

Interdomain Route Leak Mitigation: A Pragmatic Approach

A Dissertation Presented for the
Doctor of Philosophy
Degree

The University of Tennessee, Knoxville

Benjamin Tyler McDaniel

August 2021

© by Benjamin Tyler McDaniel, 2021
All Rights Reserved.

For Liz. It's been a long journey, but you put joy in every step.

Acknowledgments

I would like to thank first my parents, Burt and Tammy McDaniel. The values of servant leadership, integrity, and perseverance that I learned from watching you give me direction and purpose — without your example, I would be lost. My fiancée Liz has provided me with unending love and patience throughout this process. I could never have done any of this without you. I suppose I will keep you around a while. Of course, I must also recognize my friends, who are carefully arranged in alphabetical order: Davey, Harry, Joe, Kate, Tyler, and Zach. You have never failed to encourage and support me. You're the best.

Special thanks go to my advisor, Max Schuchard, for taking on a reclamation project student with entirely too much on his plate. Your technical expertise, direction, and advice have kept me pointed in the right direction despite my best attempts to wander off. I'd also like to acknowledge my colleagues Jared, Savannah, Parker, Jordan, and Joseph at VolSec, who have provided me with a lot of laughs - and even a little assistance on research from time to time. It's been quite a crew. To my committee members — Dr. Scott Ruoti, Dr. Jeff Nichols, and Dr. Jinyuan Stella Sun — your time and expertise are greatly appreciated.

Finally, I'd like to thank my sons, Charlie and Payton, for helping (forcing?) me to balance work and all the other parts of life. I love you, boys.

This work was supported by a grant from the National Science Foundation.

Abstract

The Internet has grown to support many vital functions, but it is not administered by any central authority. Rather, the many smaller networks that make up the Internet — called *Autonomous Systems* (ASes) — independently manage their own distinct host address space and routing policy. Routers at the borders between ASes exchange information about how to reach remote IP prefixes with neighboring networks over the *control plane* with the *Border Gateway Protocol* (BGP). This inter-AS communication connects hosts across AS boundaries to build the illusion of one large, unified global network - the Internet.

Unfortunately, BGP is a dated protocol that allows ASes to inject virtually any routing information into the control plane. The Internet's decentralized administrative structure means that ASes lack visibility of the relationships and policies of other networks, and have little means of vetting the information they receive. Routes are global, connecting hosts around the world, but AS operators can only "see" routes exchanged between their own network and directly connected neighbor networks. This mismatch between global route scope and local network operator visibility gives rise to adverse routing events like *route leaks*, which occur when an AS mistakenly advertises a route that should have been kept within its own network.

In this work, we explore our thesis: that *malicious and unintentional route leaks threaten Internet availability, but pragmatic solutions can mitigate their impact*. Leaks effectively reroute traffic meant for the leak destination along the leak path. This diversion of flows onto unexpected paths can cause broad disruption for hosts attempting to reach the leak destination, as well as obstruct the normal traffic on the leak path. These events are usually due to misconfiguration and not malicious activity, but we show in our initial work that

routing-capable adversaries can weaponize route leaks and fraudulent path advertisements to enhance data plane attacks on Internet infrastructure and services.

Existing solutions like Internet Routing Registry (IRR) filtering have not succeeded in solving the route leak problem, as globally disruptive route leaks still periodically interrupt the normal functioning of the Internet. We examine one relatively new solution - "Peerlocking" or defensive AS PATH filtering - where ASes exchange topological information to secure their networks. Our measurements reveal that Peerlock is already deployed in defense of the largest ASes, but has found little purchase elsewhere. We conclude by introducing a novel leak defense system, Corelock, designed to provide Peerlock-like protection without the scalability concerns that have limited Peerlock's scope. Corelock builds meaningful route leak filters from globally distributed route collectors and can be deployed without cooperation from other networks.

Table of Contents

1	Introduction	1
1.1	Thesis Statement	2
1.2	Outline	3
1.2.1	Chapter 3: Enhancing DDoS with Fraudulent or Intentionally Leaked Routes	3
1.2.2	Chapter 4: Peerlock Route Leak Defense: Measurement and Analysis	3
1.2.3	Chapter 5: Automated Route Leak Protection for Core Networks . .	4
2	Background	5
2.1	The Border Gateway Protocol	5
2.2	Customer Cones and Tiers	6
2.3	Route Leaks	6
2.4	BGP Poisoning	7
3	Enhancing DDoS with Fraudulent or Intentionally Leaked Routes	9
3.1	Introduction	9
3.2	Background	12
3.2.1	DDoS and Link Flooding Attacks	12
3.3	Can Botnets Target any Link?	13
3.3.1	Simulation Methodology	13
3.3.2	Vulnerability Experiments	14
3.4	The Maestro Attack	15
3.4.1	Threat Model	15

3.4.2	Maestro Concept	16
3.4.3	Poison Selection Algorithm	16
3.4.4	Evaluation	18
3.4.5	Incomplete Path Information	21
3.5	Internet Experiments	22
3.6	The Leak Attack	25
3.6.1	Attack Overview	25
3.6.2	Evaluation	26
3.6.3	Leak Defense	27
3.7	Attack Scope and Vulnerability	27
3.8	Towards Defenses	29
3.9	Related Work	31
3.10	Conclusion	31
3.10.1	Operator Engagement	31
4	Peerlock Route Leak Defense: Measurement and Analysis	33
4.1	Introduction	33
4.2	Background	36
4.2.1	Route Leak Prevention	36
4.3	The Peerlock System	37
4.3.1	Peerlock	38
4.3.2	Peerlock-lite	40
4.4	Measuring Peerlock Deployment	40
4.4.1	Measurement Methodology	41
4.4.2	Evaluation	45
4.4.3	Discussion	47
4.5	Exploring Peerlock’s Practical Impact	49
4.5.1	Simulation Methodology	50
4.5.2	Evaluation	51
4.5.3	Discussion	52

4.6	Related Work	54
4.7	Conclusion	55
4.7.1	Operator Engagement	56
4.7.2	Future Directions	57
5	Automated Route Leak Protection for Core Networks	58
5.1	Introduction	58
5.2	Background	60
5.2.1	Provider-Peer Observed Customer Cones	60
5.3	The Corelock System	61
5.3.1	The Core Enclave Ideal	61
5.3.2	Corelock Design	62
5.3.3	Corelock Protection Rules	63
5.3.4	Incomplete Data and False Positives	64
5.3.5	Alternative Approaches	65
5.4	Simulating Corelock Deployment	65
5.4.1	Methodology	65
5.4.2	Evaluation	66
5.4.3	Discussion	68
5.5	Relevant Historical Trends	70
5.5.1	Stability of the Internet’s Core Membership	70
5.5.2	Longitudinal Evaluation of Violation Rate	70
5.6	Related Work	70
5.7	Conclusion	71
6	Future Direction and Conclusion	73
6.1	Future Work	73
6.1.1	Routing-Enabled DDoS	73
6.1.2	Mitigating Leaks	74
6.2	Conclusion	75

Bibliography	77
Appendix	89
A Figures	90
Vita	107

List of Tables

1	Opportunistic Maestro: key links exposed to Maestro vs. Crossfire.	94
2	Strategic Adversary: Maestro result summary, provider to customer between- ness link selection, customer-only attack	94
3	Common leak filtering systems.	94
4	Most common encodings with number and percentage of ASes exporting leaks.	100
5	Analyzing leak segments by UCLA classes transited.	101
6	Route Collectors in Corelock Study.	101

List of Figures

1	2019 Verizon/Cloudflare leak. Other destination services were also affected. .	90
2	BGP poisoning. AS 1 originates a route with AS 2 prepended to path; AS 2 filters the update, but AS 3 propagates.	90
3	Flow density gain results (post-attack density - pre-attack density) by link selection strategy, Mirai botnet model	91
4	Deeper look at the customer attack success, betweenness link sample	91
5	LFA vulnerability as function of betweenness	91
6	Maestro: BGP poisons collapse botnet traffic onto a target link.	92
7	Reversing relationship to attack cust. to prov. links.	92
8	Attacking cust. to prov. links.	92
9	Maestro performance when missing traceroute hops and bot routes compared to complete information	93
10	Notional "botnet" distribution.	93
11	Valley-path leak attack overview and results	94
12	Evaluation of Maestro defenses	95
13	Simple Peerlock deployment. Protector AS filters updates containing the peer Protected AS from unauthorized propagators.	95
14	Example Peerlock-lite deployment. Provider AS filters updates from its customer that include a Tier 1 AS.	96
15	Measurement experiment depiction. Inferences are made about Peerlock deployment based on differences between normal updates (left) vs. poisoned updates (right) arriving at collectors.	96
16	Tier 1s filtering Tier 1 leaks, 2019/2020 measurements.	97

17	Overall filtering of Tier 1 leaks, 2019/2020 measurements.	97
18	Tier 1 peer leaks, 2019/2020 measurements.	98
19	Tier 1 leak filtering for MANRS/non-MANRS ASes.	98
20	Example simulated leak. Dashed red lines indicate route leak to other providers/peers.	99
21	Peerlock/Peerlock-lite simulation results.	99
22	The "core enclave" concept - routes only permitted to transit the enclave once.101	
23	Corelock: RIB-analysis based filtering.	102
24	Corelock update classification by regional collectors, 11/20-01/21. Error bars show std. dev.	103
25	Violations by CAIDA inference of next-hop relationship, 11/20-01/21, Route- Views Equinix.	103
26	Protector next hop relationship, violation vs. prior routes (subset), 11/20- 01/21, RouteViews Equinix.	104
27	Prefix length, IPv4 violation vs. prior routes (subset), 11/20-01/21, Route- Views Equinix.	105
28	Duration of violations resolved under 24 hours (when observable), 11/20, RouteViews Equinix.	106
29	Corelock update classification, November, 2017-2020, RouteViews Equinix. .	106

Chapter 1

Introduction

The Internet is composed of many smaller networks called *Autonomous Systems* (ASes), each with distinct allocated IP prefixes and independent routing policy control. ASes exchange routes to remote prefixes with neighboring networks over the *control plane* via *Border Gateway Protocol* (BGP) updates [69]. This inter-AS communication connects hosts across AS boundaries to form an (ideally) seamless global Internet.

Unfortunately, errors or malicious actions in the route exchange process frequently compromise the availability of Internet services. *Route leaks* [86], where an AS exports routes meant to be kept internally or shared only with select neighbors, are an especially serious problem. Because BGP updates do not carry the information required to easily distinguish a valid route from an leaked one, leaks can rapidly spread throughout the topology as ASes receive and propagate the mistake.

Leaks effectively steer some traffic bound for the leak’s destination along the leaked path. This re-routing is problematic because the leaked route is, by definition, not the intended path for this traffic. Often the leaked route is ill-provisioned to handle the additional flows, resulting in relatively poor performance or complete loss of connectivity to destination services for hosts in ASes that install and export the leak. Some leaks, like the 2017 Google-Verizon leak that struck Japan [49], can broadly degrade Internet service for entire countries. Leak mitigation usually involves slow out-of-band communication and cooperation between leak destinations and route leakers, so even highly disruptive leaks can persist for hours.

The 2019 Verizon-Cloudflare leak [45] (discussed in detail in Section 2.3) is a recent and high profile example. A small AS in Pennsylvania leaked a number of routes to Verizon, including several destined for leading content distribution network Cloudflare. Verizon transmitted these routes globally, funneling Cloudflare-bound traffic through the leaking edge network. This redirection caused two hours of degraded service for websites and services backed by Cloudflare while the firm’s engineers worked to contact the leaker [90]. At the incident’s peak, Cloudflare measured a 15% reduction in traffic moving through their network [45]. This event, and other regionally or globally disruptive route leaks [40, 22, 103, 78, 95, 50, 51] highlight the scale of the problem.

Most existing leak mitigation resources, like the many Internet Routing Registries (IRRs), are challenging to leverage or limited in scope. IRRs are databases where ASes can publish their routing policies. Other ASes can then convert IRR-stored policies into filters that validate received routes against remote ASes’ published intent. IRR-based filtering is limited by the correctness and freshness of information stored by remote ASes, however, so its effectiveness hinges on broad inter-network participation. This is a serious limitation as the motivations and sophistication of network operators varies greatly between ASes [41]. Other BGP security extensions, like the Resource Public Key Infrastructure (RPKI) [43], only enable filtering for a subset of leaks (e.g. re-origination or too-specific leaks for RPKI).

1.1 Thesis Statement

We argue here that: *Malicious and unintentional route leaks threaten Internet availability, but pragmatic solutions can mitigate their impact.* To demonstrate our thesis, we begin in Chapter 3 by evaluating and highlighting the danger posed by routing capable adversaries who weaponize BGP, showing how such an adversary can expose strategically valuable target links to data plane assets. This work leads to an evaluation of current leak prevention systems in Chapter 4, where we measure the distribution of one mitigation approach - defensive AS path filtering or "Peerlocking" - among the largest transit providers, the Tier 1 networks. Finally, we develop a novel system in Chapter 5 to extend protections against route leaks beyond the Tier 1 peering clique. All figures and tables will be presented in the appendix.

1.2 Outline

1.2.1 Chapter 3: Enhancing DDoS with Fraudulent or Intentionally Leaked Routes

In our preliminary work, we introduce novel attacks to demonstrate how routing-capable adversaries can exploit the lack of security controls in BGP to manipulate distributed denial-of-service (DDoS) flows [53]. This work motivates our study of adversarial routing countermeasures. Our first attack, Maestro, describes how adversaries can inject fraudulent BGP updates into the control plane to steer bot traffic onto target inter-AS links. Prior DDoS attacks against these links (called *Link Flooding Attacks* or LFAs) failed to consider how the AS-level topology partially or completely shields some links from botnets. We show both where vulnerable links exist, and how specially formed (poisoned) BGP updates can expose previously shielded links to attack by steering bot flows. We also outline a related leak attack, wherein an adversary triggers an intentional route leak to expose links traversed by the leak to malicious flows.

1.2.2 Chapter 4: Peerlock Route Leak Defense: Measurement and Analysis

Discussions with AS operators about mitigating these attacks lead us to examine a relatively new defensive filtering system, Peerlock [84]. Peerlock was introduced in 2016 to address the need for a common-sense leak solution that large transit ASes can deploy between their own networks without cooperation from un-incentivized peripheral ASes. While this system was designed to stop leaks, it also prevents the spread of Maestro’s BGP updates. Our preliminary work includes an active measurement of this system and one variant, Peerlock-lite, on the control plane [55]. We demonstrate that Peerlock/Peerlock-lite are present on the Internet, with deployment concentrated in the most influential global networks. Further, we show that the filtering protection provided by these systems is limited to leaks that transit core networks. This limitation is an artifact of Peerlock/Peerlock-lite design;

these systems generally require out-of-band manual communication/configuration for each additional protected network.

1.2.3 Chapter 5: Automated Route Leak Protection for Core Networks

Recent systems, informed by a long history of halting BGP security rollouts, have prioritized deployability at the expense of bulletproof security guarantees [26, 79, 84]. This thrust acknowledges inter-domain routing’s fractured governance structure and enables transit operators to make meaningful progress in protecting their own downstream networks with an automated system inspired by Peerlock. Corelock defines practical leak filters that can be assembled with only the tools available to an AS operator, e.g.: neighboring AS relationships, publicly-available routing data from regional routing registries or other collection points, and well-established fundamental routing models.

Chapter 2

Background

2.1 The Border Gateway Protocol

The Internet is a confederation of about 68,000 smaller networks, called *Autonomous Systems* or *ASes*. ASes exchange routing information via the Border Gateway Protocol (BGP) to enable global connectivity. Each AS originates routes to its hosted prefixes; these routes are advertised to neighbors via *BGP updates*. Each update contains a prefix and a collection of other attributes, including an AS PATH that describes the route's AS-level hops. ASes compare all received updates via the BGP decision process to select a single best path to every destination prefix. Both path qualities (like AS PATH length) and local network policies (e.g., business relationship with advertising AS) are taken in account when selecting a best path, but policies take precedence in the process. Once an AS selects a best path for a given prefix, it prepends its unique AS number (ASN) to the path and advertises *only* that path to its neighbors.

Paths learned from customer ASes - those purchasing transit - are advertised to all connections. Provider-learned routes, meanwhile, are generally only advertised to an AS's customers. Peer ASes exchange traffic without compensation, and likewise advertise routes learned from one another only to customer ASes. Limitations on non-customer learned route export prevents customer ASes from transiting traffic between peers/providers at its own expense. This dynamic, known as the Gao-Rexford or *valley-free routing* model [17],

guides the exchange of routes on the control plane. No widely-deployed mechanism enforces this model, but the economic incentives it describes shape AS path export behavior.

2.2 Customer Cones and Tiers

The *customer cone* [20] is an important abstraction arising from the valley-free model. An AS's *recursive customer cone* is the set of all ASes reachable from the AS via only provider to customer links. Stated simply, these are the AS's direct and indirect customers. Customer cone size is one of the few publicly observable features commonly used to judge an AS's influence on the control plane, e.g. in CAIDA's AS ranking [97]. Customer cone size is the basis for the UCLA classification presented in [64] that is widely used in research on this topic [81, 96, 2, 10, 101]. This scheme separates ASes into: 1) Tier 1 ASes, who have no providers, form a peering clique, and can transit traffic to any prefix without compensation; 2) large ISPs with more than 50 customer cone ASes; 3) small ISPs with 5-50 customer cone ASes; and 4) stub ASes with fewer than 5 direct/indirect customers. In general, we will use the term *customer cone* to refer to the recursive customer cone. In Chapters 4 and 5, we will employ a related abstraction: the *provider-peer observed customer cone*.

2.3 Route Leaks

Despite its vital role in binding together Internet networks, BGP is missing key security features like cryptographic hardening of routes exchanged between ASes or trusted certification binding ASes to owned prefixes. This leads to two common classes of major inter-domain routing mishaps, *prefix hijacking* and *route leaks*. Prefix hijacks occur when a network, often unintentionally, originates or advertises a fraudulent route to prefixes owned by another AS. Traffic destined for those prefixes is then intercepted by the hijacker. A number of recent studies focus on hijack mitigation; we focus on route leaks, so this falls outside the scope of our work [26, 77, 72, 100].

Route leaks are defined in RFC 7908 as the propagation of an advertisement beyond its intended scope [86]. Type 1-4 leaks all cover various valley-free routing violations, i.e.

advertising one peer/provider’s routes to another peer/provider. Because remote ASes have little or no information on relationships between non-neighbor networks, they generally cannot distinguish leaks from valid routes, and may propagate them throughout the topology. Type 5 leaks occur when one provider’s routes are announced to another with the AS PATH stripped, effectively re-originating the prefix from the leaker. Finally, a Type 6 leak involves an AS announcing routes used internally to its neighbors. These routes are often more specific than externally announced routes; this makes the leaks more attractive in the BGP decision process and encourages their spread to other remote networks.

As noted in the introduction, globally disruptive route leaks occur frequently [40, 78, 22, 50, 103, 51]. To describe how leaks propagate in greater detail, we examine the 2019 Verizon-Cloudflare leak [45]. A small ISP, AS 33154, leaked specific internal prefixes (Type 6) to Cloudflare and many other destinations to its customer, AS 396531. AS 396531 committed a Type 1 leak by advertising this route to its other provider, AS 701 Verizon. Verizon propagated the leak, which spread widely on the control plane because it was more specific than legitimate available routes (see depiction in Fig. 1). This leak exhibits a number of features common to routing failures with wide impact: 1) large networks are involved in distribution, 2) many prefixes or prefixes that host in-demand services are leaked, and 3) outages and high relative latency continue until the leak is corrected via out-of-band communication.

2.4 BGP Poisoning

BGP poisoning is a technique designed to manipulate the BGP decision process in remote networks. ASes originating a prefix can *poison* an advertisement by including the ASNs of remote networks in the AS PATH. Often, the poisoned ASNs will be inserted between copies of the origin’s ASN. This "sandwiching" ensures traffic is routed properly and that the advertisement is valid for Route Origin Verification (ROV) filtering purposes (see Fig. 2). BGP prevents cycles from forming in the topology by requiring ASes to drop routes containing their own ASN in the AS PATH; this feature is known as BGP loop detection. So, all networks included in the poisoned update’s AS PATH - the *poisoned* ASes - will filter

it. Poisons are generally used for inbound traffic engineering purposes [82, 81, 39], but we also employ them in our preliminary work to mimic route leaks transiting the poisoned AS.

Chapter 3

Enhancing DDoS with Fraudulent or Intentionally Leaked Routes

3.1 Introduction

Our initial work demonstrates the danger posed by fraudulent routes to proper Internet functioning. In this chapter, we develop combined control- and data-plane attacks that allow adversaries to concentrate (amplify) distributed denial of service (DDoS) attacks against target links. 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 [52]. *Link Flooding Attacks* or *LFAs* are DDoS attacks against infrastructure links rather than end hosts. Prominent examples in the literature include the Coremelt [91], Crossfire [35], and CXPST [74] 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 [76].

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 choose

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 [91] 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 [74] 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 [35] 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 Section 3.3, 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 more than 75% of bots. Even if we restrict our set of targets to the 10% most traversed links, the majority are reachable by less than 20% of the botnet.

