model is developed from similar techniques as presented in [6].

3.5 Attack Samples

To evaluate the effectiveness of our algorithm as presented in Section 3.3.2, we chose thousands of target links to attack - and adversary ASes to attack from - in an effort to derive link vulnerability characteristics and better understand how the topological position of target and adversary affect flow density gain. The following subsections describe our sampling methods.

3.5.1 Link Selection

- **Uniform random:** Our first and most straightforward link sample set is 2000 links selected uniformly at random from all links in our inferred topology.

- **Betweenness-based:** An important insight of the Crossfire attack is that degrading links in the dense core of the Internet would create broad disruption [41]. These links are characterized by high *betweenness*, where betweenness is quantified by the number of times a link appears on paths between all ASes in the pre-attack inferred topology. So, for our second sample set, we split all links in the CAIDA AS relationship dataset [3] by their betweenness decile, and sample 100 links each from 1) below the 1st decile (fringe links), 2) between the 5th and 6th decile (moderately utilized links), and above the 9th decile (core links). This will allow us to compare the vulnerability of links to the attack based on their path usage.

- **Flow density-based:** Our third and final target link set is also sampled from low, middle, and high decile ranges, but is based on pre-attack *flow-density* rather than betweenness. For each of our three botnet models, we sample 100 links each from the low, middle, and high decile ranges. This will illustrate how effective the attack is in *both* improving the flow density for links that already have some number of bots able to direct traffic over the link *and* moving flows onto links that were previously unreachable by the botmaster. These results are presented in the appendix (see Section A).

3.5.2 Adversary Selection

We must also decide how to select the adversary ASes that will be used to issue poisoned advertisements. Intuitively, we expect that an AS's ability to steer traffic onto a selected link will dissipate with increased topological distance from the link. So, we constrain our adversary selection to ASes that are within 3 topological hops from the target, a number chosen to be less than the average BGP path length (3.5) to convey some sense of proximity [33]. To establish how distance affects attack success, we sample adversary ASes from one, two, and three hops distant from the target link.

- **General selection:** BGP relationships are another important consideration in selecting an adversary. We observe that the existence of valley-free paths from infected ASes to the adversary AS over the target link - *complete paths* - is a necessary condition for the attack to succeed. So, we constrain adversary selection to ASes that lie along a valley-free path from the To AS. Also, since path export rules are different for providers, customers, and peers (see Section 2.1), the prevalence of complete paths may be affected by relationships. To explore these dynamics, we ensure that ASes connected to the To AS via customers, peers, and providers (ASes in the customer, peer, and provider *regions*) are represented in the sampling. Figure 3.2 shows an example sampling respecting these considerations. Note that sampling for a customer-to-provider link is depicted; fewer options are available for provider-to-customer targets due to BGP path export rules. Peer links have different export rules than provider/customer links, and are not attacked in any of the experiments in this work.

- **Customer-only selection:** The customer cone of an AS has the highest possible visibility of routes exported from the AS; naturally, the AS seeks to provide its customers with all of its known best paths in hopes of transiting customer traffic to the maximum number of destinations. In some scenarios, we further limit adversary selection by sampling only from the To AS customer cone. We expect that these ASes will have the maximum number of *complete paths* among possible adversaries. For a depiction of this type of sampling, see Figure 3.3. Note that this selection type produces a subset of the adversaries selected by the other method.

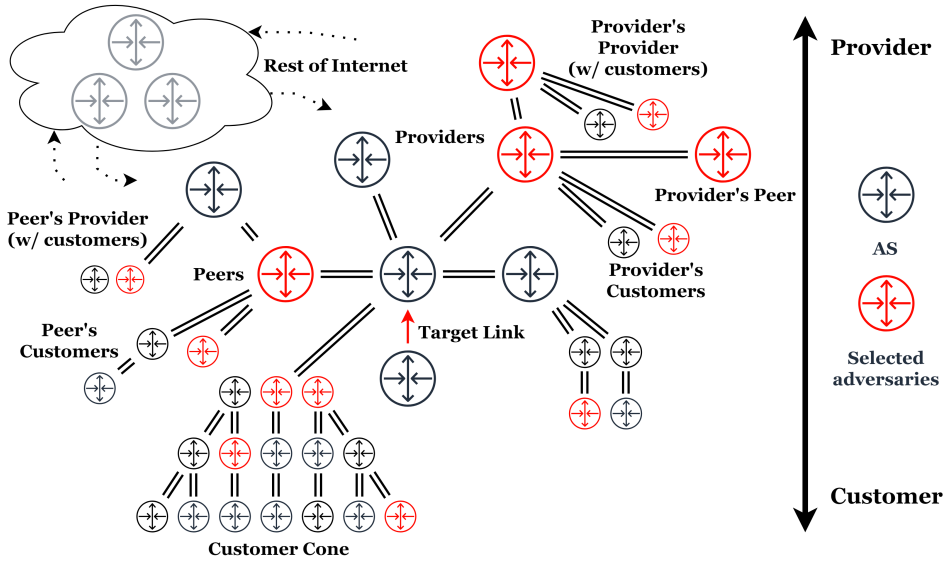


Figure 3.2: Sampling adversary ASes at random along valley free paths from the To AS, within 3 topological hops, with adversaries reached from To AS's peers, customers, and providers included.

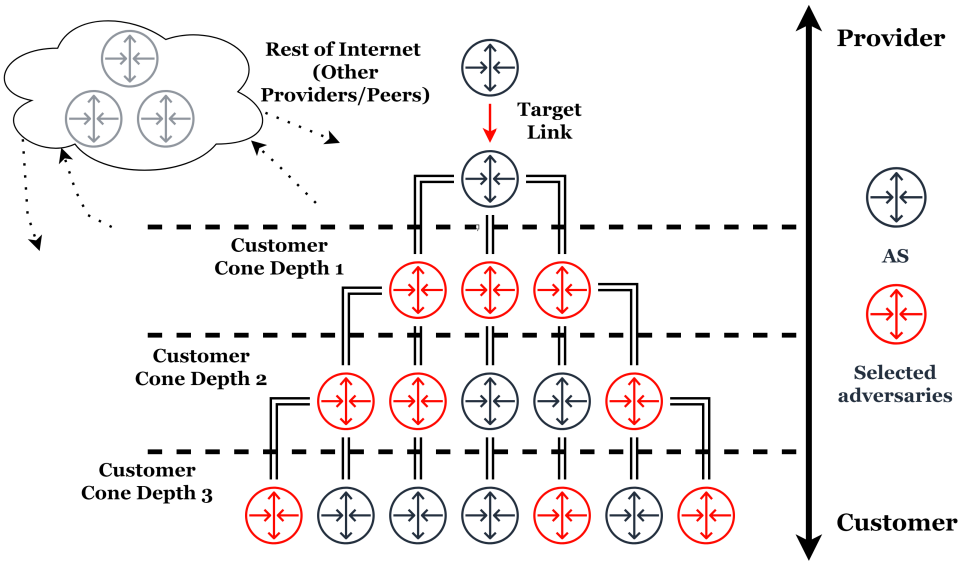


Figure 3.3: Sampling adversary ASes at random from multiple depths (1-3) into the To AS customer cone.

We present results for several attack scenarios in Section 4. These scenarios combine the above listed methods for selecting target links and adversary ASes (with the exception of the last scenario; this is a special case). The results for each experiment serve to highlight the roles that target selection and BGP positioning play in determining attack success.