In Section 3.4, 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 Section 3.7, where we analyze two Maestro adversary types.

We validate Maestro’s underlying mechanisms with experimental "attacks" on the live Internet in Section 3.5. 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.

We make the following key contributions:

- We measure how Internet routing properties limit Link Flooding Attacks in Section 3.3.
- We develop a technique to overcome these limitations: the Maestro attack. We describe and evaluate the attack via realistic simulations on an up-to-date Internet model in Section 3.4.
- We ethically execute Maestro proofs-of-concept on the live Internet in Section 3.5.
- We develop a broad understanding of the Internet’s vulnerability to Maestro in Section 3.7.
- We describe a related valley path leak attack targeting Tier 1 peering links in Section 3.6 and evaluate it with the same simulation framework.
- We propose and evaluate mitigations to our attack, as well as seek operator feedback from mailing lists in Section 3.8.

3.2 Background

3.2.1 DDoS and Link Flooding Attacks

Distributed Denial of Service (DDoS) describes a coordinated attack on a target link or end host availability with traffic from multiple sources. A common flow source for these attacks are botnets, or networks of compromised end hosts (bots) under an attacker's control. Botnets can include PCs, IoT devices, and/or SCADA systems, and are available for rent as attack sources from illicit online marketplaces. The most common form of DDoS is volumetric DDoS, where an attacker overwhelms a target with malicious traffic flows. These attacks have been carried out by nation-states [71, 25], and can also be used to isolate or degrade Internet performance for large geographic regions by targeting core DNS or network infrastructure providers.

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 Coremelt [91]. Coremelt specifies that a botmaster: 1) maps which links are present on paths on routes between bots; 2) targets a specific link used on paths between many bots; and 3) directs bot traffic *between 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 Internet links, but has the more ambitious goal of isolating an entire region (military installation, university, geographic region, etc.) by targeting key links [35]. 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, a pattern that makes Crossfire extremely difficult to detect and counter.

3.3 Can Botnets Target any Link?

Link Flooding Attacks (see Section 3.2.1 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 [35]; 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 [96]; and/or 3) target links most accessible to botnet flows, as in Coremelt [91], CXPST [74], 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.

3.3.1 Simulation Methodology

Our experiments are performed by extending the Chaos BGP simulator from prior work [75, 96, 82, 74] with new modules. Chaos builds the Internet’s BGP topology based on publicly available inferred AS relationship data from CAIDA [98]. 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. Since 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 [80, 39, 2].

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 [60]. 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 [94]. The *Blackenergy* model is a SCADA-focused botnet developed from similar techniques as presented in [9] 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.

3.3.2 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. Fig. 5a 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. 5b 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. 5c. **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 we found that low betweenness links are most difficult to reach, even highly trafficked links are not always fully exposed to botnet flows.

3.4 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.

3.4.1 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 [67]. **Recent events demonstrate that multiple feasible avenues exist for adversaries to gain routing capability.** The 3ve fraud operation [23] demonstrated the most straightforward route - simply registering a new AS. Insider attacks from disgruntled network operators, e.g. the Canadian bitcoin hijack [47], 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 [33]. However, recent Cisco router vulnerability disclosures show that remote attacks on edge routers are also a realistic attack vector [13, 11, 12]. Finally, BGP has previously been weaponized for intelligence gathering [16] and censorship [15] by nation states.

3.4.2 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. Fig. 6 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 $5 \mapsto 8$.

3.4.3 Poison Selection Algorithm

BGP poisoning is a primitive that moves inbound traffic *off* a chosen link as seen in prior related work [39, 82]. 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 [96]. The adversary uses traceroutes [32] to determine bot paths as in [35, 91].

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
     $B = \{b \mid b \text{ is a bgp path } To \mapsto Adv\}$ 
     $Sacred = \{From, To, Adv\} + \bigcap_{i=1}^{|B|} B_i$ 
     $Success += \{s \mid s \in Sources \text{ and } \{From, To\} \in s \mapsto Adv\}$ 
     $Disconn += \{s \mid s \in Sources \text{ and no specific-prefix path } s \mapsto Adv\}$ 
     $Sources -= \{Sacred \cup Success \cup Disconn\}$ 
     $Score = [0] * |Sources|$ 
    foreach  $s_i \in Sources$  do
      foreach  $s_j \in s_i \mapsto Adv$  do
         $Score_j += 1$ 
      end
    end
     $Poisons += \{max(Score)\}$ 
    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 all 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 Section 2.4) causes the AS PATH to exceed the size AS operators will almost certainly filter in practice: around 254 hops [96, 66]. We show in Section 3.8 that this condition is rarely met.

3.4.4 Evaluation

To evaluate Maestro’s impact, we choose thousands of target link/adversary AS pairings for simulated botnet attacks in the framework described in Section 3.3.1. 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 Figures 4a and 4b. 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. 4b). 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 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. 4c. Virtually all successful cases for this experiment were on provider to customer links.

Fig 4a displays pre vs. post attack flow density for the same betweenness link sample in Fig 3a, 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 greater than 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. Fig. 7 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. 8.

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 this chapter, 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.

3.4.5 Incomplete Path Information

The Maestro adversary must poison bottlenecks in off-target paths from bots to the compromised AS to concentrate traffic on the target link. As in prior LFAs [35, 91], path information is collected via the traceroute utility [32]. Traceroutes are resolved to AS-level paths by a IP-to-ASN mapping service like Team Cymru’s [93]. While traceroute is commonly used for topological discovery, the information provided by the tool is incomplete in practice for a number of reasons. For example, traceroute responses are delivered via ICMP, which some router operators disable or rate limit for security reasons [24]. These ICMP controls yield occasional missing hops in the path data provided by traceroute. So, we want to examine Maestro’s sensitivity to incomplete bot path information.

We investigate two classes of missing information: 1) missing some AS-level hops across all bot traceroutes, and 2) missing path information for some bot-hosting ASes. The first class models failure to respond to traceroute probes by some routers along bot traffic paths, possibly due to outgoing ICMP controls, packet loss, or opaque tunneling. We consider missing *AS-level* hops; this is a more challenging condition than missing single IP-level hops, since any one reported intra-AS hop is sufficient to resolve the AS-level hop. The second class represents failure to collect traceroute data from some bot-hosting ASes due to command-and-control software problems, intermittent connectivity issues, or network controls on incoming ICMP. Because BGP requires ASes to select a single best path to any destination prefix, all bots within an AS are expected to utilize the same path to the compromised AS. For this reason, a single traceroute per bot AS is sufficient to find

bottlenecks, but we consider the stronger case where no path information is provided for a sampled subset of bot ASes.

A 2006 study found 2% of router hops were anonymized in traceroutes [3], but we know of no current, authoritative source for the percentage of routers anonymized or missing from traces. Similarly, we cannot determine a realistic figure for how many bot traces could be lost entirely. Given this lack of data, we perform our experiments by re-simulating the customer cone attack from Section 3.4.4 with the Mirai botnet on the betweenness link sample across a broad range of missing information percentages - 2, 10, 20, 40, and 60% missing information. This is the percentage of ASes not appearing on traceroutes for the missing AS-level hops experiment, and separately, the percentage of bot ASes reporting no path information. The results are presented in Fig. 9. The percentage of missing information is on the x -axis and the ratio of flow density gain with missing information to flow density gain with perfect information is on the y -axis.

We find that the missing traceroute hop information has a measurable but modest effect on achieved flow density. Even with 20% of AS-level hops deleted from bot to compromised AS paths, an order of magnitude more than reported in [3], more than 75% of the full information attack effect is achieved. This is an intuitive result, as the adversary does not require the complete path to select the next poison; rather, only the bottleneck must be determined. Additionally, even if the ideal bottleneck hop does not respond to probing, poisoning another hop along shared off-target paths is likely to shift some traffic onto the target link. The absence of traceroute data from some bot ASes (dotted line) has a more significant impact on Maestro performance, but nearly 60% of the complete information effect is retained even when 10% of bot ASes do not provide any path information.

3.5 Internet Experiments

We built an experimental Maestro implementation with the PEERING BGP testbed [73] and the RIPE Atlas framework [89] 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, 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 [93]. Total flow density includes all anchors with paths to one another or the compromised AS that transit the target. Figure A shows the global distribution of our "bots".

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 Section 3.4.2 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 [3] 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 $209 \mapsto 2722$, 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 [98]. 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 [84] 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 $209 \mapsto 2722$. 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, $174 \mapsto 209$, 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 [6]. 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. This filtering is largely a byproduct of defensive AS PATH filtering as described in Chapter 4.

3.6 The Leak Attack

None of our previous evaluations have targeted peer links. In the CAIDA inferred topology, however, about 410,000 of 660,000 total Internet links are peering links. These links play a critical role in transiting traffic between Tier 1 providers in the Internet’s core. Like customer to provider links, attack path viability in this case is limited by BGP path export rules. Unlike those targets, we cannot simply reverse the direction of the attack; peer link export rules are the same for both endpoints. A Maestro attacker is therefore limited to bot flow sources that exist inside the From AS customer cone.

3.6.1 Attack Overview

An AS located in the From AS customer cone of a peer link target is not limited to the Maestro attack. An adversary who has compromised an AS there has a much simpler means of altering the control plane to increase flows: leaking valley paths that cross the target link. By *leaking*, we mean that the adversary AS advertises to its other providers/peers valley paths over the target link to bot-infected ASes in the To AS customer cone. That

is, the adversary exports a route that crosses the target link en route to a direct/indirect customer of the To AS. Normally, these paths would only be advertised to the compromised AS’s customers; the compromised AS has no interest in transiting traffic between providers at its own expense. But if the AS has been compromised by an adversary, or the AS’s operators value attacking the target link above the cost of the attack traffic, this technique could be an effective means of exposing the peer leak to malicious flows. Figures 11a and 11b illustrate this technique, which we term a *leak attack*. Note that nothing prevents data-plane traffic from transiting valley paths; the Gao Rexford model is upheld strictly by path export decisions. While we only evaluate the leak attack in the context of peer links, an adversary could similarly leak a customer to provider link.

Unlike the Maestro attack, leak attack traffic transits the adversary AS *before* crossing the target link. So, this attack is limited in flow density improvement by the capacity of the adversary AS’ inbound links, as well as the capacity of links between the adversary AS and the From AS (its direct or indirect provider). Of course, the attacker does not need to consume the entire capacity of the target peer link, only to increase flow density such that normal and malicious traffic *together* exhaust link capacity. Here we present results without consideration for potential pre-target bottlenecks, with an examination of relative link capacities left to future work.

3.6.2 Evaluation

We evaluate this leak attack on the Tier 1 clique; that is, all 400 peer links connecting the 20 Tier 1 providers. For each target link, we sample adversaries from 1-3 hops in the From AS customer cone, and measure bot-to-bot flow density before and after the adversary AS leaks all routes containing the target link. Fig. 11c shows the pre vs. post attack CDF by flow density for these simulations. According to CAIDA’s inferred topology and our real-world botnet IP mappings, these critical links in the Internet’s core are almost completely isolated from potential LFA traffic before the attack. Under all botnet models, we observed significant improvements. For the Mirai botnet, for instance, roughly half of Tier 1 link/adversary AS pairings had post-attack flow densities of 20% or greater. Notably, these Tier 1 to Tier 1

links have *virtually no vulnerability* before the attack, and *50% or greater exposure to IPs in major botnets* after leak execution from a small customer.

3.6.3 Leak Defense

Leak attack mitigation is a challenging problem. No current, widely deployed method exists to filter advertised valley paths in general. Worse, ground-truth AS relationships are not publicly available. In fact, valley paths have been observed on the Internet in practice and studied in prior work [21]. The cost of transiting attack traffic over the leaked route could be prohibitive for some adversaries, though this would not dis-incentivize an attacker who has compromised (does not actually own) the AS. The surest defense for upstream providers who wish to defend themselves from this attack is *monitoring advertisements from their customer cone for leaked routes*. In response to suspect advertisements, the defender could respond by disconnecting the violator. Finally, defensive filtering with out-of-band relationship information as observed in Section 3.5 and in the SICO attack [6] could deter this attack.

3.7 Attack Scope and Vulnerability

As shown in Section 3.4.4, 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 [98], 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 [64]: 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.

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 Section 3.4.4, which contains randomly selected high, intermediate, and low betweenness links. We analyze the results of the Maestro attack simulations from Section 3.4.4 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 [97] 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.

3.8 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 [42], deployment of next-generation architectures [4], or additional hardware [34, 46, 68]. Nyx, a re-routing system [82] 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.

The path length defense - rejecting advertisements above some limit - is one easy-to-implement response to this attack. Unfortunately, as shown in Fig. 12a, 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. [96] 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. Fig. 12b shows the impact if all ASes limited advertisement AS PATH length at various levels, including those observed by prior work (30 and 75) [96] and implemented in common routing hardware (254) [66]. 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 Section 2.4. Alternatively, ASes could prevent Meastro 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. Note that the Peerlock leak defense system (see Chapter 4) will filter some poisoned advertisements as deployed. However, deployments are limited to the largest ASes, and other inbound path control techniques - like the NO EXPORT communities employed in the SICO attack [6] - could be used in place of poisoning.

Fig. 12c 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). This scenario is broader than the Peerlock protection measured in 4, as Peerlock filters largely ignore non-Tier 1 poisons.

3.9 Related Work

The Coremelt [91] and Crossfire [35] attacks are discussed in detail in the background, Section 3.2.1. Bellovin's link cutting attack [5] 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 [74]. Select LFA mitigation work is presented in Section 3.8. Other uses of BGP poisoning include LIFEGUARD from Katz-Bassett et al. [38, 39] as well as Colitti et al. and Anwar et al.'s policy exploration studies [2, 14]. Nyx [82] from Smith et al. employs BGP poisoning for DDoS mitigation. The propagation of poisoned advertisements throughout the Internet is actively measured in [80]. Our work in Chapter 4 employs poisoning to trigger route leak detection systems.

3.10 Conclusion

3.10.1 Operator Engagement

We submitted a preprint of the study behind this chapter 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 [84]. 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 [55]. We encountered some evidence of Tier 1 peer locking in Section 3.5, as did Smith et al. [80] in their measurement study. Our Tier 1 filtering examination from Fig. 12c provides a window into the effect of full Tier 1 peer locking.

In this chapter, 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. Additionally, the leak attack introduced here serves to illustrate how route leaks can be weaponized in the absence of common-sense filtering schemes.

Chapter 4

Peerlock Route Leak Defense: Measurement and Analysis

4.1 Introduction

As discussed in Chapter 2, the Internet consists of many ASes with distinct IP prefixes, routing policies, and inter-AS connections. These networks exchange routes with neighboring ASes over the control plane to connect hosts in disparate ASes and create the illusion for users of a single, unified Internet. Unfortunately, there are few security controls on route exchange. ASes behaving adversarially, whether intentionally (as in the prior chapter) or by mistake, can export routes that should be kept internally or shared with only a subset of their neighbors as shown in the prior chapter. Because the language ASes use to communicate - BGP - does not package validation information with routes, remote networks often receive and propagate these route leaks throughout the control plane. Leaks frequently steer user traffic on the data plane onto unintended paths that lack capacity for the additional traffic. The end result is soaring latency or complete availability loss for destination services. Recent route leaks to prefixes hosting major content/service providers like Spotify [40], Cloudflare [45] and Google [22] have highlighted the global impact of this problem.

Existing tools designed to curtail leaks, like the many Internet Routing Registries (IRRs), are challenging to deploy or limited in scope. IRRs are databases where ASes can publish their routing policies. Other ASes can then convert IRR-stored policies into filters to validate

received routes. IRR-based filtering is limited by its requirement for broad AS participation, however, as the motivations and sophistication of network operators varies greatly between ASes [41]. Other BGP security extensions, like the Resource Public Key Infrastructure (RPKI), only enable filtering for a subset of leaks (e.g. re-origination leaks for RPKI).

We encountered selective filtering of poisoned BGP updates in our proof-of-concept live Internet "attacks" in Chapter 3. Given the relative rarity of poison filtering observed in prior work [81] and the broad use of poisoning as a primitive in attacks, measurements, and defensive systems [2, 54, 6, 39, 82], we sought to uncover the design and placement of these filters within the control plane. As we will show in this chapter, our poisoned attack advertisements were likely arrested by new systems deployed to mitigate route leaks.

The *Peerlock* [83, 84] leak defense system was presented in 2016 to address the need for a deployable leak mitigation solution. Each Peerlock deployment occurs between two neighboring ASes, the *protector* AS and *protected* AS. The protector AS agrees to filter routes that transit the protected AS unless they arrive directly from the protected AS or one of its designated upstreams. The filter prevents the protector AS from propagating or steering its traffic onto any leaked route that transits the protected AS, regardless of origin AS/destination prefix. Peerlock is designed to leverage the rich web of relationships that exist between transit networks in the Internet's core, and functions without coordination with other ASes on potential leak paths. This makes Peerlock especially viable in the *peering clique* formed by the 19 *Tier 1* ASes that sit atop the inter-domain routing hierarchy. A related technique, Peerlock-lite, enables networks to spot likely leaks without prior out-of-band communication. ASes deploying Peerlock-lite drop routes arriving from customers that contain a Tier 1 AS; it is highly improbable that customers are providing transit for large global networks.

Our first contribution is a measurement of Peerlock/Peerlock-lite deployment on the control plane. In Section 4.4 we design, execute, and evaluate active Internet measurements to search for evidence of filtering consistent with these systems. Our experiments use BGP poisoning, a technique used in prior work for traffic engineering [82] and path discovery [2], to mimic route leaks that transit some target AS. We then listen for which networks propagate - or filter - these "leaks" relative to control advertisements. This information feeds several

inference techniques designed to uncover which ASes are Peerlocking for (protecting) the target AS.

Notably, we find substantial Peerlock deployment within the peering clique: about 48% of possible filtering rules (153/342) are already implemented within this set. Further, many non-Tier 1 ASes - including nearly 40% of large ISPs observed during our experiments - perform some Peerlock-lite filtering on Tier 1 AS leaks. Evidence for Peerlock filtering of non-Tier 1 leaks is virtually nonexistent, though three Tier 1 networks (AS 12956, AS 2914, and AS 3320) each filter leaks for more than 20 non-Tier 1 ASes.

After detecting current Peerlock/Peerlock-lite deployments, we ask how well these systems mitigate Tier 1 leaks. Internet-scale BGP simulations in Section 4.5 test over 6,000 simulated Tier 1 leaks against observed Peerlock/Peerlock-lite levels to quantify the effect of these systems as deployed. We test the same leaks against six hypothetical extended deployment scenarios to understand where additional filters should be placed to isolate leaks.

We find that Peerlock filtering within the peering clique is helpful, but not sufficient to mitigate Tier 1 route leaks on its own. Consistent with prior work on BGP filtering [18], our experiments show that positioning filters at relatively few ASes - the roughly 600 large ISPs - can play a decisive role in leak prevention. About 80% of simulated leaks were completely mitigated by uniform Peerlock-lite filter deployment at large ISPs, with fewer than 10% of leaks spreading beyond 10% of the topology. These figures are especially encouraging because Peerlock-lite is based on a simple route validity check informed by the valley-free routing model [17] that requires no out-of-band communication.

In this chapter, we make the following key contributions:

- We give an overview of the Peerlock and Peerlock-lite filtering systems, and consider their benefits and limitations relative to existing tools in Section 4.3.
- We describe how we adapt existing Internet measurement techniques to probe Peerlock/Peerlock-lite deployment on the control plane and introduce a novel inference method in Section 4.4.

- We actively measure where Peerlock and Peerlock-lite filters are deployed with PEERING [73] and CAIDA’s BGPStream [65] in Section 4.4.2, with a discussion of results in Section 4.4.3.
- We simulate thousands of Tier 1 route leaks against several protection scenarios in Section 4.5.2, and present a new path encoding method to understand how these scenarios influence leak propagation and export in Section 4.5.3.

4.2 Background

4.2.1 Route Leak Prevention

There are a number of tools available to assist network operators in preventing route leaks. The Resource Public Key Infrastructure [43] is a trusted repository for certificates that bind IP prefixes to owning ASes’s public keys, called Route Origin Authorizations (ROAs). Remote networks can validate BGP updates against ROAs in the RPKI, a process called Route Origin Validation (ROV). Widespread ROV filtering could prevent Type 5 leaks and many prefix hijacking attacks. Unfortunately, ROA/ROV deployment has suffered from circular deployment logic; it is meaningless for origin ASes to invest in publishing ROAs until ROV is widely implemented, but ROV is ineffective without ROAs. This issue has been identified as a major obstacle to ROV deployment [18, 26]. NIST estimates that just 21% of prefixes are covered by a valid ROA [62].