Chapter 4

Evaluation

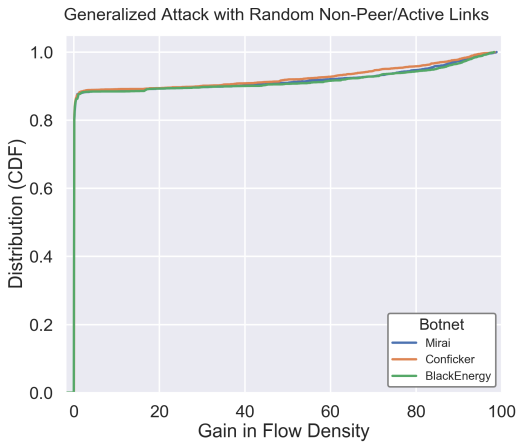
We evaluate our attack for various link/adversary selection schemes in the following subsections. Our success metric, in general, is *flow density*. Recall that flow density is simply the percentage of bot IPs in the botnet that have a path to another bot (or the adversary) over the target link. In most graphs, we present *flow density gain*, calculated via *post-attack flow density - pre-attack flow density*, while in others we plot both pre- and post-attack flow density for comparison. Link and adversary selection schemes for each scenario are summarized in Table 4.1.

4.1 Random Link Scenario

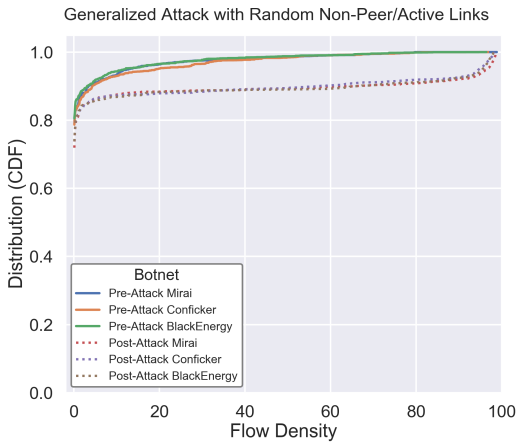
Our first scenario consists of 2000 links selected uniformly at random from the topology. The only links excluded from selection are 1) links with zero simulated betweenness, 2) last-mile links to edge ASes (those with no customers) that can be hit via traditional DDoS rather than an LFA, and 3) peer links, which are governed by different export rules as discussed in

Table 4.1: Experiments Presented

Experiment	Link Selection	Adversary Selection
Random	Uniform random	Generalized
Customer Cone	Betweenness-based	Customer cone
Generalized Position	Betweenness-based	Generalized
Infected Cone	Custom (highly infected from AS cust cone)	Generalized



(a) Flow density gain distribution for targets, 3 botnets, random link selection.



(b) Pre vs post attack flow density, 3 botnets, random link selection

Figure 4.1: Random link attack results.

Section 2.1. Adversarial selection for this scenario is performed as described in 3.5.2 under *General selection*. Three adversaries are sampled at each depth 1-3 from each region (ASes reached from customers, providers, and peers of the To AS), for a total of about 27 attackers per link. Note that there will be fewer attackers when the To AS has a limited number of customers/peers to sample from. The results are shown in Figure 4.1.

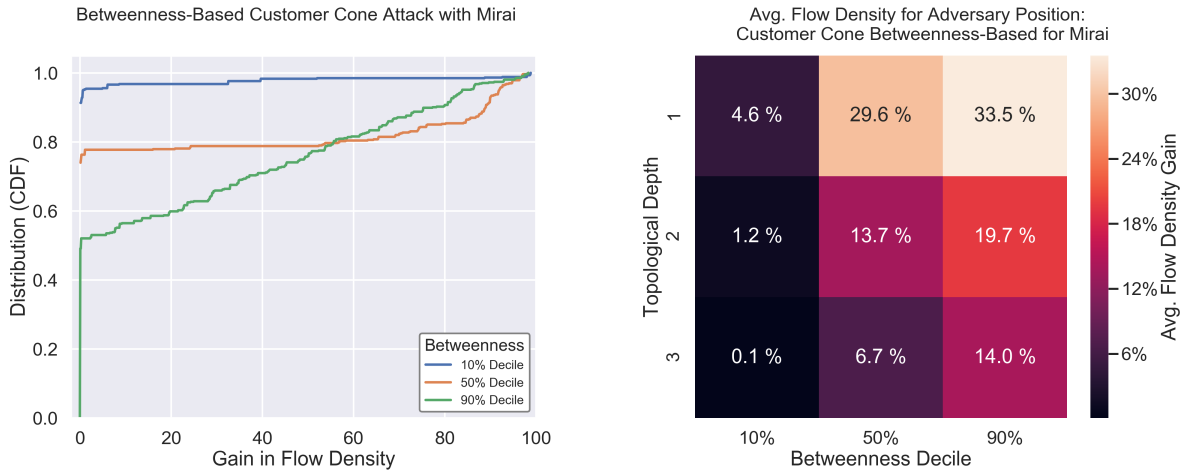
We make two observations about this initial experiment: first, that results from each of our three botnet models, despite having different distributions of infected hosts in the topology, generally exhibit similar steering behavior. This dynamic is consistent across all of our experiments, so we will henceforth only display results for the Mirai model. Graphs for the other botnets will be included in the appendix (see Section B) for completeness.

Second, we see that these results are frankly underwhelming. For more than 80 percent of sampled targets, *no* improvement was seen in flow density after the attack, nor were there specific cases where the attack resulted in dramatic improvement. An analysis of the few successful cases, however, revealed some important common factors. Successful adversaries were almost always *close* to the target link (confirming our suspicion that this was likely to play a major role in moving traffic) and, interestingly, were often located in the *customer cone* of the target link destination (the To AS). An AS's customer cone consists of all ASes

an AS can reach along customer links from itself [28]; in short, direct or indirect customers of the AS. To further explore these trends, we narrow our adversary selection in the next scenario to ASes in the To AS's customer cone.

4.2 Customer Cone Scenario

This scenario examines our most privileged adversary from a positioning standpoint. In this section, we will present results for betweenness-based link selection and customer cone adversary selection as described in Section 3.5, with three adversaries sampled from each depth in the To AS customer cone. The results are shown in Figure 4.2. The adversary's expected success in this case is dramatically improved; for direct customers of high betweenness links, the average flow density gain is *greater than 30 percent* (Figure 4.2b). Note that this figure is not percent gain relative to existing flow density - rather, an additional 30 percent *of the botnet* has been directed on to the link.



(a) Flow density gain distribution for links by betweenness decile group, customer cone attack, Mirai botnet.

(b) Heatmap of average flow density gain by adversary's topological distance from target, customer cone attack, Mirai botnet.

Figure 4.2: Customer cone attack results.

For the Mirai botnet model, 30 percent flow density gain means an additional *1 million* infected hosts, on average, directing flows over the target link. Even adversary ASes located deeper in the customer cone of the To AS can expect significant flow density improvements.

For low betweenness links, attack impact is negligible, but this is neither surprising nor particularly interesting; these links are not primary targets for LFAs. Moderate betweenness link vulnerability is generally low, but under certain conditions can be affected by the attack. We plan to examine these specific cases in greater detail in future work. The pre- and post-attack vulnerability in absolute terms of high betweenness (core) links is illustrated in Figure 4.3. Note that the region between the curves in this figure represents the attacker's gain from executing the Maestro attack.