Internet routing registries (IRRs) back another leak prevention system. IRRs are databases where AS operators can store their routing policies. Remote networks can ingest these policies to inform filters that block unintended/invalid advertisements. IRR databases are operated by private firms, regional Internet registries, and other interests [56], and policy entries are often mirrored between them. A complete, up-to-date IRR would eliminate many Type 1-4 route leaks. Like ROV filtering, though, IRR filtering is hampered by deployment headaches. ASes’ routing policies are interdependent, so changes to one network’s stored policies can render many others obsolete. Operators in smaller, resource-limited networks can avoid periodic updates by configuring permissive routing policies; large transit ASes

have complex, dynamic routing policies that require frequent changes to dependent networks' filters [41]. These issues, combined with poor or non-existent authentication, have resulted in inconsistent and out-of-date IRRs. Though leading organizations like RIPE have launched efforts to improve IRR quality [70], operator incentive and dependency issues will continue to limit their usefulness.

Other filtering techniques include max-prefix limit filtering, where a network caps the number of prefixes it will accept from a neighbor. This prevents mass prefix spills like the 2017 Level 3 leak [50], but not more targeted (yet highly disruptive) leaks like the Verizon/Cloudflare incident described earlier. BGPsec [44] is a protocol extension for cryptographic AS path hardening. This would prevent some types of hijacking, but BGPsec has not been commercially implemented and is not designed to prevent route leaks.

A communities-based "down-only" (DO) leak solution has also been proposed [87]. Large BGP communities [85] are signals containing three integers that can be attached to routes. The DO system relies on providers/peers marking a route "down-only" using the first two integers in a large community, with their ASN included as the third integer, before passing the route to customers or peers. If these customers/peers attempt to re-advertise the route to other providers/peers, the attached DO community will clearly signal a route leak. While this system would prevent many leaks if properly implemented, it relies on customers/peers preserving DO information when propagating advertisements. Moreover, some leaks - like the internal route leaks in the Verizon/Cloudflare incident - would not be arrested by this system.

Finally, Peerlock/Peerlock-lite are designed to detect Type 1-4 route leaks by applying out-of-band information about the ASes on an advertised path. These systems are explored in great detail in this chapter.

4.3 The Peerlock System

The challenge of leak filtering stems from the topological scope asymmetry between BGP routes and the perspective of individual AS operators who evaluate them. Routes span the topology (global scope); operators only know their own relationships with adjoining ASes

(local scope). Filtering systems built on the RPKI [29] and IRRs [61] partially address this asymmetry by applying additional information to the route evaluation process. However, these existing solutions have limitations that have hamstrung their deployment. Most critically, their effectiveness depends on the cooperation of many unincentivized remote ASes as detailed in Section 4.2.

4.3.1 Peerlock

Peerlock, first detailed by NTT in 2016 [83, 84], is a leak filtering scheme based on out-of-band information exchange between BGP neighbors. Peerlock requires a single AS (the *protected AS*) to designate authorized upstreams to their BGP neighbor (the *protector AS*). This communication distributes AS relationship information between peers to decrease route/filterer scope asymmetry. The protector AS then rejects any BGP update whose AS PATH contains the protected AS unless received 1) directly from the protected AS, or 2) from an authorized upstream, with the protected AS immediately following the authorized upstream in the AS PATH. We say that the protector AS is Peerlocking for the protected AS. See Fig. 13 for a depiction of the system. In this chapter, we will often refer to a single instance of Peerlock - that is, one protector/protected pairing - as a *Peerlock rule*.

Here we describe Peerlock’s benefits and drawbacks relative to previous leak prevention systems, each of which is described in detail in Section 4.2. These comparisons are summarized in Table A.

RPKI/ROV Comparison: Peerlock provides broader leak type coverage than RPKI/ROV filtering without a trust infrastructure requirement. However, Peerlock only applies to leaks that violate configured topological rules (Types 1-4), so Type 5 (re-origination) and Type 6 (internal route) leaks fall outside its scope. ROAs tie prefixes to valid originating ASes, so ROV filtering can prevent Type 5 leaks. Additionally, ROAs can be configured with a max prefix length to prevent some internal route leaks and hijacks, although recent work has identified vulnerabilities in this feature [19]. Because Peerlock and RPKI/ROV filtering cover different leak types, Peerlock is *complementary* to ROV filtering rather than a replacement.

IRR Comparison: IRRs are policy object databases capable of storing participating networks’ routing intentions with great detail and fine granularity (prefix level). Any AS wishing to enforce these intentions can automatically derive filters from stored objects using software tools, whereas Peerlock rule configuration requires setup between each protector/protected AS pair. Unfortunately, IRRs suffer from incentive misalignment, governance, and rule dependency issues as described in Section 4.2. Peerlock rules are self-contained, and changes do not affect other rules. This encapsulation avoids the cascading dependency problem exhibited by IRRs, where one AS’s policy changes may render many other AS’s entries obsolete.

Most importantly, Peerlock allows the protector AS to stop leaks that transit the protected network regardless of the actions of ASes along potential leak paths; the value of IRR-based filters depends on many remote networks to store accurate policy entries. Peerlock’s relatively light cooperation requirement only requires that ASes with an existing relationship communicate information between themselves. This dynamic enables the best resourced, positioned, and incentivized networks (i.e., those serving the most customers) to block route leak propagation regardless of other remote networks’ actions.

Max Prefix Comparison: BGP’s max prefix feature enables networks to limit the number of prefixes they will accept over each neighboring AS connection. Mass route leaks - those involving many prefixes - are filtered once prefix volume over an inter-AS link exceeds the preset limit. Max prefix filtering affords broad protection across leak types, but cannot stop leaks involving few (potentially critical/highly trafficked) prefixes. On the other hand, Peerlock cannot stop leaks that do not violate established topological constraints regardless of volume, but is effective against more selective leaks unprotected by max prefix limits.

Other Considerations: Currently, each Peerlock rule must be manually configured, although at least one method has been proposed to automate Peerlock [30]. Peerlock also lacks a standard to describe how out-of-band information is exchanged between participants. Without a detailed and secure protocol for rule configuration, Peerlock is vulnerable to exploitation; fraudulent rules affect route export, and could be used to engineer traffic flows.

Furthermore, operators must define their own ad-hoc protocols for communicating rules that may not guarantee authenticity and/or confidentiality. Virtually all leak solutions discussed here, including IRR, RPKI/ROV, and AS PATH filtering, are recommended by the best practices group Mutually Agreed Norms for Routing Security (MANRS) [57].

4.3.2 Peerlock-lite

Peerlock-lite [84] (or Tier 1 filter, "big networks" filter) is a related technique, based on the assumption that transit providers should never receive a route whose AS PATH includes a Tier 1 AS from a customer. This is a valid assumption under the valley-free routing model [17], because such an update implies the customer is providing transit for the Tier 1 AS; otherwise, the customer would not export (leak) the route to another provider. However, Tier 1 ASes have no providers by definition. This logic can be extended heuristically to any other large non-Tier 1 networks that the provider does not expect the customer to export.

This simple logic yields an equally simple filtering rule for transit providers - reject any updates from customers that contain a Tier 1/large transit ASN. See Fig. 14 for a depiction of this filtering technique. Peerlock-lite filters are limited to Tier 1/large transit provider leaks, but they require no out-of-band information to configure. Moreover, Tier 1 ASes' position at the Internet's core results in their frequent presence on AS PATHs of highly disruptive leaks, e.g. the Verizon/Cloudflare leak [45] and the Enzu/AWS/Spotify leak [40].

4.4 Measuring Peerlock Deployment

Our initial experiments seek to establish the current state of Peerlock deployment on the control plane. As discussed in the previous section, every Peerlock rule is configured between a pair of networks: the protector AS and the protected AS. Each of the experiments in this section works to identify some or all Peerlock/Peerlock-lite protectors for a targeted AS.

4.4.1 Measurement Methodology

Experimental Design

Each set of measurement experiments in this section is designed to discover Peerlock rules for a set of potential protected ASes, called *target ASes*. For each target AS, we advertise a /24 prefix from many points-of-presence (*PoPs*) on the control plane. This is the *control advertisement*. It is a normal /24 origination in every way, except that our university AS - which we know not to be protected by any Peerlock rule - is poisoned (i.e., prepended to the advertisement's AS PATH - see Section 2.4). We then listen at varied collection sites, called *collectors*, for BGP updates triggered by our advertisement. The AS PATH for each such update that arrives at collectors lists in encounter order the ASes that received and re-issued the update as described in Section 2.1.

Taken together, the gathered AS PATHs form a directed acyclic graph (DAG) that describes the control advertisement's propagation through the control plane; each AS appearing on at least one AS PATH forms a node in the DAG, and AS ordering within paths allows us to form directed edges between nodes. BGP loop detection prevents cycles as noted in Section 2.4. We call this graph the *control DAG*. Note that all of the ASes appearing in the control DAG propagated (and thus did not filter) control updates that include a poisoned AS.

We then wait 30 minutes for update propagation before issuing an explicit withdrawal for the /24 prefix. This timing is built conservatively from empirical measurements of propagation times through the control plane (see update propagation experiments in appendix). After another waiting period to ensure the withdrawal has completely propagated, we issue a *leak advertisement* for the same /24 prefix. This advertisement matches the control advertisement in every way, except that the target AS is poisoned. This leak advertisement structure is designed to mimic a leak for the purposes of Peerlock while avoiding other common filtering systems. The target AS's presence on update paths triggers filtering for any Peerlock protector ASes.

Finally, we gather all BGP updates for the leak advertisement from our collectors. The ASes that appear on AS PATHs in any of these updates are added to a set called the *leak set*.

Since they propagated poisoned updates, we know these ASes did not filter the "leak". With the control DAG and leak set together, we can reason about which ASes are Peerlocking for the target AS using two techniques: 1) clique inference and 2) DAG inference.

For detecting Tier 1 protector ASes, we use *clique inference*. This simple technique relies on the fact that Tier 1 ASes form a peering clique by definition. According to the valley-free routing model [17], ASes share all updates received from customers with their peers; this maximizes the traffic the AS transits for its customers (and thus the AS operator's compensation). Further, ASes should not share a peer's updates with another peer, as this is a Type 2 route leak [86]. So, in general, if a Tier 1 AS is observed propagating an update, *all* Tier 1s should receive the update via their peering relationships. Because we observe at least one Tier 1 propagating control and leak updates across all experiments, we define a simple rule for inferring Tier 1 protector ASes: any Tier 1 AS that appears in the control DAG but not the leak set is Peerlocking for the target AS.

Inferring other protector ASes requires a more general technique. Outside the structural guarantees provided by the Tier 1 clique, there is significantly more uncertainty about which networks are filtering leak updates. Specifically, it is difficult to distinguish an AS filtering updates from an AS not receiving updates at all due to filtering by other upstream/downstream networks. This challenge leads us to make three separate inferences for these ASes for each leak target.

First and simplest is the *max inference* set, defined as all control DAG ASes minus the leak set. This set includes all ASes who *may have* filtered leak updates, but also ASes who did not receive the leak update because it was filtered by an intermediate AS. Secondly, we build a *min inference* set. This set is built by deleting all leak set ASes from the control DAG, and collecting the root of every weakly connected component that remains. This isolates the ASes that filtered leak updates from ASes in their "shadow" who did not receive the updates. The min inference set contains those ASes who likely filtered leak updates based on routes we observed. Note that the min/max inference techniques closely align with those employed in the long path filtering experiments in Smith et al.'s study on BGP poisoning as a re-routing primitive [81].

Our last inference set is the *likely inference*. Because ASes only export their best path to our /24 prefix, we cannot observe every edge that should exist in the control DAG (i.e., every potential propagation path for updates). So, this set’s is built like the min inference set, except that we augment the control DAG with edges from CAIDA’s provider-peer observed customer cone inference [98]. That is, we add edges to the control DAG where CAIDA’s data indicates there are links between ASes that we did not observe due to policy decisions. This forms a superset of the min inference set and a subset of the max inference set that contains the most likely filterers. This is a novel technique not used to our knowledge in any prior work on this topic.

These three inference sets are formed for each target from differences in control and leak update propagation. In addition to these sets, we also build a *min/max/likely poison filtering set* by following the same steps listed above, but with a *unpoisoned advertisement*’s updates compared against the control advertisement’s updates. These sets are built to explore the prevalence of general poison filtering as in Smith et al. [81].

Framework Details

The control-plane measurement framework for these experiments consists of 1) 13 PoPs to issue BGP advertisements and 2) 54 BGP collectors to listen for propagation. We employ the PEERING testbed [73] for the first requirement. PEERING allows us to advertise three assigned /24 prefixes from edge routers at thirteen PoPs worldwide. For collecting BGP updates, we used CAIDA’s BGPStream [98] tool. This tool draws updates from 54 globally distributed collectors, including 30 RouteViews [99] and 24 RIPE RIS [59] collectors. While most of these collectors are positioned in North America and Europe, every populated continent is represented by at least one collector.

Measurement Limitations

While our framework allows us to effectively probe the control plane for evidence of Peerlock and related techniques, a number of limitations prevent complete certainty regarding

Peerlock filter placement. The most important of these obstacles are imperfect collector coverage, topological instability, and the presence of other filtering systems. Here we discuss each of these factors in turn.

BGP policies prohibit us from viewing the entirety of the topology with our framework; there are few collectors in stub networks, and stub/remote ASes do not export received updates back "up" through provider networks. This means our *observation window* - the ASes on update paths at collectors - is biased toward transit networks in the Internet's core as in [64]. Fortunately, this is the most important/influential region to monitor, as these network's policies have the widest impact on the control plane. Altogether, we observed 610 ASes during our experiments, including 181/605 large ISPs and all 19 Tier 1 networks. Most observed ASes (332) were present in the observation window during all experiments conducted from August 2019-May 2020. Note that while we can only infer *protector* ASes from our observation window, we can poison any AS. So, our window does not limit our inference regarding which ASes are *protected*.

To account for instability in our observation window, we limit our filtering inferences to those ASes observed in control updates both before and after the leak advertisement (i.e., for the current and next target AS experiment). Additionally, we repeat experiments - issue control/leak advertisements for the same target ASes - over several months. These observations are combined to reduce the "noise" of topological dynamism from our inferences. Specifically, we remove ASes from a target's filtering inference sets if we later observe them propagating a leak update for that target; in this case, the earlier inference was likely caused by the AS's intermittent presence in the observation window during the experiment.

Most importantly, we acknowledge that we cannot be certain Peerlock/Peerlock-lite exactly as described by NTT [83, 84] is responsible for all observed filtering, but our experiments are designed to avoid common leak filtering systems. First, since the leak and control advertisements in our experiments share an origin AS/prefix, their updates present identically for ROV filtering purposes. Additionally, since we observe all ASes in the control DAG propagating control updates, we infer those ASes will not apply common IRR or max-prefix limit filtering to the same /24 in leak updates. Finally, while prior work indicates that short poisoned paths are frequently present on the control plane [96] and rarely filtered [81],

the poisoning in the control advertisement ensures that we do not conflate poison filtering and Peerlocking.

Despite our efforts to avoid common filtering practices, local policies grant network operators extensive discretion in how routes are vetted and exported. This flexibility means we cannot be certain that experimental updates are not sometimes blocked by AS specific, ad-hoc AS PATH filtering techniques. We know of no way to distinguish such functionally similar filters from Peerlock.

Ethics

We issued only well-formed BGP advertisements using the PEERING software client and adhered to all rules published by PEERING. We advertised only our assigned /24 prefixes, which are reserved for experimental use, and thus did not disturb Internet control or data plane operation for any non-experimental IP addresses. Our experiments did require poisoned advertisements, but this is a common practice used both in research [6, 81] and in traffic engineering [96]. One network operator observed and inquired about our experiments to PEERING, but did not report any resultant adverse effects. No data-plane traffic was sent during the conduct of our experiments.

4.4.2 Evaluation

Target Set 1, Tier 1s

The 19 Tier 1 ASes form our first target AS set, i.e. the potential protected ASes for which we are inferring Peerlock rules. The Tier 1 peering clique includes the most influential networks by one of the few observable metrics, customer cone size [98], and often creates [50] or distributes [45, 22, 78] leaks that disrupt global Internet services. Paradoxically, deploying filters for leaks that include Tier 1 ASes is also relatively simple for non-Tier 1 networks via the Peerlock-lite system described above. We iteratively issued unpoisoned, control, and leak advertisements that covered this set (repeated our measurements) every two months from August 2019 to May 2020. This repetition allows us to capture filtering rules for ASes with

inconsistent presence in our observation window, and to explore how deployments change over time.

We first present results for protection within the Tier 1 clique in Fig. 16. Note that because of BGP loop detection, every AS filters leak updates that include their own ASN regardless of Peerlock deployment. The peering clique is fortunately the most stable feature in our observation window, enabling us to measure the presence/absence of nearly every potential Peerlock rule within the clique. We have marked the exceptions for which we were unable to measure filtering rules in pink in Fig. 16. We see that Peerlock deployment is significant but unevenly distributed within the clique. Some ASes - e.g. AS 2914 NTT, AS 701 Verizon - filter leak updates for virtually the entire clique. For five others - e.g. AS 3491 PCCW Global, AS 6762 Telecom Italia - we found no evidence of Tier 1 Peerlock filtering at all.

Our measurement results for Peerlock/Peerlock-lite protection of Tier 1s by all observed ASes are depicted in Fig. 17. Fig 17a shows both our inferences about which networks filter poisoned updates in general (blue lines) and which filter Tier 1 leaks (red lines). These are displayed as a cumulative distribution function (CDF) over Tier 1 targets; likely inferred filtering levels range from about 3% (AS 6830) to 15% (AS 701) of observed ASes. Note that per the experimental design described above, we cannot make Peerlock protection inferences for ASes filtering all poisoned updates; however, this is a small set without Tier 1/large ISP members (max inference size = 9 ASes). Fig. 17 shows the number of ASes in each UCLA class (see Section 3.2) protecting at least one Tier 1 target.

Target Set 2, Tier 1 Peers

Our second target set includes the non-Tier 1 peers of Tier 1 ASes (about 600 ASes) as inferred by CAIDA [98]. These experiments explore whether Tier 1 ASes are extending Peerlock protection to their non-Tier 1 peers. Additionally, despite covering about 1% of all ASes, this set includes a third of all large ISPs. The presence of these large ISPs in the target set allows us to investigate whether non-Tier 1 ASes apply Peerlock-lite filters to large

transit networks outside the peering clique. These experiments were conducted from Oct 2019 to May 2020, with every included network targeted at least twice.

The overall results are presented in Fig. 18a. Clearly, **filtering for these leaks is less prevalent within our observation window**. 80% of Tier 1 peer leaks were filtered by fewer than 2% of observed ASes, but a few exceptional targets did trigger significant filtering behavior. Our poison filtering inference for these targets is, as expected, nearly identical to that derived from the Tier 1 leak experiments. Fig. 18b displays filtering levels for each Tier 1 ASes by peering status with the target. All Tier 1s protect 10 or fewer peer networks from this set. More variance exists in non-peer filtering behavior, as we will explore in the following discussion.

4.4.3 Discussion

Consistent with Smith et al. [81], we find no evidence for widespread filtering of otherwise unremarkable poisoned paths. Their study also found that poisoning high degree ASes in an update is associated with reduced propagation. Specifically, sub-20% update propagation rates were observed for some Tier 1 ASes, including AS 174 (Cogent/Tier 1) and AS 3356 (Level 3/Tier 1). Birge-Lee et al. [6] likewise found that using AS poisoning rather than communities as a path export control primitive significantly reduced update spread, especially when large transit providers were poisoned. Defensive AS-path filtering (e.g., Peerlock/Peerlock-lite) is identified as a likely culprit for this effect. Our work systematically examines how and where these filters are deployed on the control plane (within the limits of our observation window).

Tier 1 Leak Filtering

The greatest protection within our observation window is clearly afforded to Tier 1 ASes. Our initial experiments in August 2019 discovered evidence for 133/342 ($19^2 - 19$) possible Tier 1-Tier 1 filtering rules (about 39%). Each measurement that followed uncovered at least two new filtering rules, and by our final experiment in May 2020, 153 rules had been observed, a

nearly 15% uptick in Peerlock deployment. We had previously observed a negative filtering result for every additional rule, indicating this increase results from genuinely new Peerlock deployments rather than instability in the observation window.

Non-Tier 1 ASes also filter Tier 1 leaks, though this behavior is far from uniform. Overall, Tier 1 leak filtering ranged from 3% to 15% of observed ASes across Tier 1 AS targets. Most of this is likely due to Peerlock-lite filtering, as it is simpler to deploy. Moreover, fewer than 10% of the more than 1,000 observed Tier 1 filtering rules exist between peers, and only about 20% (236 rules) involved a Tier 1’s indirect customers filtering leak updates. This suggests that ASes are installing Peerlock-lite filters for all Tier 1s rather than simply protecting their upstream providers.

Mutually Agreed Norms for Routing Security (MANRS) [58] is an initiative whose ISP members agree to best routing practices (like AS path filtering) to secure inter-domain routing. While Peerlock and Peerlock-lite are not specifically included in MANRS expected filtering actions, they are both suggested in the implementation guide [57]. Fig.19 displays as a CDF the proportion of MANRS and non-MANRS ASes filtering Tier 1 leaks. 73 of 502 MANRS ASes fall within our observation window; the proportion of observed MANRS ASes that filtered Tier 1 leaks ranged from 2-18% depending on Tier 1 target. Non-MANRS filtering over the same target set ranged from 2 to 12%.