4.3 Generalized Position Results

In these experiments, we combine the betwenness-based link selection from Section 4.2 with the more general adversary selection from Section 4.1.

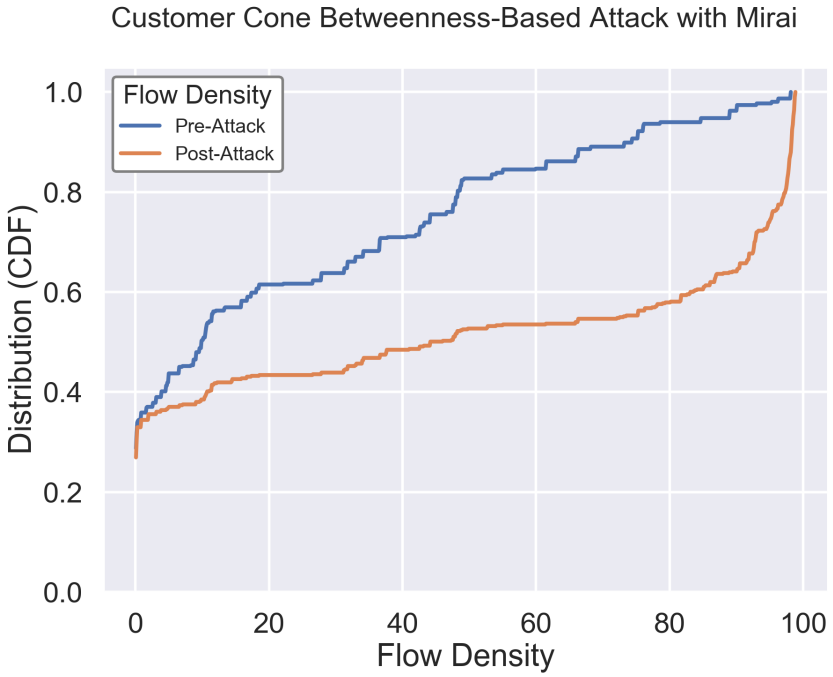
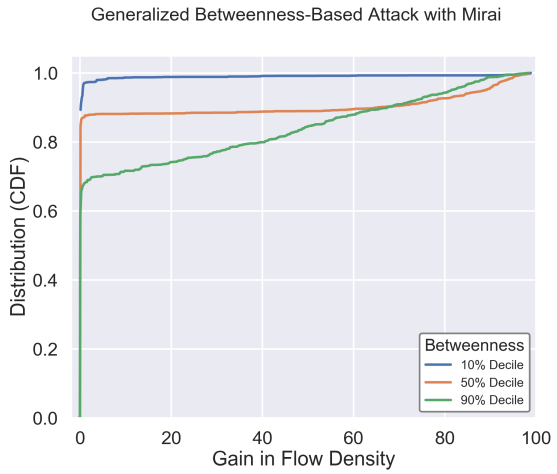


Figure 4.3: Flow density pre vs post attack CDF for links above highest betweenness decile, customer cone attack, Mirai botnet.

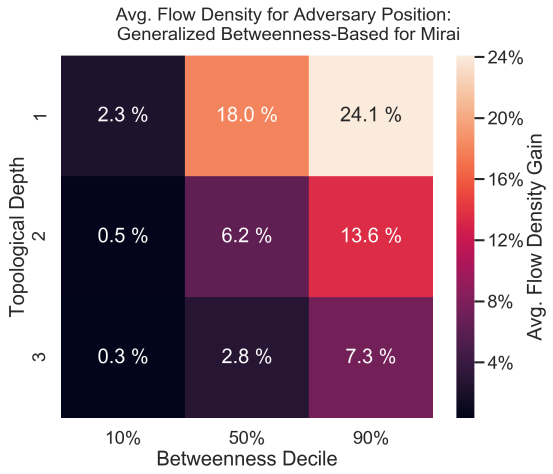
Our motivation is to discover new successful scenarios with similar effect to those in Section 4.2 where core (high betweeness) links can be significantly impacted with adversaries positioned *outside* the To AS customer cone. Figure 3.2 displays our results.

Our conditions for success appear to follow the same pattern as those in the preceding section, with low and moderate betweenness links mostly outside the range of the attack. For links above the highest betweenness decile, we had a modest number of successful cases. However, we examined the best (most improved flow density) cases in this experiment more closely, and found that nearly all were produced by provider-to-customer target links as shown in Figure 4.5.

Recall that the general adversary selection scheme used in this scenario (see Section 3.5.2) is restricted to selecting adversaries within the To AS customer cone for provider-to-customer links. This is because complete paths (paths from infected hosts to the adversary over the link) must be valley-free, and transiting to a customer over the target link means that such paths only exist for adversaries in to the To AS customer cone.



(a) Flow density gain CDF for links by betweenness decile group, generalized attack, Mirai botnet.



(b) Heatmap of average flow density gain by adversary's topological distance from target, generalized attack, Mirai botnet.

Figure 4.4: Generalized attack results.

Clearly, though, attacking from the customer cone is not nearly as effective for customer-to-provider links; every link in this experiment included adversaries sampled from the To AS customer cone, to little effect as shown in Figure 4.5.

Still, the customer-to-provider link direction was successfully attacked in some cases. To complete this set of experiments, we investigate the conditions under which those attacks can significantly enhance flow density.

4.4 Infected Cone Results

As discussed in Section 2.1, we expect ASes to export routes learned from customers to all their neighbors as a result of economic incentive. For customer-to-provider links, consider that any bots that must transit a customer link to reach the target link *should not* then transit the target link to the adversary; doing so would be a violation of valley-free routing.

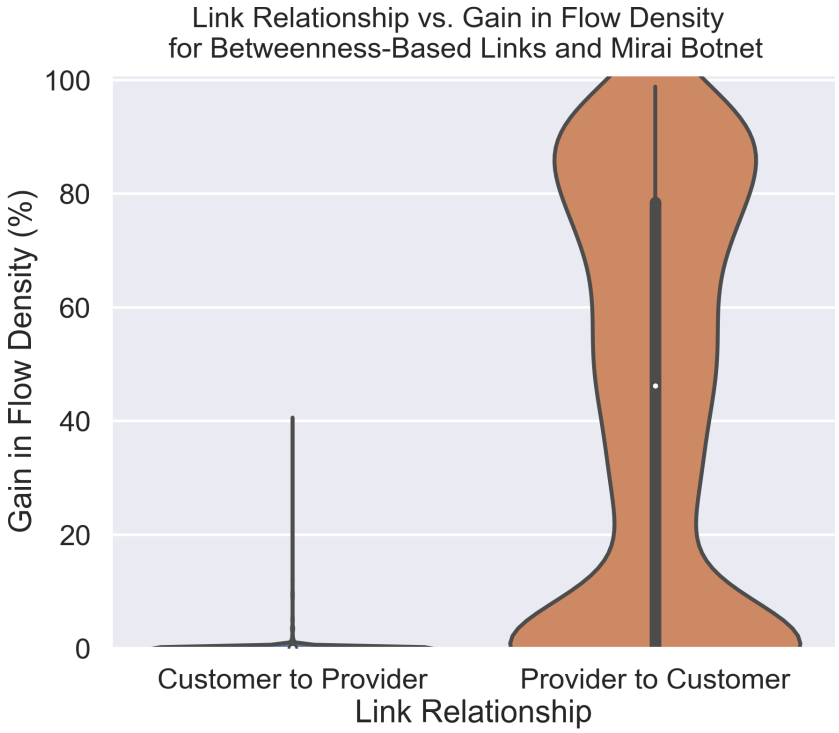


Figure 4.5: Achieved flow density distribution by link relationship, generalized attack, Mirai botnet. Violin width at a given y-value (density) indicates proportion of attacks with that density. Note that violin widths cannot be compared across violins.