As shown in Fig 17b, the proportion of ASes with Tier 1 leak filters rises with UCLA class [64]. Intuitively, networks with larger customer cones have the resources for sophisticated configurations and the imperative to prevent issues for downstream customers, and have previously been associated with differing responses to BGP events [81, 7]. This dynamic hampers systems requiring wide participation like ROV [18] and IRR filtering [41], but does not limit Peerlock or Peerlock-lite deployment.

Tier 1 Peer Leak Filtering

Our non-Tier 1 leak experiments met with relatively sporadic filtering. For more than 80% of targets in this set, nearly every observation window AS ($\geq 98\%$) propagated leaks. As described in Section 4.3, Peerlock-lite filters for non-Tier 1 ASes require more careful

deployment. The outliers in this target set (see the long tail in Fig. 18a) are invariably near-Tier 1 networks like AS 1273 Vodafone, AS 6939 Hurricane Electric, and AS 7843 Charter that are safe for most ASes to include in a Peerlock-lite filter.

Tier 1 filtering of this leak set was likewise reduced compared to Tier 1 leaks. In general, Tier 1 networks deploy fewer than 5 Peerlock filters for non-clique peers. Nearly all of these cover near-Tier 1s like AS 7922 Comcast and AS 1273 Vodafone, or ASes administered by Tier 1s e.g. AS 702/703 Verizon and AS 3549 Level 3. Notably, three networks extend protection to more than 15 non-peers (per CAIDA’s inference). AS 2914 NTT’s non-peer filtering rules all cover various Comcast ASNs, while AS 12956 Telefonica’s rules appear to be regionally-based: zero rules are applied to customer cone ASes, but 23/31 apply to other European ISPs of varying size. 13/20 of AS 3320 Deutsche Telecom’s non-peer filtering rules, on the other hand, cover ASes within its customer cone.

In summary, Peerlock is widely deployed and expanding within the peering clique. Deployment outside the peering clique is relatively limited, however. Up to 20% of non-clique networks also deploy Peerlock-lite (or a similar mechanism) to filter leaks containing Tier 1 or near-Tier 1 ASes. These deployments are proportionally more common in ISPs and rarely seen in stub ASes within our observation window. Fortunately, the effectiveness of Peerlock/Peerlock-lite deployments is less sensitive to scattershot deployment than other filtering solutions. Prior work [18] and our simulations in the following Section 4.5 suggest that filtering by large ISPs can have an outsized impact on global leak propagation.

4.5 Exploring Peerlock’s Practical Impact

The substantial but limited Peerlock/Peerlock-lite filtering measured in the previous section leads us to investigate these systems’ protective benefit in partial deployment. We have interest both in how well these systems protect the control plane from Tier 1 leaks as deployed, and in the relative effect of realistic additional deployment (e.g. adding filters at large transit networks). To answer these questions, we quantify Peerlock’s practical impact with Internet-scale leak simulations against several filter deployment schemes.

4.5.1 Simulation Methodology

These experiments are conducted via extensions to a BGP simulator, an approach consistent with prior work on this topic [74, 75, 82, 96]. We construct a simulated AS-level topology from CAIDA’s inferred relationship dataset (Jan. 2020 data) [98]. ASes within the topology evaluate and export routes using the BGP decision process; longest-prefix matching, LOCAL PREF, and AS PATH guide path selection, while route export is governed by local policy to enforce valley-free routing. This ensures the simulator models the central dynamic of control plane propagation - the Gao-Rexford model [17], and allows for the closest approximation of control plane behavior we can devise without ASes’ full (private) routing policies.

Each simulation is driven by a *protection scenario* that maps protector ASes to those they are protecting. As with Peerlock in practice, these protectors drop all received routes that transit a protected AS unless they arrive directly from that AS. Some scenarios also include Peerlock-lite deployments; for these experiments, some set of ASes filter all customer-exported routes that transit Tier 1 ASes. Once we establish the protection scenario, we iterate over all Tier 1 to Tier 1 links (with 19 Tier 1 ASes, this is $n = 19, n^2 - n = 342$ links). These links describe a unidirectional connection from one Tier 1, called the *link start*, to another Tier 1, called the *link end*.

For every link in this set, we sample 20 ASes from the link start’s customer cone to serve as *leakers*. Each leaker will, in turn, randomly select a *destination* AS in the link end’s customer cone, and advertise a route to the destination over the link to all of its peers/providers (see Fig. 20). This models a Type 1 route leak of a path over the peering clique [86]. After the leak, we allow the topology to converge and measure how many ASes 1) received leak updates and 2) installed the leak path. Additionally, we capture all the AS PATH of all leak updates for analysis. With 20 leaker/destination pairings per link and 342 Tier 1 links, we simulate 6,840 leaks in total.

Our simulations focus on leaks with Tier 1 leaks for two reasons. First, we do not find substantial real-world Peerlock/Peerlock-lite protection of non-Tier 1 ASes as outlined in Section 4.4. Second, many consequential leaks are propagated globally over the Tier 1

backbone, e.g. [50, 45, 78, 22]. Some of our protection schemes will investigate whether leaks can propagate throughout the Internet without Tier 1 distribution.

4.5.2 Evaluation

We evaluate seven different protection schemes for Tier 1 leaks.

- **No filters.**
- **Inferred:** Tier 1 Peerlock levels observed during Internet measurements.
- **Full T1:** All Tier 1s Peerlock for all other Tier 1s.
- **Full T1 + large ISP lock:** Same as full T1, but all large ISPs Peerlock their Tier 1 peers.
- **Full T1 + large ISP lite:** Same as full T1, but all large ISPs deploy Peerlock-lite to protect clique ASes.
- **Full T1 + large ISP both:** Same as full T1, but all large ISPs deploy Peerlock-lite filters and Peerlock for their Tier 1 peers.
- **Inferred + large ISP lite:** Same as inferred, but all large ISPs deploy Peerlock-lite.

While it is simpler to filter customer-learned routes with Peerlock-lite than to deploy Tier 1 Peerlock filters for large ISPs, we include both Peerlock and Peerlock-lite filtering by these ASes to study how leaks are propagated within the topology. The results of these experiments are presented in Fig. 21, which displays both the proportion of ASes in the topology receiving leak updates (Fig. 21a), and the proportion selecting/exporting the leak path (Fig. 21b).

A critical feature revealed by Fig. 21 is the insufficiency of Tier 1 protection alone (blue lines). Full Tier 1 Peerlocking prevents all distribution of studied leaks over the peering clique, but leak updates still spread to the majority of the topology for most experiments. Adding large ISP Peerlock protection has a relatively significant impact on both propagation and installation.

Peerlock-lite deployment by these ASes (red lines) benefits from more filterers with wider protection per filterer. Naturally, these scenarios are much more effective at preventing propagation. For most leak cases, less than 10% of the topology receives leak updates. This highlights the leverage large ISPs have within the topology; filtering at these ASes (<1% of all networks) generates an extensive shielding effect. The distinct "shoulder" on the Peerlock-lite curves in Fig. 21b suggests the impact on ASes *using* the leak is even more pronounced. There is virtually no impact on target link usage for 75% of simulated leaks when Peerlock-lite is deployed by all large ISPs. Interestingly, the combination of Peerlock and Peerlock-lite filtering by large ISPs (green line) adds little value over Peerlock-lite alone.

4.5.3 Discussion

Path Encoding

To analyze how each of these scenarios shapes leak propagation (and route selection/export), we collect the AS PATH of all leaks exported during the above experiments. We use a novel *path encoding* whereby each AS on leak AS PATHs is converted to a 2-tuple with the form (relationship to next AS, UCLA class [64]). Only the AS PATH segment from the first customer to provider link to the leaker ASN - the *leak segment* - is encoded. This trimming discards the "down" segment prepended as leaks propagate within customer cones, as well as the segment connecting leaker and destination that is invariant across leaks. We include AS relationship in the encoding because of its importance in path export behavior as described in Section 2.1; UCLA class informs us regarding where leaks travel through the routing hierarchy. Taken together, these factors help us understand broadly the topological dynamics at play in leak propagation, and to capture the dominant leak propagation vectors under each protection scenario.

Relationship is encoded as "C" (customer), "R" (peer), or "P" (provider). UCLA classes are indicated by "T" (Tier 1), "L" (large ISP), "S" (small ISP), and "U" (stub). Example: [LR, TP] encodes a leak path exported to a Tier 1 provider by the leaker, who then passes the leak to a large ISP peer. The progress of the leak through the large ISP's customer cone

would continue to the left of "LR", and the path from leaker to destination would continue to the right of "TP", but these segments are omitted as explained above.

We will use two tables in analyzing our results. Table A depicts the three most common leak encodings for each scenario; these account for at least a quarter of leak paths regardless of filter placement. We also list the sum and percentage of ASes exporting leaks accounted for by each encoding. Table A gives summary statistics for leak segments, including their average length and the percentage of leak segments transiting each UCLA class. Because we do not encode customer cone propagation in leak segments, stubs are transited in <10% of paths across all protection scenarios, are omitted from the table.

First, we observe that even under the "no filters" scenario, leaks re-transiting the Tier 1 clique are not the most common path encoding in Table A. Table A shows they are present in fewer than 35% of leak segments under all scenarios. This result is an artifact of the BGP decision process; paths learned from customers are preferred over those exported by peers, and peer routes are favored over provider-learned ones. So, with all other selection criteria equal, routes exported from providers "above" an AS in the topology - e.g. the peering clique - will generally only be installed and exported if the AS has not received an update from peers/customers "below". Since Tier 1 providers cap the routing hierarchy, we expect ASes will prefer non-Tier 1 routes when provided alternatives by their connectivity. This dynamic explains why the additional protection afforded by complete Peerlock within the peering clique vs. current levels is muted in Fig. 21b.

This effect also brings large ISPs to the fore in our simulations. As noted in [64], these networks are densely connected with peering links. Their connectivity allows them to bypass the Tier 1 clique for many routes - and makes them the primary channel for leak propagation. The most common encoding for every scenario in Table A includes a large ISP, and 18/21 of the top encodings transit at least one. More than 70% of leak segments transit these ASes for all protection scenarios (see Table A). In fact, in the scenarios without Peerlock-lite (top four listed), leak segments on average transit - and could be filtered by - multiple large ISPs. These statistics motivate the scenarios that place Peerlock-lite filtering at these ASes (bottom three in tables).

Interestingly, Peerlock-lite diminishes leak usage and propagation unequally as shown in Fig. 21. Fig. 21a shows about 20% of leak segments propagate to 20% or more of the topology with large ISP Peerlock-lite deployment, but Fig. 21b shows that fewer than 5% are installed/exported by at least 20% of ASes. Table A hints at why this is the case - a third or more of leak segments in Peerlock-lite scenarios are exported to large ISP peers, who propagate them directly into their customer cones (indicated by [LR]). Large ISPs with any customer-learned or preferential (e.g. shorter) peer-learned paths to the leak destination will prefer their existing route, so the [LR] only includes a subset of the leaker’s peers. Large ISP peers advertising the leak to customers could reach many ASes, but as a provider-learned route, the leak will be disadvantaged in the BGP decision process.

We see in Table A and Fig. 21 that small ISPs do not have the connectivity to propagate leaks globally when the large ISP provider channel is blocked by Peerlock-lite. Under all scenarios, most leak segments do not transit a small ISP (though they may be transited during propagation into customer cones). This feature suggests a less prominent role in route exchange for these networks relative to large ISPs.

To summarize, we find large ISPs are the most critical players in halting the spread and installation of Tier 1 leaks. These networks are interconnected enough to globally disseminate route leaks *without the peering clique* in many cases. Moreover, adding simple Peerlock-lite filters at these ASes to the currently deployed Peerlock filters in the peering clique causes a 94% reduction in total leak export across 6,840 leak simulations. Table A suggests that peer connections among ISPs are the largest remaining vulnerability for Tier 1 leaks given uniform large ISP Peerlock-lite deployment. These channels are out of reach for Peerlock-lite as described, but could be mitigated by 1) additional peering relationships/Peerlock rules to protect important leak targets and/or 2) complementary leak prevention systems like IRR filtering.

4.6 Related Work

Smith et al.’s 2020 study on the efficacy of poison filtering for inbound re-routing [81] similarly employed the PEERING framework to probe the behavior of remote networks. That work

encountered some evidence for poison filtering, and noted that filtering rates increase with poisoned AS degree, but did not seek to describe the underlying filtering mechanism or measure which ASes filter poisons. Similarly, Birge-Lee et al. [6] and our own work in the prior chapter 3 trialed poisons as primitives for novel BGP attacks. We and the SICO authors both encountered filtering when attempting to poison large transit networks, but did not examine filtering position or prevalence in that work.

Hlavacek et al. [26] introduced the DISCO system for preventing BGP hijacking. While not designed to prevent route leaks, the approach taken by DISCO is ideologically similar to Peerlock - DISCO emphasizes deployability/usability at the expense of some security guarantees relative to RPKI/ROV filtering. This line of thinking is informed by a long history of glacial deployment rates for security features that harden BGP including BGPsec [88] and the RPKI [18, 62].

Previous work that relies on BGP poisoning often assumes 1) unpoisoned ASes will forward poisoned updates and 2) poisoned ASes will drop such updates (see Section 2.4). Katz-Bassett et al.’s LIFEGUARD fault detection and remediation system [37, 39], for instance, employs poisoning to steer ASes around link failures. Smith and Schuchard’s Nyx defense [82] depends on rerouting with poisons for DDOS/Link Flooding Attack mitigation. Anwar et al.’s path discovery technique [2] is also driven by BGP poisoning. While we discovered little evidence for general poison filtering, the prevalence of Tier 1/large transit network filtering could present an obstacle to these systems. Specifically, **the assumption that unpoisoned networks will propagate poisons does not hold in all cases.**

4.7 Conclusion

This work probes the current deployment of Peerlock/Peerlock-lite on the control plane with active Internet measurements in Section 4.4. We find substantial evidence for deployment of these leak defense systems, especially in large transit networks, and measure a rise in Peerlock deployment within the peering clique during our experiments. While the range of protected networks is still narrow within our observation window, with most filterers protecting only Tier 1 ASes, many of the most disruptive recent route leaks contain these

networks. Defensive systems [82, 39], measurement techniques [2], and attacks [6] that may poison Peerlock-protected networks could inadvertently trigger these filters, and must not assume poisons will be propagated by unpoisoned networks. BGP simulators should likewise account for the presence of Peerlock to faithfully reproduce control plane behavior.

We also examine how the position and prevalence of filtering impacts leak propagation in the AS-level topology in Section 4.5. Notably, we find that large ISPs filtering plays a major role in global leak dissemination, signaling that Tier 1 clique deployment of Peerlock alone is not sufficient to isolate leaks. Strategic placement of filters at these large transit providers, which account for fewer than 1% of all ASes, completely mitigates 80% of simulated Tier 1 route leaks.

The MANRS filtering guide encourages AS PATH filtering by member ISPs, particularly for screening customer advertisements, and gives Peerlock/Peerlock-lite as examples. But these systems are not explicitly required, unlike IRR filtering (see [57] Section 4.1.1.1). Given the many indirect/direct customers these networks serve, ISPs are best equipped and best incentivized to deploy effective filters. Moreover, neither Peerlock nor Peerlock-lite is technically complex or burdensome to configure. Therefore, we argue for broad application of these common-sense leak prevention techniques by ISPs as a meaningful step in securing inter-domain routing.

4.7.1 Operator Engagement

The preprint for the study behind this chapter was posted to the NANOG and RIPE operator mailing lists in June and August 2020. While operator response was limited, email correspondence around the study did yield some helpful insights. One European AS operator claimed that at least one (and possibly additional) transit networks deployed techniques similar to Peerlock around 2007, roughly ten years before NTT’s codification of the method [84]. Separately, a Tier 1 network operator suggested that 1) differences in network automation sophistication could account for observed uneven filtering within the peering clique and 2) defensive filtering may have partially mitigated the Verizon/Cloudflare leaks detailed in Section 1. This idea is supported by Cloudflare’s discussion on the incident [45]. Cloudflare identifies some networks that filtered the leaks (including ASes

1299 Telia, 2914 NTT, and 7018 AT&T). Bandwidth graphs presented in that post indicate little or no impact on Cloudflare-to-AT&T data plane operation, while Cloudflare traffic to leak propagator Verizon was drastically reduced for hours after the incident.

4.7.2 Future Directions

Widespread adoption of Peerlock will likely depend on addressing scalability issues. Rule configuration currently requires non-standard, manual out-of-band communication between protector/protected ASes. Automating this process is a crucial step in extending Peerlock beyond core networks. Communities designating authorized upstreams for routes, as proposed in [30], could take the place of out-of-band communication. Alternatively, RPKI registration of direct/indirect customers [31] could distribute trusted topological information relevant to filtering. In the next chapter, we present still another approach - RIB analysis for automated "peer locking".

Chapter 5

Automated Route Leak Protection for Core Networks

5.1 Introduction

As seen in the prior chapter, the absence of a ground-truth, global AS-level topology is central to the route leak problem. AS relationships are often considered business secrets and held private by operators; without complete knowledge of all AS relationships, leaks cannot be distinguished from valid routes with complete certainty. Many benign events, including link failure response, load balancing, and new AS business relationships may demonstrate the same rare export patterns seen in leaks.

The Peerlock system, introduced in Chapter 4, works to mitigate leaks by sharing topological information between peering ASes. Protected ASes communicate their authorized upstreams to protectors out of band; then, any route that transits the protected AS but arrives from a non-authorized advertiser can be effectively filtered by the protector. This pairing approach works to establish defenses between the largest ASes in the Internet’s core, but as we observed in Section 4.4, its manual configuration costs have largely limited its scope to protected Tier 1 ASes.

Unfortunately, as shown in Section 4.5, large ISPs can form a global channel for route propagation even without Tier 1 participation. This dynamic motivates the development of a simple, scalable approach to Peerlock-like route leak detection that can be applied broadly

by large ISPs. The broad disparities present in networks' sophistication, resources, and incentives for deploying security systems also dictates that our system function without the cooperation of smaller networks on the Internet's fringe (stubs).

In this chapter, we introduce Corelock, an automated, simple, and scalable system for defending against re-transmission of route leaks through the core. Section 5.3 describes the "core enclave" ideal that underlies both Corelock and Peerlock. We also describe there how Corelock builds minimally intrusive BGP route filters from CAIDA's public provider-peer observed customer cones [48] and relationship inferences. By combining well-known topological abstractions with a simple, relationship-based ruleset, we aim to mitigate core re-transmission leaks without intermediate network participation.

In Section 5.4.2, we demonstrate expected false positive rates for our system by gathering routes from BGP collectors at Internet Exchange Points (IXPs) in the ARIN (North America), LACNIC (Latin America/Caribbean), RIPE (Europe), AFRINIC (Africa), and APNIC (Asia/Pacific) regions. We apply Corelock "filters" between large transit networks on all observed routes, and record how often these routes violate Corelock rules. Overall, we achieve a 5% violation ratio - that is, 5% of analyzed routes violate Corelock rules as deployed between core ASes. Section 5.4.3 delves into why routes diverge from export expectations built from historical routing data. While our notional Corelock deployment is sensitive to correct and complete relationship inferences, deployers in practice will have full knowledge of their next hop relationships.

Section 5.4.2 also matches routes that violate Corelock to those deployers were previously using to reach violation destinations. We compare the performance-related characteristics visible on the control plane - prefix and path length - as well as economic considerations like next hop relationship. We find that violation routes are, on average, about half an AS-level hop shorter than existing routes. Further, violation routes are modestly more likely to transit settlement-free peering relationships: about 62% of violations arrived via peers versus 53% of prior routes. However, we also show that

In this chapter, we make the following key contributions:

- We give an overview of the automated Corelock filtering system in Section 5.3.

- We measure how often routes on the control plane depart from propagation patterns expected by Corelock’s source abstractions, with analysis of our results, in Section 5.4.
- In the same section, we compare the properties of Corelock-violating routes to those they replace.
- We consider how these results could change over time in Section 5.5.

5.2 Background

5.2.1 Provider-Peer Observed Customer Cones

In Section 2.2, we introduced the *recursive customer cone* abstraction. These cones are helpful for ranking an AS’s influence on the control plane, as well as understanding relationships between remote ASes. However, ASes often build complex relationships that blur traditional customer/provider/peer classification. For example, one AS may peer with another in some regions/points-of-presence, but serve as a provider for the same AS at another interconnection point (a *hybrid* relationship [20]). The customer cone’s recursive construction ignores these complexities and assumes ASes provide transit for all customers of downstream networks.

This simplification overestimates how many indirect customers an AS has, and limits the usefulness of the recursive customer cone as a predictor of route export behavior. The *provider-peer observed* or (*PPO*) customer cone [20] is a related abstraction designed to address these limitations. PPO customer cones are built empirically, with ASes’ direct/indirect customer sets derived via analysis of routes that networks export to their peers and providers, directly linking cone construction with observed policy decisions. For this reason, PPO cones are used when making likely filtering inferences in Section 4.4.

The PPO customer cone technique is limited by route visibility, an important factor given the incomplete distribution of public route collection points on the control plane [65]. However, as we observed in Section 4.4, collector placement (and, therefore, route visibility) is biased toward the Internet’s core. The route leak prevention system introduced later in

this chapter (in Section 5.3) relies only on these best observed networks' PPO customer cones.