It follows, then, that bots must have paths that *only* transit customer-to-provider links to reach the target link and still be able to transit it to the adversary. But this means that the From AS must be able to reach these bots by transiting only provider-to-customer links; the potential flow sources are, by definition, in the From AS customer cone.

Analyzing the few relatively successful attacks on customer-to-provider links in the previous section indeed showed that the From AS in these cases always had an above-average numbers of bots in their customer cones. To confirm that this is indeed the most important factor in attacking customer-to-provider links with the Maestro attack, we randomly sampled 300 customer-to-provider target links from the set of target links whose From ASes were above the 9th decile in total bot IP count in their customer cones. We then sampled adversaries as in the previous section (generalized selection) and simulated attacks, producing the results in Figure 4.6. Here we limit our definition of flow density to the portion of the botnet present in the From AS customer cone, as these are the only infected networks we can steer.

Our experimental results confirmed our reasoning in this case; we can often steer this subset of the botnet to significant effect with Maestro. This presents an interesting additional set of success conditions for the attack, as some of the largest ASes have customer cones that cover nearly half the Internet [2].

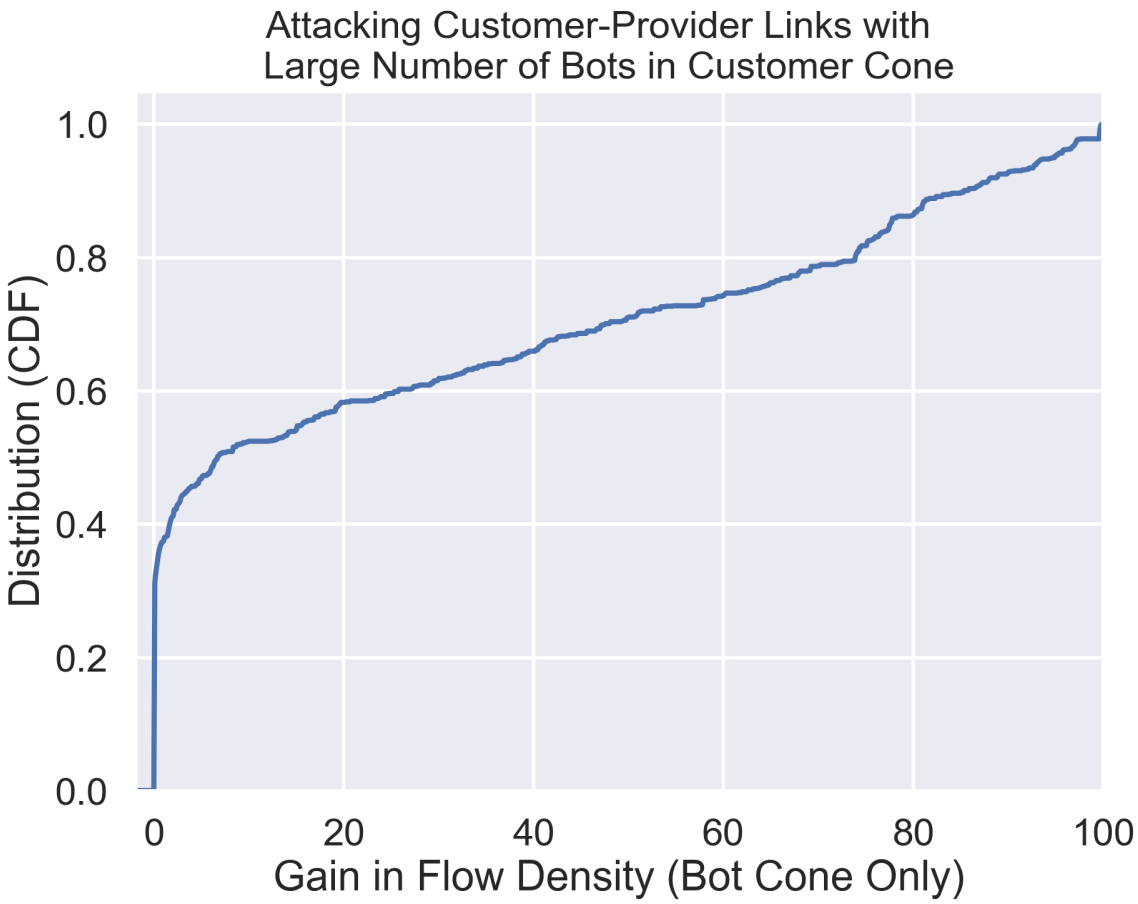


Figure 4.6: Distribution of gain in flow density (as pct of AS customer cone bots) over target links.

Chapter 5

Related Work

The Coremelt [41] and Crossfire [20] attacks are discussed in detail in Section 2.3.1. Classifying links by BGP betweenness is a technique employed by in [36]. Interestingly, the attack from that work used the control-plane to attack the data-plane; here, we leverage the control-plane to augment a data-plane attack. LFA mitigation work that applies to this attack is presented in Section 6.2.

Nyx [40] and LIFEGUARD [22] employ BGP poisoning for DDoS and link failure mitigation, respectively. In [8], poisoning is used as a technique for route discovery, and the propagation of poisons throughout the Internet is actively measured in [39].

Chapter 6

Conclusion

6.1 Summary of Findings

We present a number of key takeaways from analysis of the experiments presented in Sections 1.1 and 3.

- **LFAs cannot target arbitrary links with full force in practice.** Section 1.1 demonstrates that bot-to-bot (and less restrictive) LFA models are incapable of striking any link on the Internet. In fact, even many core links can be reached by just a fraction of infected hosts in all three botnet models.

- **The Maestro attack can partially overcome these limitations.** As demonstrated in Section 4, about half of the highest betweenness (core) links in our sample set had pre-attack Mirai flow densities of 15 percent or less; after attack execution from the To AS customer cone, most links had 40 percent or greater achieved flow density.

- **High betweenness links are much more vulnerable to this attack.** All results in Section 4 clearly indicate that highly used links are much more vulnerable to Maestro attackers. This is an intuitive finding, but an important one, as it demonstrates that the most critical targets can be reached by the Maestro attack.

- **The most advantageous position for the Maestro attack adversary AS is within the customer cone of the target link destination (To AS).** The results

in Section 4.2 bear this point out. A direct customer of the To AS can expect to move fully a third of the Mirai botnet onto the target link with this attack. Importantly, *AS rank plays little role in determining success.* The Pearson correlation coefficient for flow density gain as a function of AS rank [2] is less than 0.01 for the customer cone attack.

- **Provider-to-customer targets are far more vulnerable to a Maestro attack.** Figure 4.5 shows how stark the differences are between provider-to-customer and customer-to-provider targets.

- **Customer-to-provider targets are vulnerable when link source endpoint (From ASes) has significantly infected customer cone.** While customer-to-provider links are not vulnerable in general, this is an important exception. Future work will explore how this result relates to peer links, which commonly link large transit ASes.

6.2 Mitigation

There are two broad categories of mitigations that apply to our attack. The first are general Link Flooding Attack solutions. Unfortunately, state-of-the-art countermeasures are not widely available to network operators because they require either collaboration between ASes not properly incentivized to collaborate [4, 24] or infrastructure capabilities not deployed in practice [19, 26]. The solution presented in [40] could partially mitigate the attack by moving traffic from a critical AS to the deploying AS off the attacked link, but the link would still be affected for all other source/destination AS pairings utilizing the link.

The second and more relevant category of mitigations target the poisoned route advertisements that serve as the primary primitive for Maestro attacks. Route Origin Authorization would not affect this attack, as the compromised AS owns the advertised destination prefixes [25]. BGPSEC, if widely deployed, could prevent this kind of AS PATH tampering; unfortunately, it is not deployed at scale [13].

Detecting or filtering advertisements by individual network operators is the most straightforward approach to countering a Maestro attack. However, some proposed DDoS mitigation [40] and link failure response [22] systems rely on BGP poisoning, and network operators have used it to control path propagation for traffic engineering [34]. Filtering all BGP poisons, then, may have some cost, and finer-grained responses (including careful monitoring of downstream advertisements) may be more appropriate.

6.3 Future Work

A number of avenues for future work exists for this attack. The core algorithm, poison scoring, is a simple first technique with many opportunities for improvement. The current version weighs all ASes equally when making poisoning decisions; infected hosts, on the other hand, are not uniformly distributed throughout the Internet topology. Some method of weighing poisoning decisions could therefore enhance the performance of the algorithm.

We currently limit our adversary AS to a single poisoned prefix when making advertisements; in practice, a compromised AS may have many prefixes to choose from, meaning that source ASes with conflicting poison sets could be assigned to different prefixes when making advertisements. Alternatively, after the initial simple algorithm generates an approximation to the large poisoning problem, an optimal solution for the smaller remaining disconnected AS set could be found via a MAX SAT solver. Finally, with multiple prefixes, the Maestro attack could be trivially extended to attacking multiple links for a more sophisticated, Crossfire-style isolating LFA [20].

Bibliography

[1] (2016). Dyn analysis summary of friday october 21 attack. https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/. 12

[2] (2019a). CAIDA AS rank dataset. http://as-rank.caida.org/. 34, 38

[3] (2019b). CAIDA AS relationship dataset. http://www.caida.org/data/active/as-relationships/index.xml. ix, 4, 5, 17, 23, 24

[4] Basescu, C., Reischuk, R. M., Szalachowski, P., Perrig, A., Zhang, Y., Hsiao, H.-C., Kubota, A., and Urakawa, J. (2016). SIBRA: Scalable internet bandwidth reservation architecture. In *Proceedings of Symposium on Network and Distributed System Security (NDSS)*. 38

[5] Case, D. U. (2016). Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*. 12

[6] Chang, W., Mohaisen, A., Wang, A., and Chen, S. (2015). Measuring botnets in the wild: Some new trends. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 645–650. ACM. 23

[7] Cohen, D., Cooper, M., and Jeavons, P. (2004). A complete characterization of complexity for boolean constraint optimization problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 212–226. Springer. 17

[8] Colitti, L., Di, D., et al. (2006). Internet topology discovery using active probing. 36

[9] Dainotti, A., Squarcella, C., Aben, E., Claffy, K. C., Chiesa, M., Russo, M., and Pescapé, A. (2011). Analysis of country-wide internet outages caused by censorship. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 1–18. ACM. 16

[10] Demchak, C. C. and Shavitt, Y. (2018). Chinas maxim–leave no access point unexploited: The hidden story of china telecoms bgp hijacking. *Military Cyber Affairs*, 3(1):7. 16

[11] Donnet, B. and Bonaventure, O. (2008). On bgp communities. *ACM SIGCOMM Computer Communication Review*, 38(2):55–59. 10

[12] Gao, L. (2001). On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking (ToN)*, 9(6):733–745. 9

[13] Goldberg, S. (2014). Why is it taking so long to secure internet routing? *Commun. ACM*, 57(10):56–63. 38

[14] Google Security and White Ops (2016). The hunt for 3ve. 1, 16

[15] Hawkinson, J. and Bates, T. (1996). Rfc 1930 - guidelines for creation, selection, and registration of an autonomous system. https://tools.ietf.org/html/rfc1930. 8

[16] Herzog, S. (2011). Revisiting the estonian cyber attacks: Digital threats and multinational responses. *Journal of Strategic Security*, 4(2):49–60. 12

[17] Jonker, M., Sperotto, A., van Rijswijk-Deij, R., Sadre, R., and Pras, A. (2016). Measuring the adoption of ddos protection services. In *Proceedings of the 2016 Internet Measurement Conference*, pages 279–285. ACM. 2

[18] Kalish, D., Montague, R., and Mar, G. (1964). Logic: techniques of formal reasoning. 17

[19] Kang, M. S., Gligor, V. D., and Sekar, V. (2016). Spiffy: Inducing cost-detectability tradeoffs for persistent link-flooding attacks. In *NDSS*. 38

[20] Kang, M. S., Lee, S. B., and Gligor, V. D. (2013). The crossfire attack. In *2013 IEEE Symposium on Security and Privacy*, pages 127–141. IEEE. 2, 4, 13, 36, 39

[21] Kaspersky Lab (2019). Ddos attacks in q4 2018. ix, 2

[22] Katz-Bassett, E., Scott, C., Choffnes, D. R., Cunha, Í., Valancius, V., Feamster, N., Madhyastha, H. V., Anderson, T., and Krishnamurthy, A. (2012). Lifeguard: Practical repair of persistent route failures. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 395–406. ACM. 36, 39

[23] Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84. 2, 12

[24] Lee, S. B., Kang, M. S., and Gligor, V. D. (2013). Codef: collaborative defense against large-scale link-flooding attacks. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pages 417–428. ACM. 38

[25] Lepinski, M., Kent, S., and Kong, D. (2012). Rfc 6482: A profile for route origin authorizations (roas). *Internet Engineering Task Force (IETF)*, 201(2). 38

[26] Liaskos, C., Kotronis, V., and Dimitropoulos, X. (2016). A novel framework for modeling and mitigating distributed link flooding attacks. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE. 38

[27] Litke, P. and Stewart, J. (2014). Bgp hijacking for cryptocurrency profit. 1, 16

[28] Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., et al. (2013). As relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 243–256. ACM. 30

[29] Majkowski, M. (2018). Memcrashed-major amplification attacks from udp port 11211. Technical report, Technical Report. Cloudflare. 2

[30] McPherson, D. and Gill, V. (2006). Bgp multi_exit_disc (med) considerations. Technical report. 10

[31] Netlab360 (2017). Mirai Scanner. http://data.netlab.360.com/mirai-scanner/. 23

[32] Putman, C., Nieuwenhuis, L. J., et al. (2018). Business model of a botnet. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 441–445. IEEE. 16

[33] RIPE (2012). Update on as path lengths over time. https://labs.ripe.net/Members/mirjam/update-on-as-path-lengths-over-time. 25

[34] Roughan, M., Willinger, W., Maennel, O., Perouli, D., and Bush, R. (2011). 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821. 39