5.3 The Corelock System

5.3.1 The Core Enclave Ideal

Internet-level networks vary greatly in sophistication, resources, and influence on the control plane. About 50 ASes have customer cones with greater than 1,000 networks, but the vast majority of registered networks (about 94% of all ASes) are stubs with 5 or fewer downstream customers. In this chapter, we will refer to the large ISPs (those with 50 or more customer cone members) and Tier 1 ASes as "core" networks. These ASes have been shown in prior work to exert an outsized influence on the control plane [18] and to form virtually an exclusive channel for global route propagation [55].

The concentration of routing influence within this relatively small set of networks is potentially dangerous to Internet resiliency [1], but it also represents a clear opportunity for securing the decentralized, heterogeneously administered control plane. The Peerlock system [84] discussed in Chapter 4 recognizes this dynamic in its design, which enables large transit provider partners to detect route leaks without the cooperation of intermediate networks. However, Peerlock's scalability is limited by its requirement for manual configuration between every pair of participating networks. As shown in Section 4.4, Peerlock deployment has not penetrated far beyond the Tier 1 peering clique that caps the routing hierarchy.

Core networks and their providers account for just 1.5% of all ASes. These networks could form an *enclave* preventing most external route leak propagation by uniformly applying a simple rule - reject all routes that leave and re-enter the enclave, as shown in Fig. 22. With this construction (a generalization of Peerlock), any route entering the enclave must arrive via a customer or peer of some enclave AS, which we will call the *entry point*. If the entry point learned about the new route from a peer, the entry point's export should be restricted to core and non-core customers. Conceptually, the route should only travel through core

customers and "outward". Since no non-enclave ASes have customers in the enclave, the route should not re-enter the enclave under the valley-free routing model. If the entry point instead received the route from a customer, we expect wider export to all of the entry point's connections, both within and outside the enclave. Still, the route could only exit the enclave by peering or customer links, so any re-entry to the enclave remains a valley-free routing violation. Moreover, customer-advertised routes should arrive from the entry point's own customer cone, where the entry point's topological visibility is greatest.

5.3.2 Corelock Design

While the enclave ideal is helpful for understanding how core ASes can prevent global route leak propagation, it is highly unrealistic given the historical deployment rates for BGP security extensions like BGPsec and RPKI/ROV. BGPsec still lacks any commercial implementation, and the fewer than a quarter of prefix/origin pairs have valid ROAs stored in the RPKI [62]. No governance structure exists to enforce a timely and uniform rollout of security enhancements, even within the Internet's core. Given this condition, we seek instead to design a system that is simple, scalable, and effective in mitigating globally disruptive Type 1-4 (topological) route leaks, even in partial deployment. Specifically, we seek to provide Peerlock-like protection between the roughly 650 core ASes able to spread routes globally without the need for Peerlock's manual rule configuration. Like the Peerlock system discussed in Chapter 4, non-topological leaks (like re-originations and full table spills, e.g. [50]) are outside the scope of this system; these other leak types are easily remedied by complementary security features like ROV and max-prefix filtering.

Here we present Corelock, a novel automated route leak defense system. Like Peerlock, Corelock is designed to mitigate leaks between *protector* and *protected* ASes. Corelock rules, however, are built from historical route analysis rather than manual configuration between participants. This method enables greater scalability, as protectors can establish protection rules for any remote AS without out-of-band communication. Filtering based on prior behavior also addresses problems with the RFC definition of a leak — advertisement beyond the advertiser's intended scope [86]. Since we cannot determine operator intent, we cannot designate advertisements as leaks with certainty. Our system uses historical behavior

as a proxy for present intent by 1) collecting routes ASes have advertised over a one month period and 2) using CAIDA’s PPO customer cones [48] built from this data to store export behavior for all observed ASes. For our system, a "leak" (or *violation*) is an advertisement that is topologically abnormal given prior month routing behavior. Of course, topological churn and other benign deviations from prior routing behavior will produce false positives in our system, so violations must be handled carefully.

Corelock addresses this potential problem by adopting violations when protectors lack alternative routes to the violation destination prefix rather than filtering (dropping) them. This approach is found in earlier work on mitigating anomalous routing events [36], and can be implemented in practice by lowering the LOCAL PREF attribute of violations to some minimum value rather than discarding them entirely. Falling back to violations in the absence of other routes limits Corelock’s intrusiveness by **ruling out any disconnections** between protector ASes and violation destinations that could result from filtering violations outright. However, our system’s false positives may well cause protectors to miss opportunities to install advantageous (e.g. less expensive or more performant) routes for "leaked" prefixes.

5.3.3 Corelock Protection Rules

Corelock protection rules are built from 1) classification of protector relationships with neighboring/next hop ASes, and 2) *provider-peer observed/PPO* customer cones of the protector, next hop, and protected ASes. PPO customer cones are rooted in observed AS policy and account for hybrid/regionalized relationships, as they are derived from routes exported to an AS’s provider/peers [48]. Specifically, an AS C is included in AS P ’s PPO customer cone if, during a periodic analysis of globally-distributed public RIBs, AS P is observed exporting routes through C to its providers and peers. Figure 23 demonstrates the three filter types that Corelock deploys install on neighboring connections.

Customer connections receive the strictest treatment. For updates arriving over these connections, any protected ASes present on the update’s route but absent from the protector’s customer cone will be classified as a violation. This rule follows from the valley-free routing model; we expect that the only ASes present on customer-learned routes will be indirect customers. Routes are flagged on **peer** connections if any protected ASes on the

route are missing from the peer’s customer cone, since peers should only exchange customer-learned routes. **Provider** connections are the most difficult to vet, as providers are expected to export virtually all selected routes to their customers. In this case, Corelock marks any routes with more than one AS between the deployer and the first protected AS on the path as violations. Clearly this prevents any intermediate route leaks; no matter the relationship between the protected AS and the single intermediate AS, the intermediate AS is expected to export the route to its customer. While this may seem overly restrictive, our system targets large transit providers, who have relatively little traffic over connections to providers. We show the relatively rarity of these provider-advertised violations in Section 5.4.

5.3.4 Incomplete Data and False Positives

Incorrect inferences or incomplete PPO customer cone source data are the primary sources of false positives in Corelock. In our evaluation of a notional Corelock deployment in the following section, we use CAIDA’s public monthly relationship and PPO datasets [98] to build protection rules. Luckie et al. validate CAIDA’s relationship inference algorithm by collecting validation data from operator submissions, Internet Routing Registry entries, and relationship data embedded as communities in some BGP updates. Overall, they validate about one-third of their inter-AS relationship inferences and find that roughly 95% of their inferences are consistent with validation data. However, the peer-peer or provider-customer inferences provided by CAIDA do not account for the complexity of AS relationships in practice; ASes commonly vary their role with neighboring ASes on a per-prefix or per-region basis [20]. These nuanced relationships are classified as provider-customer relationships by CAIDA’s inference algorithm [48], meaning that they are subjected to the strictest Corelock rules, which avoids propagation of potential leaks at the cost of additional false positives in our experiment. In practice, protector ASes have ground-truth knowledge of their own relationships with neighboring ASes; our notional false positive rate therefore represent an upper-bound rather than an estimate of rates likely to be seen in deployment. Similarly, incomplete PPO data will trigger violations (as protected ASes may be absent from providers’ inferred customers cones) at the expense of additional false positives.

5.3.5 Alternative Approaches

Many other choices exist for topological abstractions that model route propagation. Traditional customer cones could be used in place of PPO customer cones, but this approach would overstate customer cone size, potentially permitting some leaks. Similarly, BGP-observed customer cones [48] could be employed for this purpose. For some AS A , the BGP-observed customer cone of A includes all ASes C reachable over any number of provider to customer links from A on observed BGP paths. This provides a more accurate, policy-based estimate of customer cones than the traditional approach, but still includes the customers of A 's peers within A 's customer cone, artificially inflating its size and allowing more potential leaks. A leak detection approach based outside of customer cones, such as identifying valid upstreams/intermediate ASes between protector and protected ASes using route analysis, is also possible [79]. However, these approaches have not demonstrated acceptable false positive rates, and customer cones are a well-understood abstraction directly tied to expected route propagation.

5.4 Simulating Corelock Deployment

5.4.1 Methodology

To evaluate Corelock, we establish a notional bidirectional deployment between the roughly 650 core ASes in CAIDA's October 2020 AS relationship dataset [98] over a three month period, November 2020-January 2021. For each month, we download all routes arriving at five globally distributed RouteViews [99] and RIPE Routing Information Service [59] collectors using BGPStream [65]. Each collector is located at a distinct Regional Internet Registry (RIR) IXP; this placement allows us to cast a wide net, gathering routes from many regions for analysis. Table A shows registry and geographic location information for the route data sources in this experiment.

As we describe in the preceding section, Corelock flags routes as violations where PPO customer cone information is incomplete. So, our primary objective for this experiment is to discover the proportion of all routes marked as violations, the *violation ratio* for Corelock.

To this end, we analyze each route arriving at our collectors during the study period, and determine: 1) whether the route’s AS PATH violates expectations built from CAIDA’s PPO customer cones, and if at least one notional protector and protected AS are present on the path; and 2) whether Corelock *would have* marked the route as a violation. The first item helps us understand the completeness of CAIDA’s PPO customer cones, and the second provides an estimate for how frequently operators can expect the system to flag routes. Each Corelock violation is recorded, and routes arriving after the violation are checked to see if they *resolve* the violation - that is, if the later route demonstrates that the protector has switched onto a non-flagged path to the destination prefix. We "refresh" the PPO customer cones and relationship inferences used by Corelock every month to ensure the freshness of information provided to Corelock.

Further, we investigate the opportunity cost incurred by false positive classifications. While we lack the data-plane information necessary to fully quantify the *performance* difference between the paths networks use previous to and after installing violating routes, we will compare the prefix and path length of violation routes and the routes installed previously by protectors for the same destination. We also endeavour to understand the *economic* cost of downgrading violations’ LOCAL PREF. These downgrades may incur monetary expenses when deployers flag less expensive routes (e.g., those transiting customers) and are forced to install routes transiting peers - or worse, providers.

This investigation of costs is conducted after the initial experiment gather routes flagged by Corelock. For each violation at the RouteViews Equinix collector, *rveqix*, we search the most recent RIB dump for the collector where the violation was discovered in an attempt to locate the route the protector was previously using to reach the violation destination. We look both for direct matches, as well as less-specific (supernets) of the violation destination prefix. Such a matching is not always present in RIB dumps; the presence of the prior-used route depends on the export behavior of all ASes between protector and collector.

5.4.2 Evaluation

In total, we analyzed nearly 15 billion BGP updates, 98.6% of which transit at least one notional deployer (core AS). 2.8 billion routes (19.6%) violated export expectations built

from CAIDA PPO customer cones for at least one on-path AS (orange bars). Our Corelock "deployers" marked 750 million updates as violations, achieving an **overall violation ratio of 5.1%** (red bars). Importantly, operators' full, contractual knowledge of their next hop relationships means this number represents an upper bound rather than a precise estimate of Corelock's expected false positive rate. Figure 24 shows our results split by collector. The per-collector violation ratio ranged from a low of 2.5% at the *rrc01* collector in London to a high of 13.5% at Sao Paulo-based *rvsp*.

In Figure A we display a breakdown of the inferred relationships between the protector and its next hop relationship (the advertising neighbor) for each violation at the North American collector, *rveqix*. **Nearly 80% of all violations arrived via peers**, while just 6% were advertised to the protector by its customers. This feature is consistently observed at all other collectors, as well, and meets our expectations as we will address in the following discussion. Provider-delivered violations - which occur whenever a provider delivers a route that transits at least one intermediate ASes before reaching another core AS - were responsible for less than one in six violations.

We found prior route matches (supernet or exact) for 16% of violation destinations during our follow-on investigation into the routes protectors were using before receiving violation updates (*prior routes*). The prefix distribution of prior and violation routes match almost exactly as shown in Figure A, and closely mirror overall Internet prefix distribution statistics as reported by monitoring projects like Potaroo [28]. Figure A shows the proportion of prior and violation routes learned from inferred providers, peers, and customers of protector ASes for which matches were found. While the next hop relationship distribution shown here does not match the distribution of recorded violations, it does follow the distribution of BGP updates seen arriving at core ASes during the study period at this collector. The relationship distribution of matches between violation and There is a clear shift from provider-learned prior routes to cheaper, peer-learned violations; this means that **downgrading violation LOCAL PREFs will likely have some economic cost to deployers**.

This economic cost is informed both by next hop relationship differences between routes, as well as the duration of time violations are installed. Figure A shows the cumulative

distribution of violation times for routes observed at the *rveqix* collector, November 2020-January 2021, in cases when our framework allowed us to observe a switch. In this experiment, we only track violations for resolution over 24 hours; exceeding this time limit is infeasible with our current hardware memory constraints. Roughly half (56%) of routes were resolved within this window. As shown in Figure A, nearly 90% of violations are transient (last less than 30 minutes). Performance costs are harder to measure, as control plane path characteristics are insufficient alone to determine data plane latency. However, we did see a significant difference in path length between prior routes and violations. Prior routes were 6.4 hops in length on average, compared to 5.9 hops in violation routes. Both of these numbers are higher than the average path length reported by control plane measurement project Potaroo [28], which hovers around 4.5 hops.

5.4.3 Discussion

It is reasonable to ask whether the violations observed during in this section were the result of erroneous routing events (true leaks) or false positives. Our analysis reveals that violations are highly concentrated on certain links; for instance, the majority of violations (18m/134m, or 51%) of violations observed at *rveqix* occurred at just 5 links between 16 ASes. Similar concentrations were observed at other collectors. Moreover, violation rates are spread evenly across the observation window rather than clustered at certain points. These results indicate that it is highly likely that the overwhelming majority of recorded violations are false positives, since true route leaks would travel across many links and demonstrate violation "hot spots" in time. The extreme concentration of violations on certain links (but not a single advertiser) also suggests either incorrect/incomplete relationship inferences are behind many false positives, meaning that ASes in practice (who know their own neighboring relationships completely) may encounter many fewer false positives in practice.

The variance in violation ratio observed at difference collectors is a surprising result from our study. At *rveqix*, just 4.2% of routes triggered Corelock, while *rvsp* saw a violation ratio of 13.5%. Here violations are likewise concentrated on a few link relationships. However, at this location the top 5 links connect only 4 ASes. Brazil's academic backbone, AS1916 National Education and Research Network, advertised the overwhelming majority

of observed violations at *rvsp* (about 140m/144.5m or 97%) to its peers. It is difficult to orthogonalize the effects of relationship misclassification and incomplete customer cones, but it is likely that complex relationships play some role, since many violation routes are observed month after month. CAIDA's PPO customer cones source data from this same collector, meaning that incomplete customer cone information would be corrected over time, and is unlikely to be responsible for such persistent violations. Overall, our results indicate that a more complex model for AS relationships (including hybrid relationships) is likely necessary to reduce system false positives to an acceptable level for deployment.

Across collectors, peer links represented between 74% and 91% of all violations. This result meets our expectation, because core ASes connections are dominated by peering links, and core ASes hear more updates from peers than from providers and customers. At *veqix*, for instance, core ASes in our study received 47% of updates from peers, 28.5% from providers, and under a quarter from customers. Moreover, provider and customer-learned violation designations are dependent upon only core AS customer cone information, but flagging a peer route requires knowledge of the peer's customer cone. Some peers will lie outside the core, where our ability to observe updates (and build complete customer cones) is limited. Given these factors, and that Corelock marks routes as violations when customer cone information is incomplete, it is not unusual that the majority of violations are received over peering connections.

Our efforts to quantify the cost of Corelock are limited by our ability to fully observe control plane traffic. Just 16% of violations were matched to prior routes in RIBs. While we observed clear movement from provider-learned prior routes to peer-learned violations, the cost of "missing" peer-learned routes may be mitigated by the transient nature of many violations.

5.5 Relevant Historical Trends

5.5.1 Stability of the Internet’s Core Membership

Corelock is predicated on core ASes monitoring the presence of remote core ASes on routes to detect leaks, as these backbone ASes can distribute routes globally. So, changes to core AS membership over time will change the filtering requirements for deployers and the breadth of the systems’ protection as notionally deployed. In total, 671 ASes were observed with customer cones over size 50 during the course of our study period (November 2020 - January 2021). About 91% of these remained in the set for the entire study period, with just 61 membership edits during this period. For a wider view, we compare core AS sets in November of 2017 and 2020. Over this longer period, 681 ASes reached core designation, with 257 total edits, and has grown by 20% from 540 to 652 ASes.

5.5.2 Longitudinal Evaluation of Violation Rate

While we did not observe significant changes in violation rates during our three month observation window, it is possible that longer term trends in the AS-level topology could challenge the feasibility of the Corelock system. To address this possibility, we repeat our RIB analysis experiment from Section 5.4 at the RouteViews Equinix collector with data from November of 2017, 2018, and 2019. The results of this year-over-year comparison is shown in Figure 29. While the total number of updates is variable given control plane conditions, IXP membership levels, and other factors, the proportion of updates flagged for Corelock violations falls under 6% in every month examined. No clear positive or negative trends were observed in violation rates over time.

5.6 Related Work

BGP security research often emphasizes deployability/usability at the expense of provable security guarantees. This line of thinking is informed by a long history of glacial design and deployment rates for formal security extensions for BGP, like BGPsec [88] and the RPKI [18, 62]. For example, Karlin et al.’s Pretty Good BGP (PGBGP) [36], Sermpenzi et

al.’s ARTEMIS [77] and Hlavacek et al.’s DISCO [26] hijacking mitigation systems, and Subramanian et al.’s Listen and Whisper [92] system all enable local network operators to detect and/or respond to routing problems without global participation or a robust trust infrastructure. Unfortunately, all of the above work is designed and evaluated primarily for protection against BGP hijacks, not route leaks.

Siddiqui et al.’s recent route leak detection (RLD) technique [79], like our proposed work, aims to provide operators with a practical technique for filtering likely leaks without the technical burden of a trust infrastructure or the challenge of inter-AS cooperation. RLD unfortunately only leverages local BGP data, and is therefore ineffective in some critical situations, such as detecting peer-advertised leaks. Active control plane probing is suggested as a remedy for these deficiencies, but we assess that operators are unlikely to send bogus messages to peering ASes. Our approach fuses outside data sources with an AS’s local relationship knowledge to build a less intrusive system.

5.7 Conclusion

In this chapter, we presented the Corelock system and evaluated its notional deployment across roughly 650 large transit providers. A number of key findings stand out from this work.

First, we have shown that meaningful, Peerlock-like filters can be built in an automated fashion from historical routing data. Nearly 80% of all updates we observed on the control plane are consistent with our export model, which is built from CAIDA’s published AS relationship inferences and provider-peer observed customer cones. Future work could apply a more sophisticated model of AS relationships - like the one found in [20] - to route leak detection.

Next, we note that the assumption that networks behave homogeneously across regions is dangerous. Like [20], who noted European ASes were more likely to form complex relationships, we observed varied behavior at different geographic locations on the control plane. Our false positive rate varied from less than 3% in London to over 13.5% at a South American collector. ARIN ASes make up just 21% of all core ASes, but have twice the

average customer cone size of any other region. Future systems should account for this regionalism in their design and evaluation to produce meaningful results.

It is important to note that neither Peerlock nor Corelock prevent all route leaks. Max-prefix, ROV, and other complementary filtering systems should be applied to combat non-topological route leaks. Even with wide Corelock deployment, there is still a danger from pre-core leaks within core AS customer cones. For these, ASes must monitor their own customer cones, by enforcing up-to-date IRR entries or using automated detection systems like RLD [79].

Finally, We further observe that our work, in this and other chapters, is greatly challenged by the lack of public vantage points on the control plane. Exposing more ground-truth data on AS relationships and routes is critical to advancing BGP research. Both relationship inferences and customer cones are only as good as their input data, and both more and better data is needed to improve the quality and completeness of these abstractions.

Chapter 6

Future Direction and Conclusion

6.1 Future Work

6.1.1 Routing-Enabled DDoS

Our work in Chapter 3 demonstrates that a routing-capable adversary can manipulate attack flows using poisoned control plane announcements. Poisoning is a broadly effective tool for limiting path export at the AS level [81], but leak prevention systems and other filters limit poison propagation through the peering clique [55]. The SICO attack [6] demonstrated the viability of BGP communities as an alternative export control mechanism. While poisons represent a simple, direct approach, the communities supported by many transit providers (e.g. AS2914 NTT [63]) give downstream operators a broad toolset of supported path-export control options. These options include regional restrictions and even neighbor-specific export limitations. The Maestro and Leak attacks described in Sections 3.4 and 3.6, respectively, could use these more nuanced path control primitives to great effect. For instance, an attacker could limit the visibility of attack announcements for greater stealth, especially given that communities are often stripped from updates after their effect is applied.

Our attacks use only traditional botnets as flow sources; reflection attacks, e.g. [52] or other malicious flow sources could likely be steered with the same approaches. These alternate sources could have a very different topological and/or geographic distribution relative to the major botnets evaluated in our attack, which are highly concentrated in a

handful of networks. Our live Internet demonstration "attacks" in Section 3.5, however, are driven by widely dispersed flow sources, suggesting that this is not an obstacle to successfully executing Maestro.