[35] Schuchard, M., Geddes, J., Thompson, C., and Hopper, N. (2012). Routing around decoys. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 85–96, New York, NY, USA. ACM. 23

[36] Schuchard, M., Mohaisen, A., Foo Kune, D., Hopper, N., Kim, Y., and Vasserman, E. Y. (2010). Losing control of the internet: using the data plane to attack the control plane. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 726–728. ACM. 4, 36

[37] Scott Sr, J. and Summit, W. (2016). Rise of the machines: The dyn attack was just a practice run december 2016. 2

[38] Shin, S., Gu, G., Reddy, N., and Lee, C. P. (2012). A large-scale empirical study of conficker. *IEEE Transactions on Information Forensics and Security*, 7(2):676–690. 12

[39] Smith, J. M., Birkeland, K., and Schuchard, M. (2018). An internet-scale feasibility study of bgp poisoning as a security primitive. *arXiv preprint arXiv:1811.03716*. 23, 36

[40] Smith, J. M. and Schuchard, M. (2018). Routing around congestion: Defeating ddos attacks and adverse network conditions via reactive bgp routing. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 599–617. IEEE. 10, 18, 23, 36, 38, 39

[41] Studer, A. and Perrig, A. (2009). The coremelt attack. In *European Symposium on Research in Computer Security*, pages 37–52. Springer. 2, 3, 4, 13, 23, 24, 36

[42] Thomas, M. and Mohaisen, A. (2014). Kindred domains: Detecting and clustering botnet domains using dns traffic. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 707–712, New York, NY, USA. ACM. 23

[43] Tran, M., Kang, M. S., Hsiao, H.-C., Chiang, W.-H., Tung, S.-P., and Wang, Y.-S. (2019). On the feasibility of rerouting-based ddos defenses. In *To appear in Proceedings of IEEE Symposium on Security and Privacy (IEEE S&P)*. 4, 18, 23

[44] Vazirani, V. V. (2013). *Approximation algorithms*. Springer Science & Business Media. 17

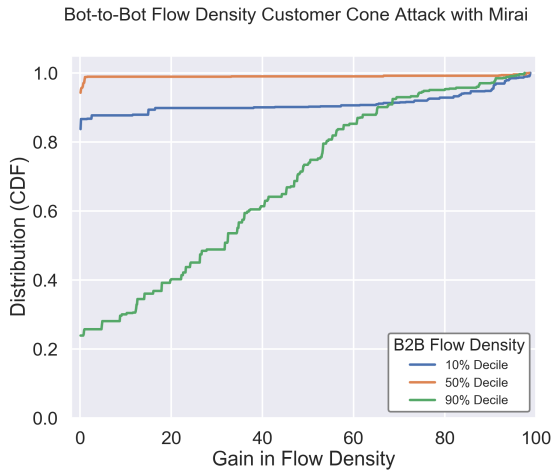
[45] White Ops (2016). The methbot operation. 1

Appendix

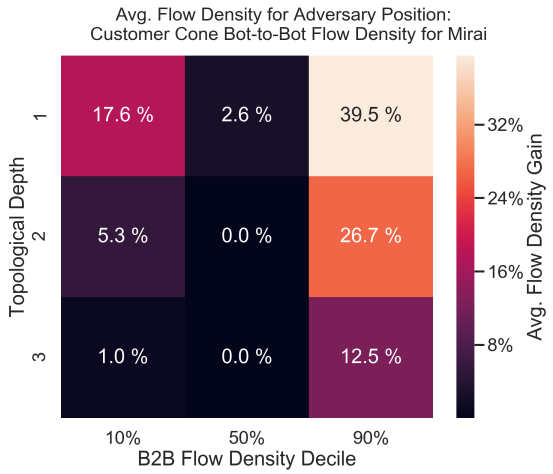
A Results for Flow Density-Based Link Sampling

In this section we present results that mirror Scenario 2 and 3 in Section 4 in design and execution, but where links are sampled by *flow density* decile rather than betweenness decile.

A.1 Customer Cone



(a) Flow density gain CDF for links by flow density decile group, customer cone attack, Mirai botnet.



(b) Heatmap of average flow density gain by adversary's topological distance from target, customer cone attack, Mirai botnet.

Figure A.1: Customer cone attack results, flow density decile sampling.

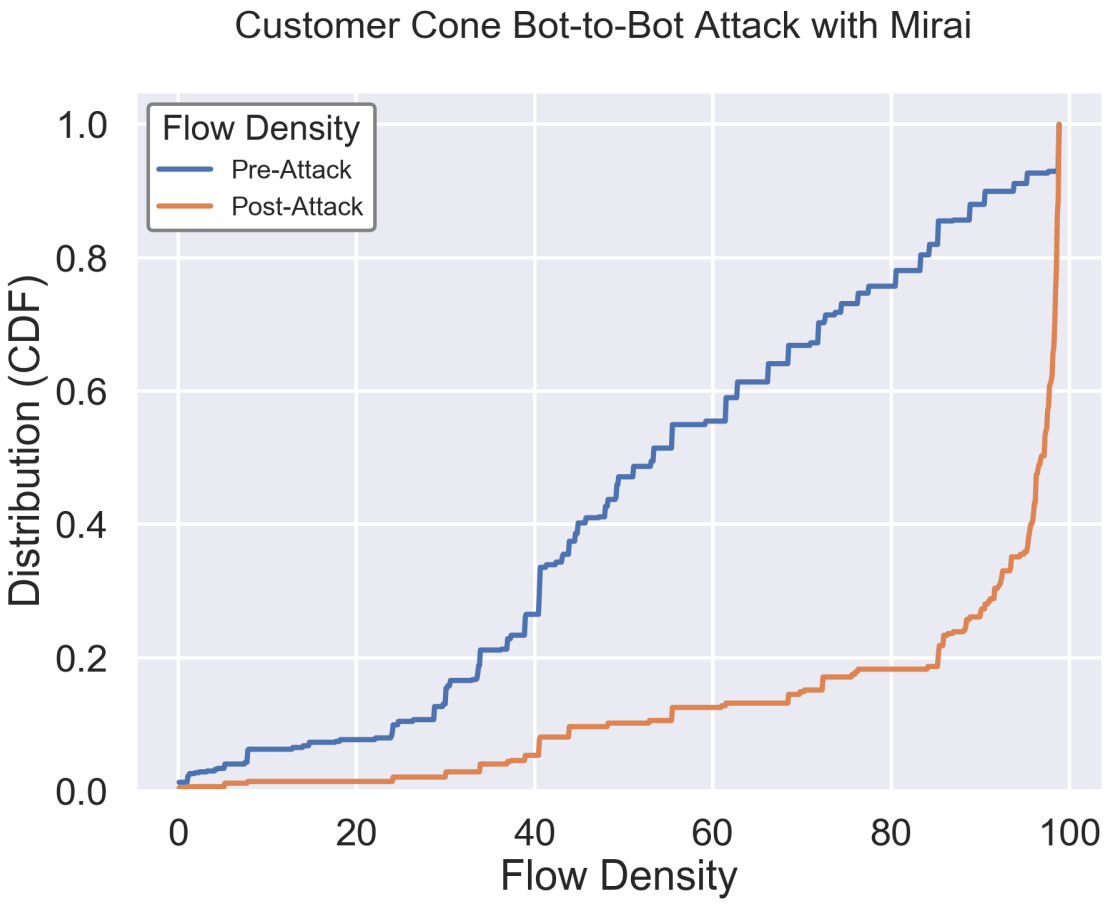
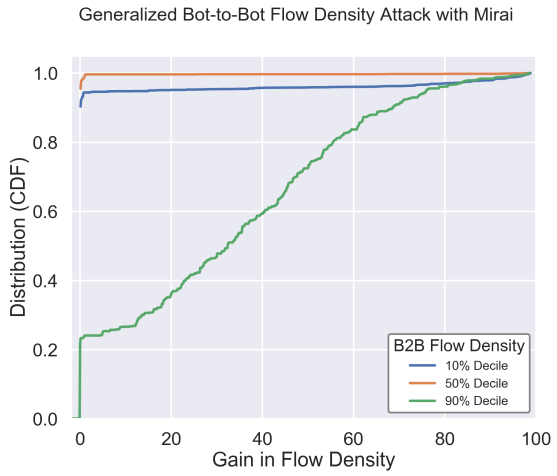
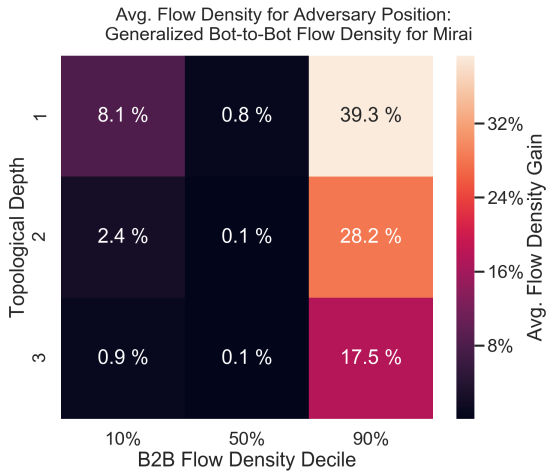