Maestro is demonstrated with a single prefix and target link; it is likely the attack could be extended to multiple, strategically placed links to effect an "Internet blockade" as in [35]. Such a broad attack might use multiple prefixes or BGP communities to separately steer flows from disparate parts of the Internet, funneling traffic over major points of connection into/out of a target region or installation. Given the critical importance of the Internet in modern society as the channel for financial markets and other critical activities, such an attack could have devastating consequences even if rapidly mitigated.

6.1.2 Mitigating Leaks

Chapters 4 and 5 give practical tools to transit providers seeking to insulate their networks from topological route leaks. The primary challenge in arresting route leaks is the mismatch between information available to operators - their own local routing "neighborhood" - and the scope of control plane routes, which span the entire Internet. Increasing the visibility of direct/indirect AS relationships, e.g. by registering customer cones in the RPKI [31] or embedding more topological information in communities [30], would greatly simplify this challenge.

A parallel can be drawn between the route leak detection challenge and the primary challenge in BGP research - the well-known AS topology incompleteness problem. The control plane is global, with a complex and ever-shifting hierarchy of ASes working under independent control to connect hosts over the Internet. Without ground-truth knowledge of AS relationships (that includes policy, not merely connectivity), BGP research is limited to simulations, conjectures, and inferences about AS policy and thus, the behavior of the control plane at large.

Resources like BGPStream [65] provide critical points of observation for research, but as shown in Chapter 5, making sweeping statements about the control plane from limited points of observation can produce misleading results. Network policy in BGP is extremely flexible, and the AS relationship structure varies from region to region. However, incentivizing ASes

to expose their control plane traffic could be difficult, as this reveals information about their business relationships without providing a clear benefit (in narrow terms) to the network itself. Nevertheless, locating additional control plane perspectives is vital to advancing BGP security research as a field.

The definition of "route leaks" is also problematic from an academic perspective, since it relies on unobservable (or difficult to observe) notions like operator intent. Future work could examine disruptive routing events more broadly, perhaps by associating control plane abnormalities with observed disruptions on the data plane using tools like CAIDA's Internet Outage and Detection Analysis (IODA) tool [8]. This fusion of data plane and control plane events could reduce the ambiguity associated with "route leaks" and serve as a productive avenue for future work on this topic.

6.2 Conclusion

This work began by demonstrating the threat posed by malicious actors who gain routing capability in Chapter 3 - such an adversary can manipulate the control plane to expose links to unintended traffic and/or direct malicious botnet flows with novel control. This chapter, as well as numerous highly disruptive routing failures in recent years [40, 22, 103, 78, 95, 50, 51], demonstrate that both malicious and unintentional route leaks pose a present threat to the proper functioning of the Internet. Our work in Chapters 4 and 5, on the other hand, demonstrate that network operators can deploy pragmatic solutions to limit their exposure to these events and protect themselves and their customers. Taken together, these ideas demonstrate our thesis: *Malicious and unintentional route leaks threaten Internet availability, but pragmatic solutions can mitigate their impact.*

Global route leaks and hijacks demonstrate the inherent fragility of an Internet built on a control plane that lacks cryptographic hardening of routes and other essential security features. It is important to note that these events are occurring within an environment of broad cooperation between network operators; that is, both leaks and hijacks are almost universally the result of misconfiguration rather than hostile intent. In general, operators across polities are working together to preserve a functioning global network. While

geopolitical foes have frequently used the Internet as a medium for delivering and defending against cyberattacks [27], attacks against the Internet infrastructure *itself* are less well-known. BGP has been weaponized for targeted hijacks, e.g. [16], but we do not know of any direct attempts at cyber-sabotage of the control plane. We have shown in Chapter 3 that adversaries can exploit routing capability to expose targets and channel attacks; it is certain that advanced persistent threats could and would leverage the control plane to pursue their objectives.

At the same time, the number of actors with routing control is ballooning rapidly. In November of 2017, the control plane was composed of about 59,000 ASes; at the time of this writing (2021), there are over 70,000 such networks. This growth introduces additional complexity and increases the opportunity for mistakes and misconfiguration. It is encouraging that security features like RPKI-based Route Origin Validation [62] are gathering steam to address the threat posed by hijacks, additional work is needed to build resiliency against leaks and other adverse routing events into the control plane. Next generation architectures like SCION [102] provide a roadmap for building a secure-by-design Internet control plane, but it is unclear when (or if) these technologies will replace BGP for directing flows between networks. In the meantime, pragmatic, common-sense solutions can - and should - be deployed to secure the critical infrastructure directing the world's digital traffic.

Bibliography

- [1] Acharya, H., Chakravarty, S., and Gosain, D. (2017). Few Throats to Choke: On the Current Structure of the Internet. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE. 61
- [2] Anwar, R., Niaz, H., Choffnes, D., Cunha, I., Gill, P., and Katz-Bassett, E. (2015). Investigating Interdomain Routing Policies in the Wild. In *Proceedings of the Internet Measurement Conference*. 6, 13, 31, 34, 55, 56
- [3] Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., and Teixeira, R. (2006). Avoiding Traceroute Anomalies with Paris Traceroute. In *ACM SIGCOMM*. 22, 23
- [4] Basescu, C., Reischuk, R., Szalachowski, P., Perrig, A., Zhang, Y., Hsiao, H., Kubota, A., and Urakawa, J. (2016). SIBRA: Scalable Internet Bandwidth Reservation Architecture. 29
- [5] Bellovin, SM and Gansner, ER (2003). Using Link Cuts to Attack Internet Routing. *AT&T Labs Research*. 31
- [6] Birge-Lee, H., Wang, L., Rexford, J., and Mittal, P. (2019). SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 25, 27, 30, 34, 45, 47, 55, 56, 73
- [7] Bush, R., Maennel, O., Roughan, M., and Uhlig, S. (2009). Internet Optometry: Assessing the Broken Glasses in Internet Reachability. In *Proceedings of the Internet Measurement Conference*. 48
- [8] CAIDA (2021). Internet Outage Detection and Analysis Tool. <https://ioda.caida.org/>. 75
- [9] Chang, W., Mohaisen, A., Wang, A., and Chen, S. (2015). Measuring Botnets in the Wild: Some New Trends. In *ACM Computer and Communications Security*. 14
- [10] Choi, J., Park, J., Cheng, P., Kim, D., and Zhang, L. (2011). Understanding BGP Next-hop Diversity. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 6

- [11] Cisco (2017). Cisco IOS and IOS XE Software Cluster Management Protocol Remote Code Execution Vulnerability. <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20170317-cmp>. 15
- [12] Cisco (2018). Cisco IOS XE Software AAA Login Authentication Remote Code Execution Vulnerability. <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20180606-aaa>. 15
- [13] Cisco (2019). Cisco REST API Container for IOS XE Software Authentication Bypass Vulnerability. <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20190828-iosxe-rest-auth-bypass>. 15
- [14] Colitti, L., Di, D. and others (2006). Internet Topology Discovery Using Active Probing. 31
- [15] Dainotti, A., Squarcella, C., Aben, E., Claffy, K., Chiesa, M., Russo, M., and Pescapé, A. (2011). Analysis of Country-wide Internet Outages Caused by Censorship. In *ACM SIGCOMM*. 15
- [16] Demchak, C. and Shavitt, Y. (2018). China’s Maxim - Leave No Access Point Unexploited: The Hidden Story of China Telecom’s BGP Hijacking. *Military Cyber Affairs*. 15, 76
- [17] Gao, L. (2001). On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*. 5, 35, 40, 42, 50
- [18] Gilad, Y., Cohen, A., Herzberg, A., Schapira, M., and Shulman, H. (2017a). Are We There Yet? On RPKI’s Deployment and Security. In *The Network And Distributed Security Symposium*. 35, 36, 48, 49, 55, 61, 70
- [19] Gilad, Y., Sagga, O., and Goldberg, S. (2017b). Maxlength considered harmful to the rpki. In *Proceedings of the International Conference on Emerging Networking EXperiments and Technologies*. 38

- [20] Giotsas, V., Luckie, M., Huffaker, B., and Claffy, K. (2014). Inferring Complex AS Relationships. In *Proceedings of the Internet Measurement Conference*. 6, 60, 64, 71
- [21] Giotsas, V. and Zhou, S. (2012). Valley-free Violation in Internet Routing—Analysis Based on BGP Community Data. In *IEEE International Conference on Communications (ICC)*. IEEE. 27
- [22] Goodin, D. (2018). Google Goes Down after Major BGP Mishap Routes Traffic Through China. <https://arstechnica.com/information-technology/2018/11/major-bgp-mishap-takes-down-google-as-traffic-improperly-travels-to-china/>. 2, 7, 33, 45, 51, 75
- [23] Google Security and White Ops (2016). The Hunt for 3ve. https://services.google.com/fh/files/blogs/3ve_google_whiteops_whitepaper_final_nov_2018.pdf. 15
- [24] Guo, H. and Heidemann, J. (2018). Detecting ICMP Rate Limiting in the Internet. In *The Passive and Active Measurement Conference*. 21
- [25] Herzog, S. (2011). Revisiting the Estonian Cyber Attacks: Digital Threats and Multinational Responses. *Journal of Strategic Security*. 12
- [26] Hlavacek, T., Cunha, I., Gilad, Y., Herzberg, A., Katz-Bassett, E., Schapira, M., and Shulman, H. (2019). DISCO: Sidestepping RPKI’s Deployment Barriers. In *The Network and Distributed Systems Security Symposium*. 4, 6, 36, 55, 71
- [27] Huang, K., Siegel, M., and Madnick, S. (2018). Systematically Understanding the Cyber Attack Business: A Survey. *ACM Computing Surveys (CSUR)*, 51(4). 76
- [28] Huston, G. (2020). AS65000 BGP Routing Table Analysis Report. <http://bgp.potaroo.net/as2.0/bgp-active.html>. 67, 68
- [29] Huston, G. and Michaelson, G. (2012). RFC 6483 - Route Origin Validation. 38
- [30] J. Snijders, J. H. and Cisco (2017). BGP Route Leak Protection Community. <https://www.ietf.org/archive/id/draft-heitz-idr-route-leak-community-00.txt>. 39, 57, 74

- [31] J. Snijders, M. S. and Aelmans, M. (2020). RPKI Autonomous Systems Cones. <https://tools.ietf.org/html/draft-ietf-grow-rpki-as-cones-02>. 57, 74
- [32] Jacobson, V. (2020). Traceroute Man Page. <https://linux.die.net/man/8/traceroute>. 16, 21
- [33] Jennings, R. (2018). BGP Hijack Steals AWS IP range; Cryptocurrency Theft Ensues. <https://techbeacon.com/security/bgp-hijack-steals-aws-ip-range-cryptocurrency-theft-ensues>. 15
- [34] Kang, M., Gligor, V., and Sekar, V. (2016). SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks. In *The Network and Distributed Systems Security Symposium*. 29
- [35] Kang, M., Lee, S., and Gligor, V. (2013). The Crossfire Attack. In *2013 IEEE Symposium on Security and Privacy*. 9, 10, 12, 13, 16, 21, 31, 74
- [36] Karlin, J., Forrest, S., and Rexford, J. (2006). Pretty Good BGP: Improving BGP by Cautiously Adopting Routes. In *Proceedings of the 2006 IEEE International Conference on Network Protocols*. IEEE. 63, 70
- [37] Katz-Bassett, E., Choffnes, D., Cunha, I., Scott, C., Anderson, T., and Krishnamurthy, A. (2011). Machiavellian Routing: Improving Internet Availability with BGP Poisoning. In *Proceedings of the ACM Workshop on Hot Topics in Networks*. 55
- [38] Katz-Bassett, E., Madhyastha, H., Adhikari, V., Scott, C., Sherry, J., van Wesep, P., Anderson, T., and Krishnamurthy, A. (2010). Reverse Traceroute. *USENIX Symposium on Networked Systems Design and Implementation*. 31
- [39] Katz-Bassett, E., Scott, C., Choffnes, D., Cunha, I., Valancius, V., Feamster, N., Madhyastha, H., Anderson, T., and Krishnamurthy, A. (2012). LIFEGUARD: Practical Repair of Persistent Route Failures. *ACM SIGCOMM Computer Communication Review*, 42(4). 8, 13, 16, 31, 34, 55, 56

- [40] Kephart, N. (2015). Finding and Diagnosing BGP route leaks. <https://www.thousandeyes.com/blog/finding-and-diagnosing-bgp-route-leaks/>. 2, 7, 33, 40, 75
- [41] Kuerbis, B. and Mueller, M. (2017). Internet Routing Registries, Data Governance, and Security. *Journal of Cyber Policy*, 2(1). 2, 34, 37, 48
- [42] Lee, S., Kang, M., and Gligor, V. (2013). CoDef: Collaborative Defense Against Large-scale Link-flooding Attacks. In *ACM International Conference on Emerging Networking EXperiments and Technologies*. 29
- [43] Lepinski, M. and Kent, S. (2012). RFC 6480 - An Infrastructure to Support Secure Internet Routing. <https://tools.ietf.org/html/rfc6480>. 2, 36
- [44] Lepinski, M. and Sriram, K. (2013). RFC 8205 - BGPsec Protocol Specification. *Internet Engineering Task Force*. 37
- [45] Levy, M. (2019). The Deep-dive Into How Verizon and a BGP Optimizer Knocked Large Parts of the Internet Offline Monday. <https://blog.cloudflare.com/the-deep-dive-into-how-verizon-and-a-bgp-optimizer-knocked-large-parts-of-the-internet-offline-monday/>. 2, 7, 33, 40, 45, 51, 56
- [46] Liaskos, C. and Kotronis, V. and Dimitropoulos, X. (2016). A Novel Framework for Modeling and Mitigating Distributed Link Flooding Attacks. In *IEEE INFOCOM*. 29
- [47] Litke, P. and Stewart, J. (2014). BGP Hijacking for Cryptocurrency Profit. 15
- [48] Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., et al. (2013). AS Relationships, Customer Cones, and Validation. In *Proceedings of the Internet Measurement Conference*. 59, 63, 64, 65
- [49] Madory, D. (2017a). Large BGP Leak by Google Disrupts Internet in Japan. <https://dyn.com/blog/large-bgp-leak-by-google-disrupts-internet-in-japan/>. 1
- [50] Madory, D. (2017b). Widespread Impact Caused by Level 3 BGP Route Leak. <https://medium.com/oracledevs/widespread-impact-caused-by-level-3-bgp-route-leak>

widespread-impact-caused-by-level-3-bgp-route-leak-internet-intelligence-3dbd724d9ac5
2, 7, 37, 45, 51, 62, 75

- [51] Madory, D. (2014). Why Far-Flung Parts of the Internet Broke Today. <https://dyn.com/blog/why-the-internet-broke-today/>. 2, 7, 75
- [52] Majkowski, M. (2018). Memcrashed-Major Amplification Attacks from UDP port 11211. 9, 73
- [53] McDaniel, T., Smith, J., and Schuchard, M. (2020a). The Maestro Attack: Orchestrating Malicious Flows with BGP. In *International Conference on Security and Privacy in Communication Systems*. Springer. 3
- [54] McDaniel, T., Smith, J., and Schuchard, M. (2020b). The Maestro Attack: Orchestrating Malicious Flows with BGP. In *Proceedings of the EAI International Conference on Security and Privacy in Communication Networks (SecureComm)*. 34
- [55] McDaniel, T., Smith, J., and Schuchard, M. (2021). Flexsealing BGP Against Route Leaks: Peerlock Active Measurement and Analysis. *The Network and Distributed Systems Security Symposium*. 3, 32, 61, 73
- [56] Merit Network Inc (2020). List of Routing Registries. <http://www.irr.net/docs/list.html>. 36
- [57] Mutually Agreed Norms for Routing Security (2020a). MANRS Filtering Guide. <https://www.manrs.org/isps/guide/filtering/>. 40, 48, 56
- [58] Mutually Agreed Norms for Routing Security (2020b). MANRS Website. <https://www.manrs.org/>. 48
- [59] NCC, R. (2020). Routing Information Service. <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris>. 43, 65
- [60] Netlab360 (2017). Mirai Scanner. <http://data.netlab.360.com/mirai-scanner/>. 14
- [61] Network Working Group (1999). IRR RPSL Reference Guide. <https://tools.ietf.org/html/rfc2622>. 38

- [62] NIST (2020). RPKI Deployment Monitor. <https://rpki-monitor.antd.nist.gov/>. 36, 55, 62, 70, 76
- [63] NTT Communications (2021). NTT Routing Policies and Procedures. <https://www.gin.ntt.net/support-center/policies-procedures/routing/>. 73
- [64] Oliveira, R. and Pei, D. and Willinger, W. and Zhang, B. and Zhang, L. (2010). The (In) completeness of the Observed Internet AS-level Structure. *IEEE/ACM Transactions on Networking*. 6, 28, 44, 48, 52, 53
- [65] Orsini, C., King, A., Giordano, D., Giotsas, V., and Dainotti, A. (2016). BGPStream: A Software Framework for Live and Historical BGP Data Analysis. In *Proceedings of the Internet Measurement Conference*. 36, 60, 65, 74
- [66] Pepelnjak, I. (2009). Limit the Maximum BGP Path Length. http://wiki.nil.com/Limit_the_maximum_BGP_AS-path_length. 18, 30
- [67] Putman, C., Nieuwenhuis, J., et al. (2018). Business Model of a Botnet. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE. 15
- [68] Ravi, N., Shalinie, S., and Theres, D. (2020). Balance: Link flooding attack detection and mitigation via hybrid-sdn. *IEEE Transactions on Network and Service Management*. 29
- [69] Rekhter, Y., Li, T., and Hares, S. (1995). RFC 4721 - A Border Gateway Protocol 4. <https://tools.ietf.org/html/rfc4271>. 1
- [70] RIPE NCC (2019). RIPE NCC IRR Database Non-Authoritative Route Object Cleanup. <https://www.ripe.net/publications/docs/ripe-731>. 37
- [71] Robert, L., Michael, J., and Conway, T. (2016). Analysis of the Cyber Attack on the Ukrainian Power Grid. *USA: Electricity Information Sharing and Analysis Centre (E-ISAC)*. 12

- [72] Schlamp, J., Holz, R., Jacquemart, Q., Carle, G., and Biersack, E. (2016). HEAP: Reliable Assessment of BGP Hijacking Attacks. *IEEE Journal on Selected Areas in Communications*, 34(6). 6
- [73] Schlinker, B., Arnold, T., Cunha, I., and Katz-Bassett, E. (2019). PEERING: Virtualizing BGP at the Edge for Research. In *Proceedings of the International Conference on Emerging Networking Experiments And Technologies*. 22, 36, 43
- [74] Schuchard, M., Mohaisen, A., Foo Kune, D., Hopper, N., Kim, Y., and Vasserman, E. (2010). Losing Control of the Internet: Using the Data Plane to Attack the Control Plane. In *ACM Computer and Communications Security*. ACM. 9, 10, 13, 31, 50
- [75] Schuchard, M. and Geddes, J. and Thompson, C. and Hopper, N. (2012). Routing Around Decoys. In *The ACM Conference on Computer and Communications Security*. 13, 50
- [76] Scott Sr, J. and Summit, W. (2016). Rise of the Machines: The Dyn Attack Was Just a Practice Run December 2016. 9
- [77] Sermpezis, P., Kotronis, V., Gigis, P., Dimitropoulos, X., Cicalese, D., King, A., and Dainotti, A. (2018). ARTEMIS: Neutralizing BGP Hijacking Within a Minute. *IEEE/ACM Transactions on Networking*, 26(6). 6, 71
- [78] Shapelez, A. (2020). This is How You Deal with Route Leaks. <https://habr.com/en/company/qrator/blog/495260/>. 2, 7, 45, 51, 75
- [79] Siddiqui, M., Montero, D., Yannuzzi, M., Serral-Gracià, R., Masip-Bruin, X., and Ramírez, W. (2014). Route Leak Detection Using Real-time Analytics on Local BGP Information. In *2014 IEEE Global Communications Conference*. IEEE. 4, 65, 71, 72
- [80] Smith, J., Birkeland, K., McDaniel, T., and Schuchard, M. (2020a). Withdrawing the BGP Re-Routing Curtain: Understanding and Analyzing the Security Impact of BGP Poisoning through Real-World Measurements. *The Network and Distributed Systems Security Symposium*. 13, 31, 32

- [81] Smith, J., Birkeland, K., McDaniel, T., and Schuchard, M. (2020b). Withdrawing the BGP Re-Routing Curtain: Understanding the Security Impact of BGP Poisoning through Real-World Measurements. In *The Network and Distributed Systems Security Symposium*. 6, 8, 34, 42, 43, 44, 45, 47, 48, 54, 73
- [82] Smith, J. and Schuchard, M. (2018). Routing Around Congestion: Defeating DDoS Attacks and Adverse Network Conditions via Reactive BGP Routing. In *IEEE Symposium on Security and Privacy (SP)*. IEEE. 8, 13, 16, 29, 31, 34, 50, 55, 56
- [83] Snijders, J. (2016a). Everyday Practical BGP Filtering. <https://peerlock.net>. 34, 38, 44
- [84] Snijders, J. (2016b). NTT Peer Locking. http://instituut.net/~job/peerlock_manual.pdf. 3, 4, 24, 31, 34, 38, 40, 44, 56, 61
- [85] Snijders, J., Heasley, J. and Schmidt, M. (2017). Use of BGP Large Communities. <https://tools.ietf.org/html/rfc8195>. 37
- [86] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and Dickson, B. (2016). RFC 7908 - Problem Definition and Classification of BGP Route Leaks. <https://tools.ietf.org/html/rfc7908>. 1, 6, 42, 50, 62
- [87] Sriram, K. and Azimov, A. (2020). Methods for Detection and Mitigation of BGP Route Leaks. <https://tools.ietf.org/pdf/draft-ietf-grow-route-leak-detection-mitigation-02.pdf>. 37
- [88] Sriram, K. and Montgomery, DC (2019). Resilient Interdomain Traffic Exchange: BGP Security and DDoS Mitigation. *NIST*. 55, 70
- [89] Staff, RN (2015). RIPE Atlas: A Global Internet Measurement Network. *IP Journal*. 22
- [90] Strickx, T. (2019). How Verizon and a BGP Optimizer Knocked Large Parts of the Internet Offline Today. <https://blog.cloudflare.com/>

[how-verizon-and-a-bgp-optimizer-knocked-large-parts-of-the-internet-offline-today/](#).