Figure A.2: Flow density pre vs post attack CDF for links above highest flow density decile, customer cone attack, Mirai botnet.

A.2 Generalized



(a) Flow density gain CDF for links by flow density decile group, generalized attack, Mirai botnet.



(b) Heatmap of average flow density gain by adversary's topological distance from target, generalized attack, Mirai botnet.

Figure A.3: Generalized attack results.

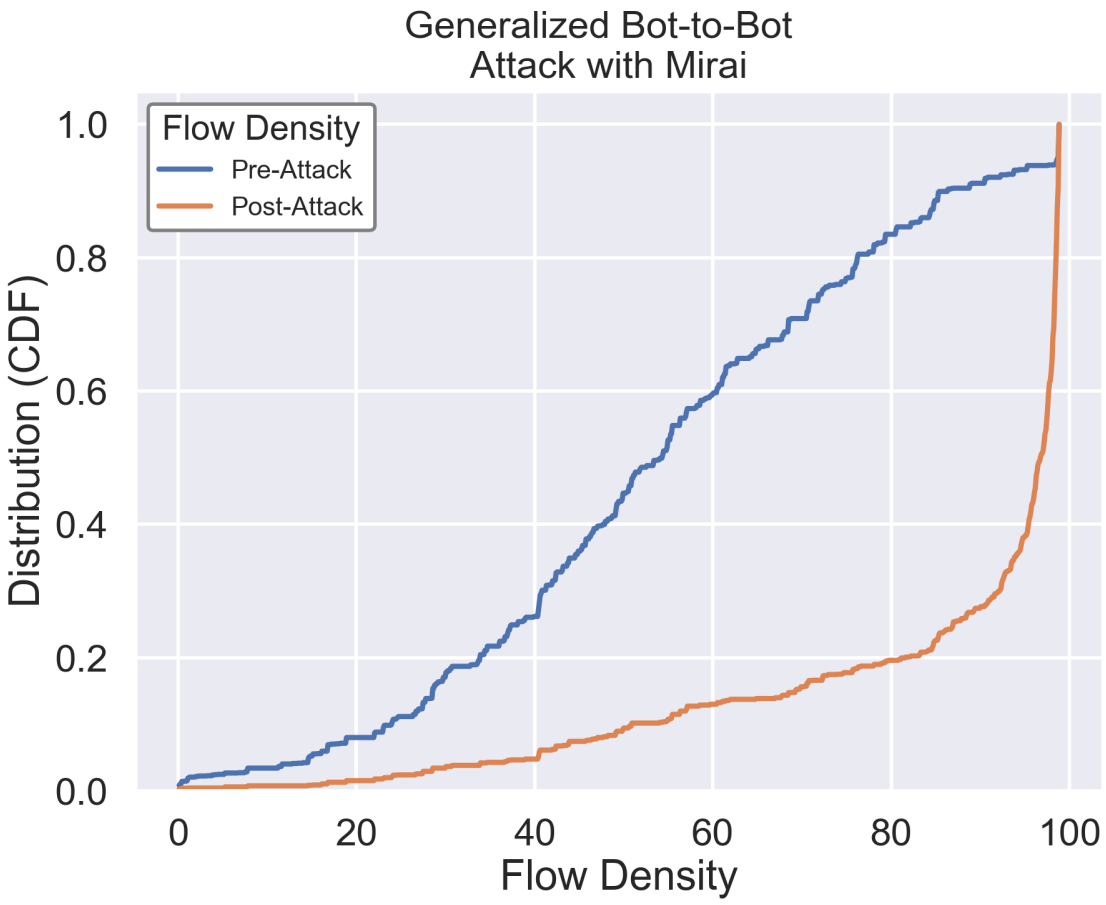


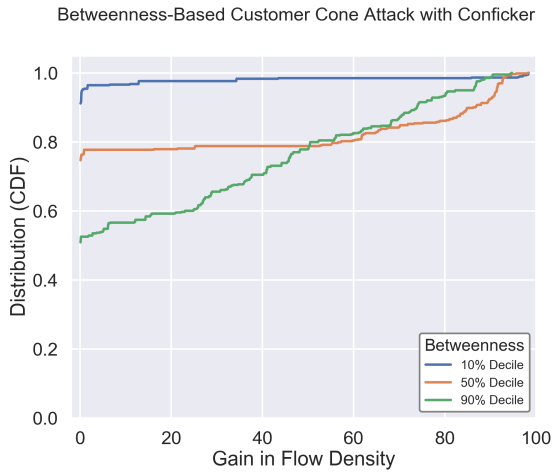
Figure A.4: Flow density pre vs post attack CDF for links above highest flow density decile, generalized attack, Mirai botnet.

B Results for Other Botnet Models

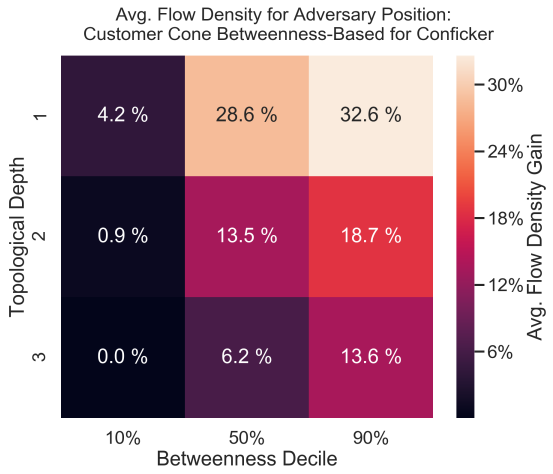
Virtually all of the results presented in Section 4 were for attacks executed with the Mirai botnet model. In general, data from attacks utilizing our other two botnet models - Conficker and Blackenergy - exhibited the same patterns as those found in the Mirai results. Here we present betweenness-based sampling results for those models for completeness.

B.1 Conficker

Customer Cone



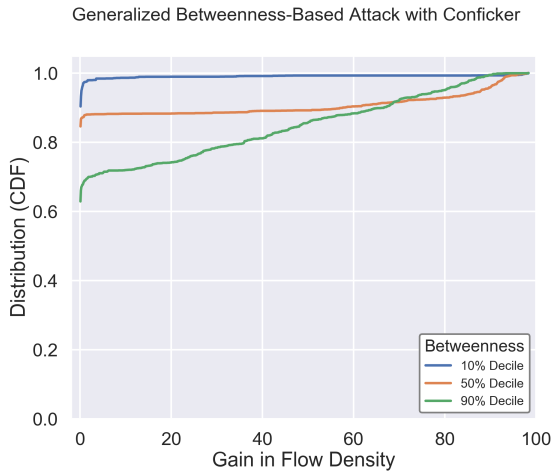
(a) Flow density gain CDF for links by betweenness decile group, customer cone attack, Conficker botnet.



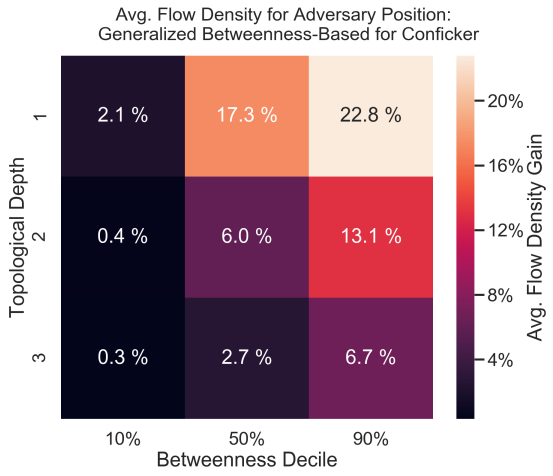
(b) Heatmap of average flow density gain by adversary's topological distance from target, customer cone attack, Conficker botnet.

Figure B.1: Customer cone attack results, betwenness decile sampling (Conficker).

Generalized



(a) Flow density gain CDF for links by betweenness decile group, generalized attack, Conficker botnet.

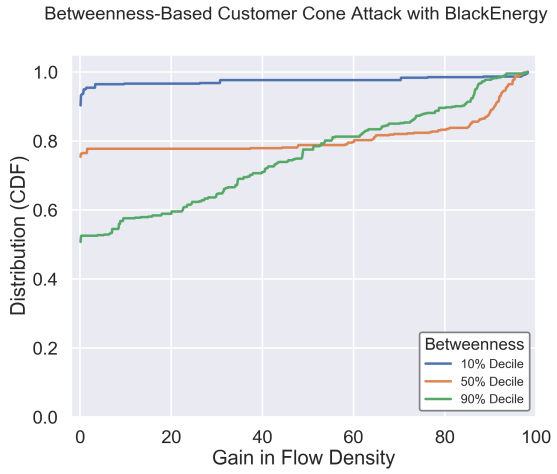


(b) Heatmap of average flow density gain by adversary's topological distance from target, generalized attack, Mirai botnet.

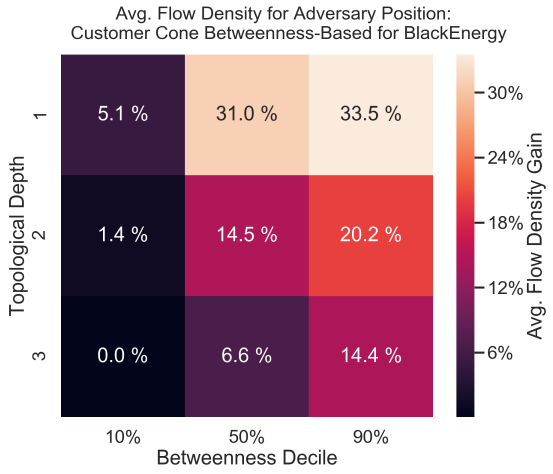
Figure B.2: Generalized attack results (Conficker).

B.2 Blackenergy

Customer Cone



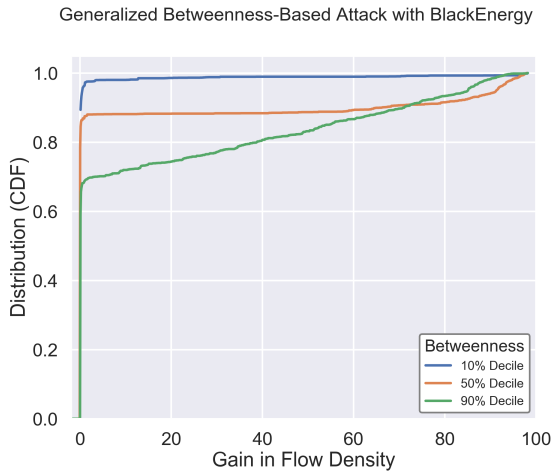
(a) Flow density gain CDF for links by betweenness decile group, customer cone attack, Blackenergy botnet.



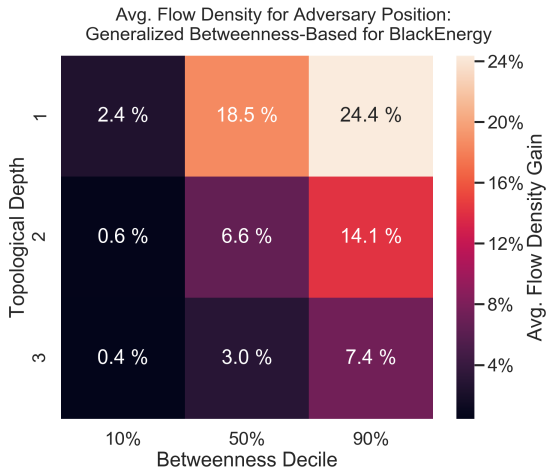
(b) Heatmap of average flow density gain by adversary's topological distance from target, customer cone attack, Blackenergy botnet.

Figure B.3: Customer cone attack results, betwenness decile sampling (Blackenergy).

Generalized



(a) Flow density gain CDF for links by betweenness decile group, generalized attack, Blackenergy botnet.



(b) Heatmap of average flow density gain by adversary's topological distance from target, generalized attack, Blackenergy botnet.

Figure B.4: Generalized attack results (Blackenergy).

Vita

Tyler is a concurrent MS/PhD student in the Department of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville. He began studying there in 2016 after graduating magna cum laude from the University of North Carolina at Asheville, where he received the Tackett Award for most distinguished CS graduate. He currently works as a graduate research assistant in the UT Computer Security Laboratory with primary interest in privacy, censorship circumvention, and anonymity. The subject of this work - routing and Internet security - is an additional focus area.

Tyler also has substantial software engineering experience, especially in the High Performance Computing domain. He has worked with several projects at Oak Ridge National Laboratory in algorithm development and implementation, including QMCPACK (Quantum Monte Carlo simulation), ASGarD (fully-kinetic 6D PDE solver), and a geospatial engineering project.