2

- [91] Studer, A. and Perrig, A. (2009). The Coremelt Attack. In *European Symposium on Research in Computer Security*. Springer. 9, 10, 12, 13, 16, 21, 31
- [92] Subramanian, L., Roth, V., Stoica, I., Shenker, S., and Katz, R. (2004). Listen and Whisper: Security Mechanisms for BGP. In *USENIX Symposium on Networked Systems Design and Implementation*, volume 4. Citeseer. 71
- [93] Team Cymru (2019). Team cymru ip to asn lookup. <https://www.team-cymru.com/IP-ASN-mapping.html>. 21, 23
- [94] Thomas, M. and Mohaisen, A. (2014). Kindred Domains: Detecting and Clustering Botnet Domains Using DNS Traffic. In *The World Wide Web Conference*. 14
- [95] Toonk, A. (2015). Massive Route Leak Causes Internet Slowdown. <https://bgpmon.net/massive-route-leak-cause-internet-slowdown/>. 2, 75
- [96] Tran, M., Kang, M., Hsiao, H., Chiang, W., Tung, S., and Wang, Y. (2019). On the Feasibility of Rerouting-based DDoS Defenses. In *IEEE Symposium on Security and Privacy*. 6, 13, 16, 18, 30, 44, 45, 50
- [97] UCSD-CAIDA (2020a). CAIDA AS Ranking. <https://asrank.caida.org/>. 6, 29
- [98] UCSD-CAIDA (2020b). CAIDA AS Relationship Dataset. <https://www.caida.org/catalog/datasets/as-relationships/>. 13, 24, 28, 43, 45, 46, 50, 64, 65
- [99] University of Oregon (2020). RouteViews Project. <http://www.routeviews.org>. 43, 65
- [100] Wählisch, M., Maennel, O., and Schmidt, T. (2012). Towards Detecting BGP Route Hijacking using the RPKI. *ACM SIGCOMM Computer Communication Review*, 42(4). 6
- [101] Yan, H., Oliveira, R., Burnett, K., Matthews, D., Zhang, L., and Massey, D. (2009). BGPmon: A Real-time, Scalable, Extensible Monitoring System. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*. IEEE. 6

- [102] Z., X., Hsiao, H., Hasker, G., Chan, H., Perrig, A., and Andersen, D. (2011). SCION - Scalability, Control, and Isolation on Next-Generation Networks. *IEEE Symposium on Security and Privacy*. 76
- [103] Zmijewski, E. (2014). Indonesia Hijacks the World. <https://dyn.com/blog/indonesia-hijacks-world/>. 2, 7, 75

Appendix

A Figures

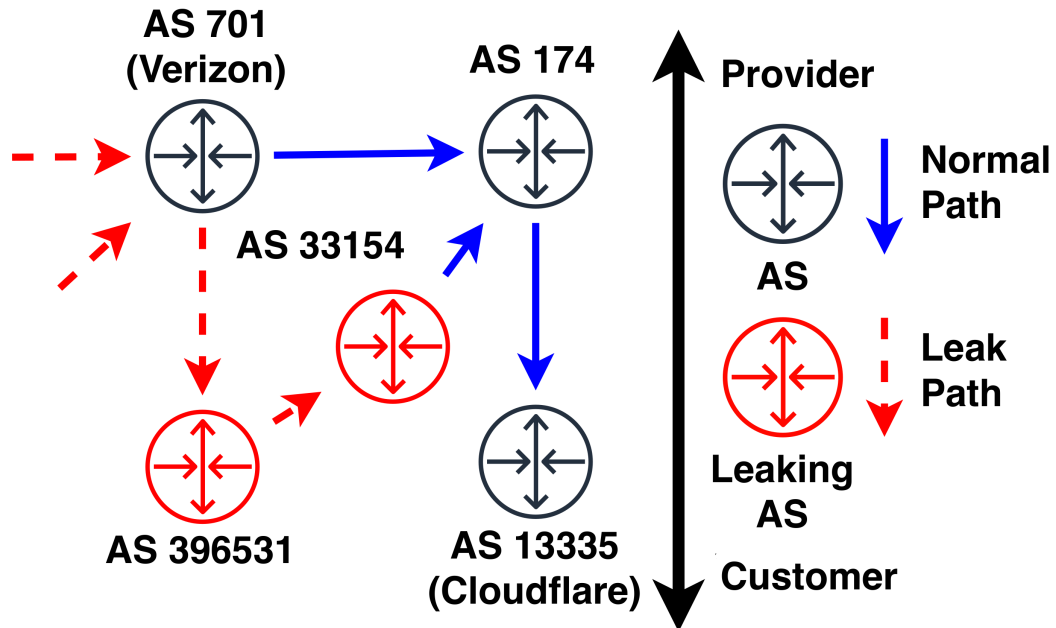


Figure 1: 2019 Verizon/Cloudflare leak. Other destination services were also affected.

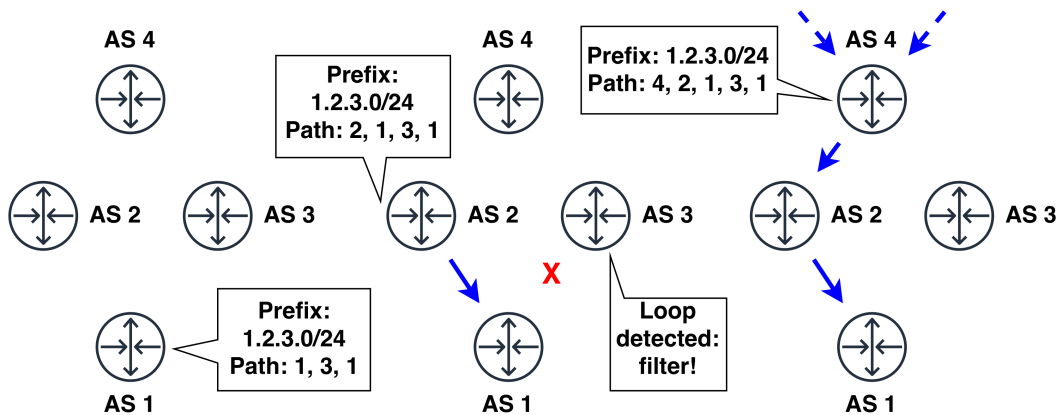


Figure 2: BGP poisoning. AS 1 originates a route with AS 2 prepended to path; AS 2 filters the update, but AS 3 propagates.

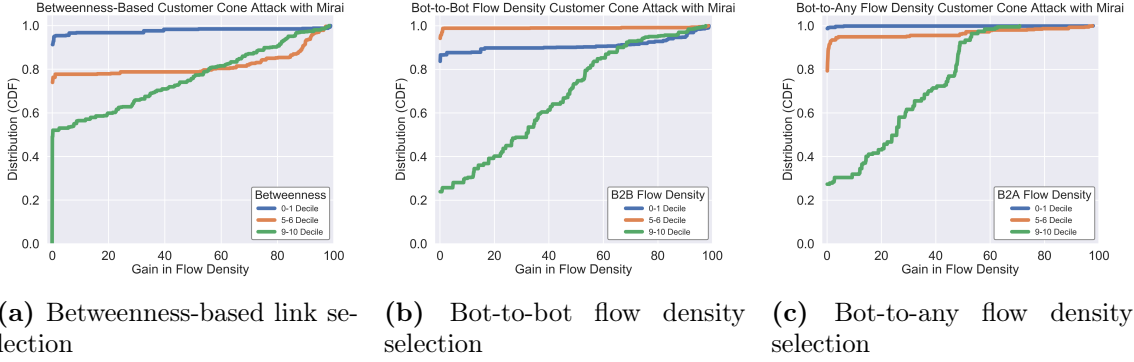


Figure 3: Flow density gain results (post-attack density - pre-attack density) by link selection strategy, **Mirai** botnet model

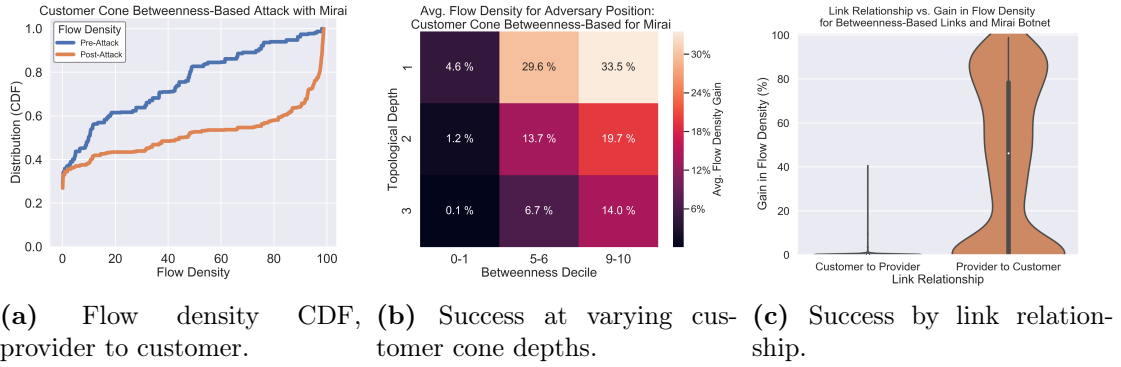


Figure 4: Deeper look at the customer attack success, betweenness link sample

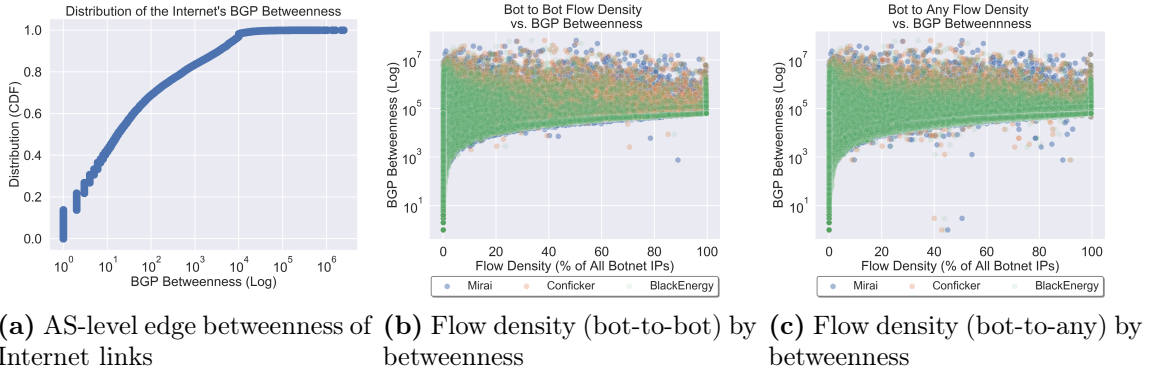


Figure 5: LFA vulnerability as function of betweenness

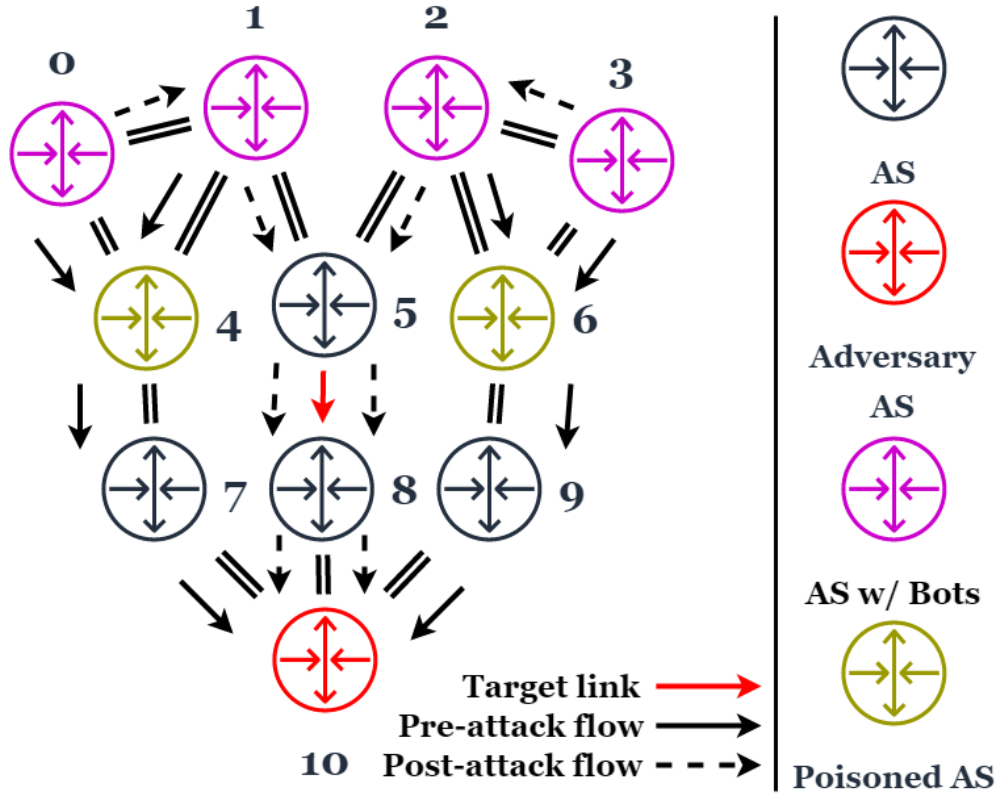


Figure 6: Maestro: BGP poisons collapse botnet traffic onto a target link.

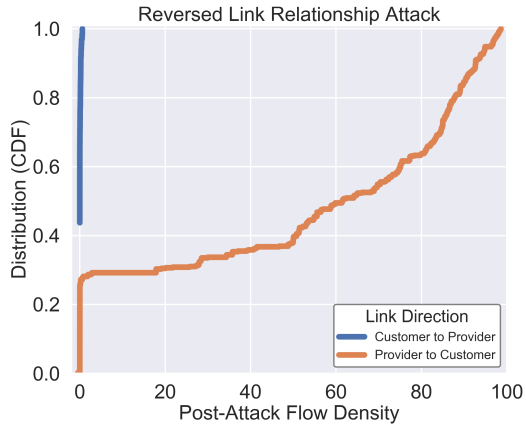


Figure 7: Reversing relationship to attack cust. to prov. links.

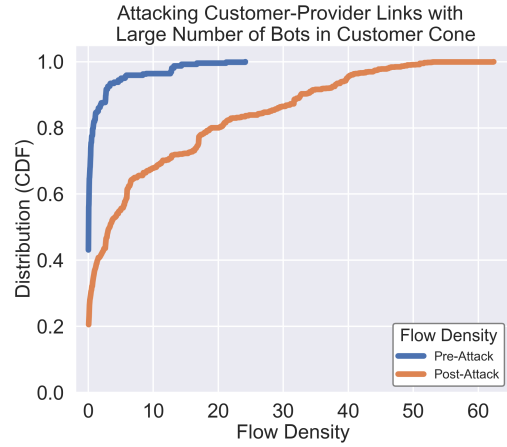


Figure 8: Attacking cust. to prov. links.

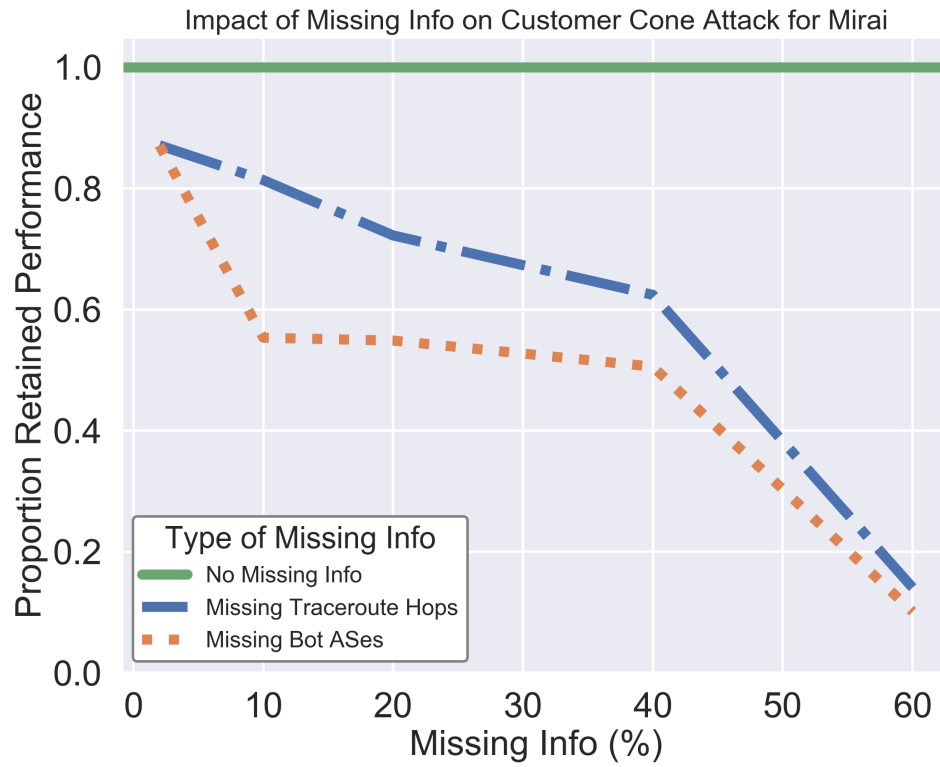


Figure 9: Maestro performance when missing traceroute hops and bot routes compared to complete information

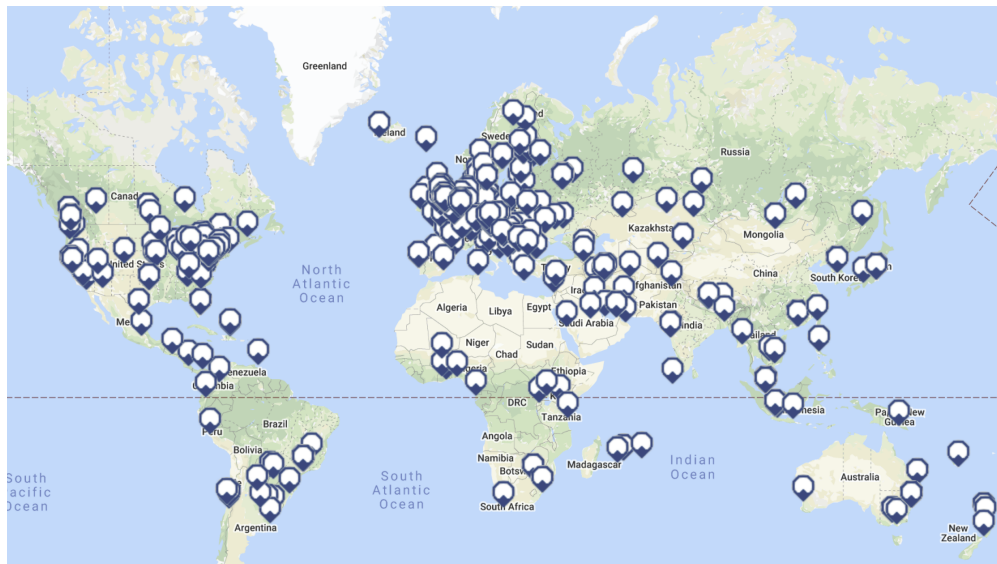


Figure 10: Notional "botnet" distribution.

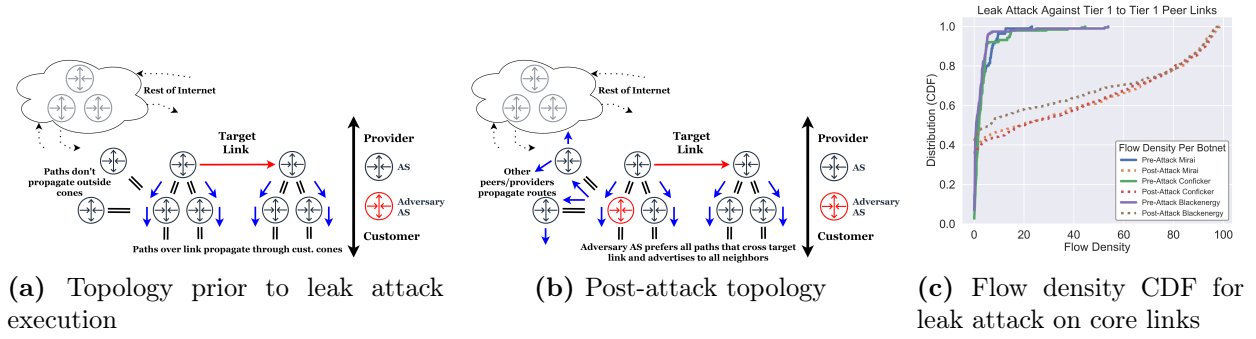


Figure 11: Valley-path leak attack overview and results

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

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

Vuln. Threshold	Vuln. Before			Vuln. After			Avg Adversaries			Stddev Adversaries		
	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

Table 3: Common leak filtering systems.

System	Coverage	Requirements	Notes
RPKI/ROV	Type 5 internal leak, Type 6 re-origination	RPKI trust infrastructure, local ROA registration & remote ROV checks	Type 5 coverage depends on optional ROA max length
IRR Filtering	Potentially all leak types; depends on stored policy object specificity.	Correct, fresh policy objects & derived filters along potential leak path	Quality issues, misaligned incentives
Max Prefix	All leak types	Filter with meaningful max prefix limit somewhere on potential leak path	Only effective when many prefixes are leaked

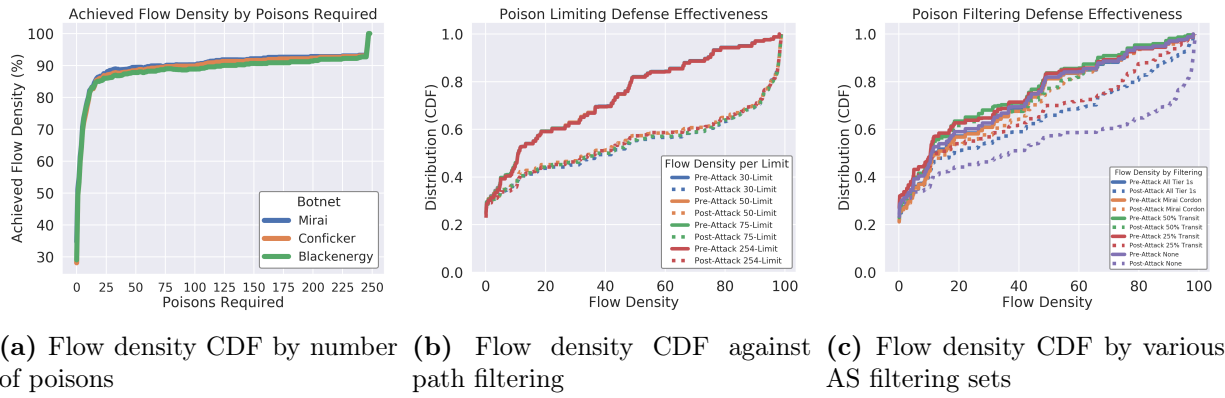


Figure 12: Evaluation of Maestro defenses

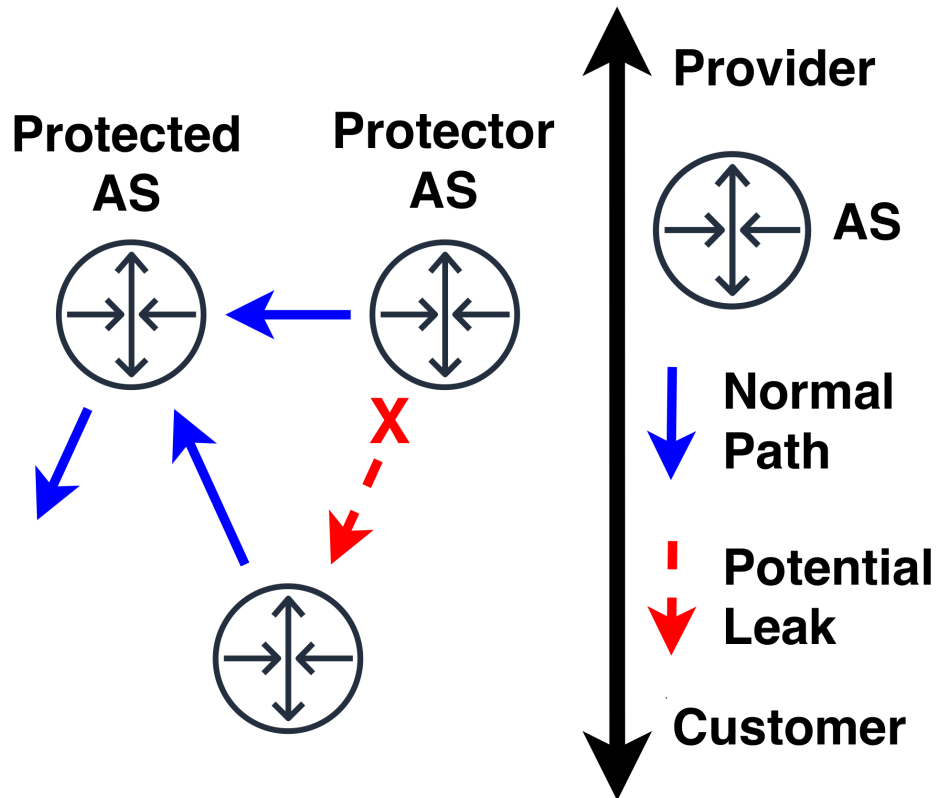


Figure 13: Simple Peerlock deployment. Protector AS filters updates containing the peer Protected AS from unauthorized propagators.

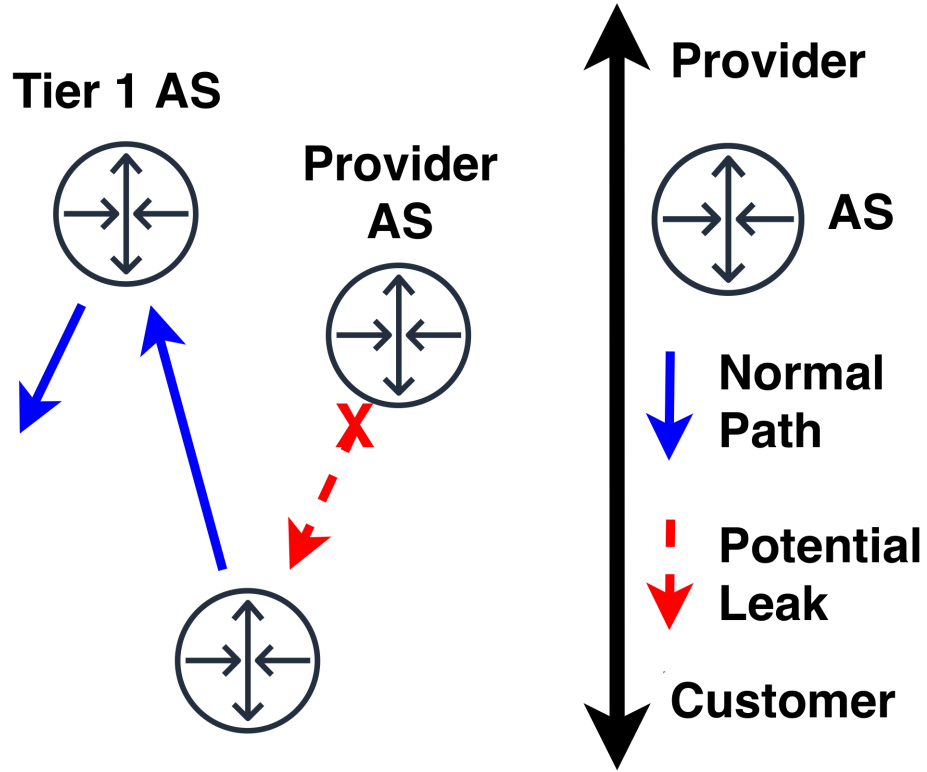


Figure 14: Example Peerlock-lite deployment. Provider AS filters updates from its customer that include a Tier 1 AS.

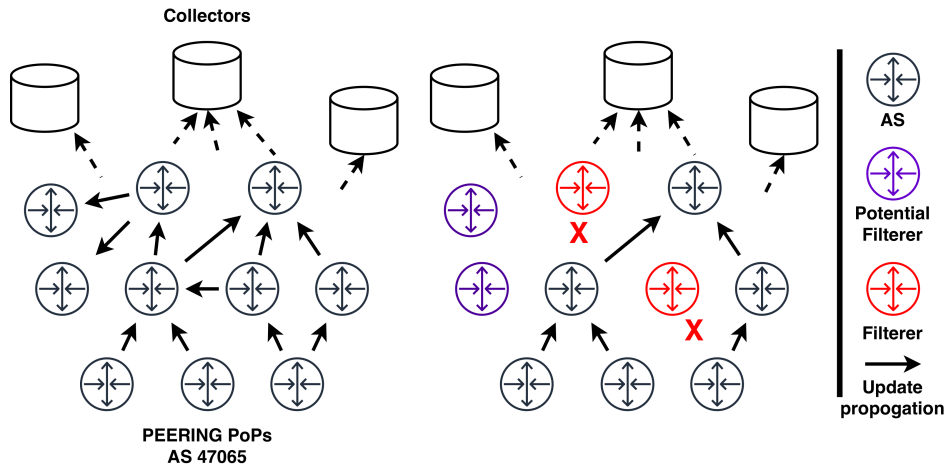
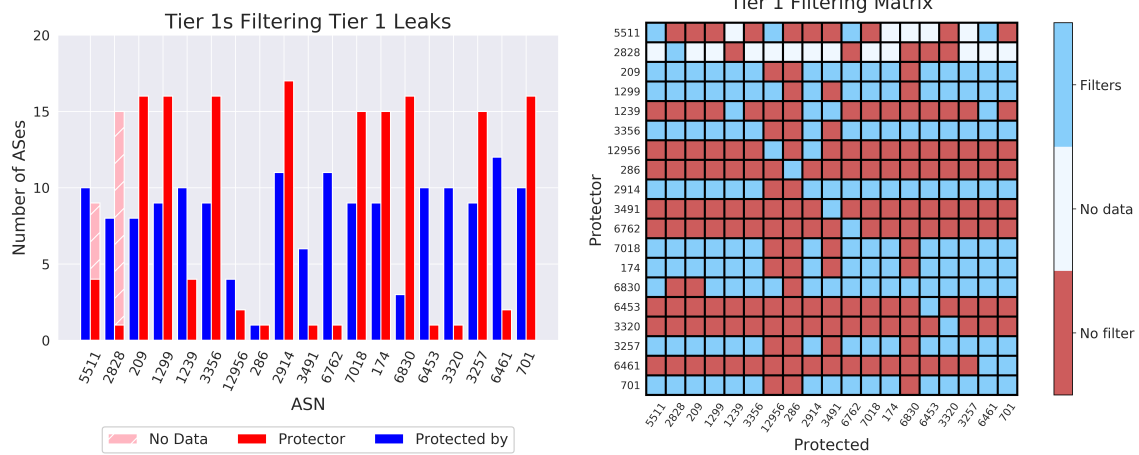
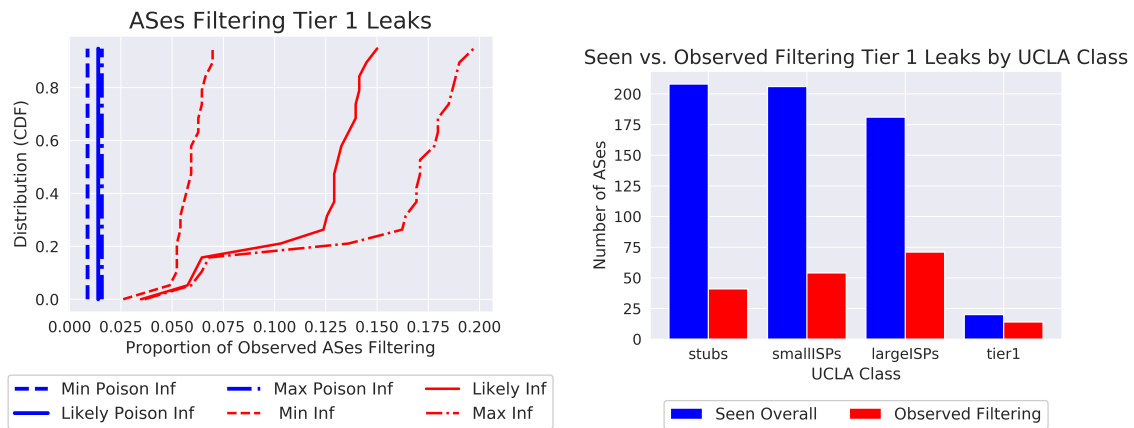


Figure 15: Measurement experiment depiction. Inferences are made about Peerlock deployment based on differences between normal updates (left) vs. poisoned updates (right) arriving at collectors.



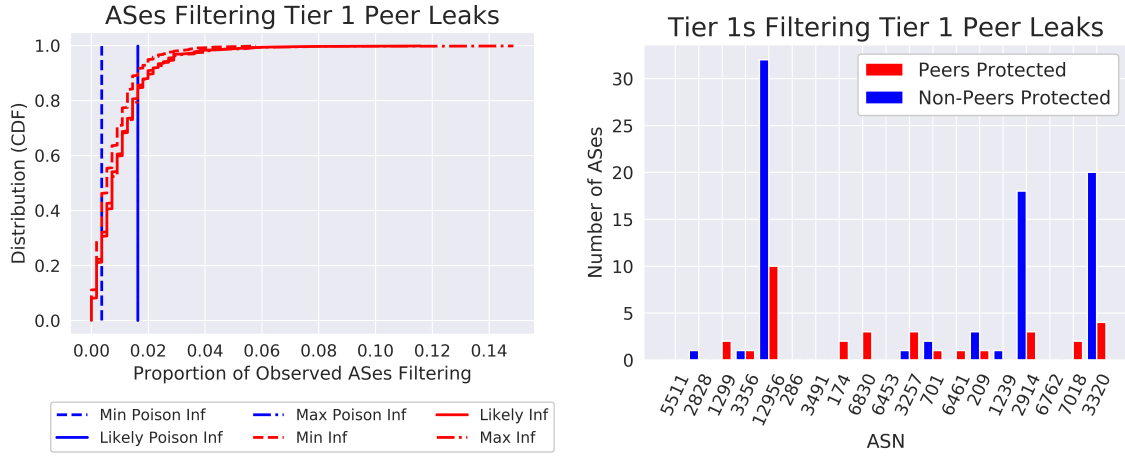
(a) Number of protector/protected rules by ASN. Protector numbers include ASes protecting their own ASN via loop detection. (b) Depiction of Tier 1 protection rules.

Figure 16: Tier 1s filtering Tier 1 leaks, 2019/2020 measurements.



(a) Blue lines show poison filtering; red lines depict Tier 1 leak filtering. (b) Blue bars show no. ASes in observation window; red bars show no. ASes filtering at least 1 Tier 1 leak.

Figure 17: Overall filtering of Tier 1 leaks, 2019/2020 measurements.



(a) Overall filtering levels for Tier 1 peer leaks. (b) Tier 1 filtering of Tier 1 peer leaks (peers Max and likely poison inferences match for this within clique excluded). set.

Figure 18: Tier 1 peer leaks, 2019/2020 measurements.

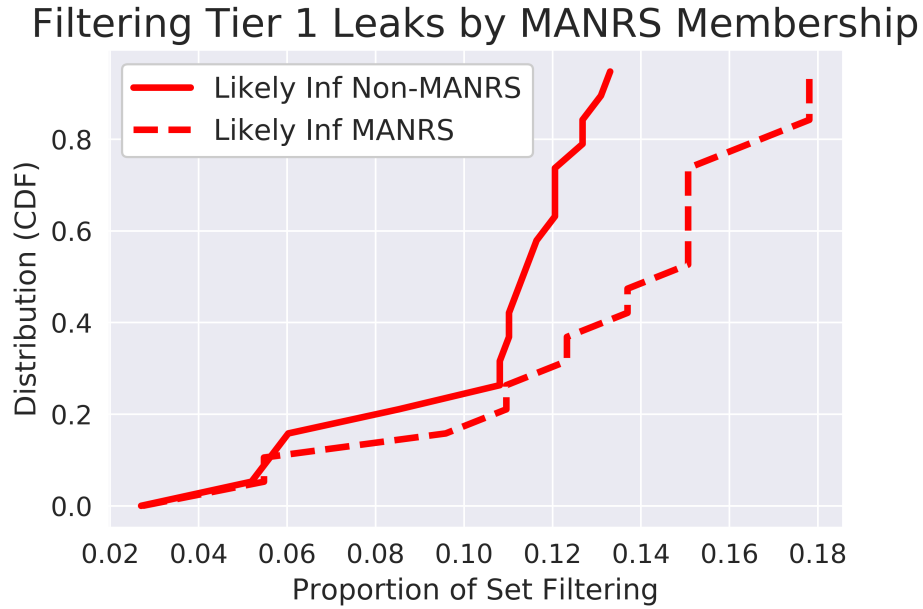


Figure 19: Tier 1 leak filtering for MANRS/non-MANRS ASes.

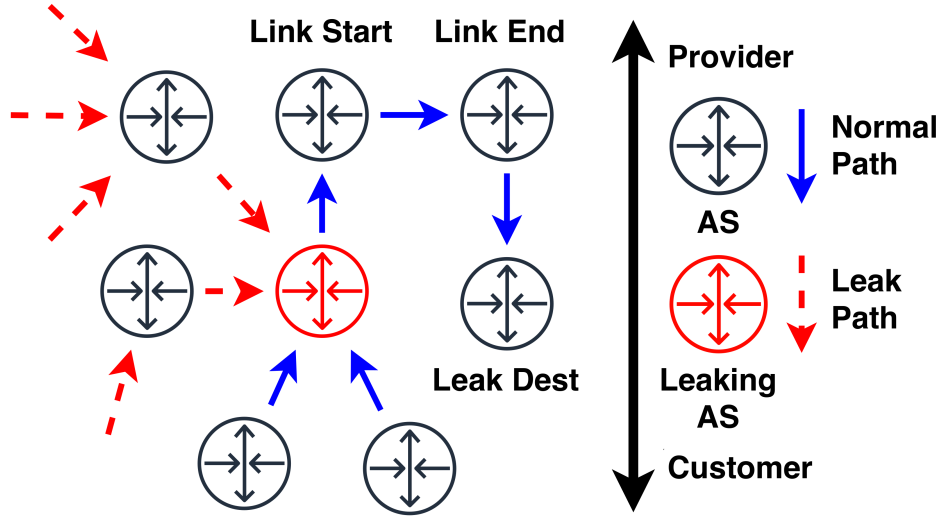
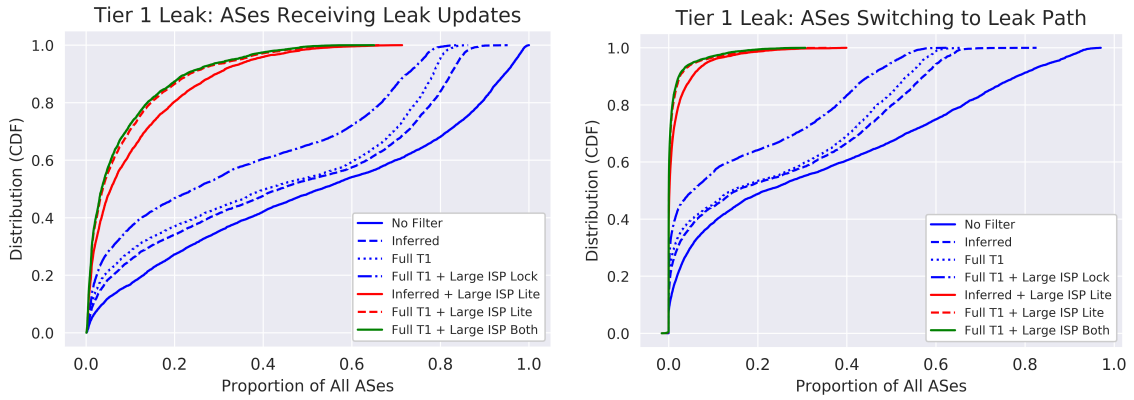


Figure 20: Example simulated leak. Dashed red lines indicate route leak to other providers/peers.



(a) Impact of various deployment scenarios on leak update propagation. (b) Note increased Peerlock-lite performance for path switching vs. leak update propagation.

Figure 21: Peerlock/Peerlock-lite simulation results.

Table 4: Most common encodings with number and percentage of ASes exporting leaks.

Common Leak Segment Encodings		
Scenario/Encoding	No. exporting ASes	% of exporting ASes
No filters	141,797,992	100%
[LR, LP]	14,892,311	11%
[TP]	10,254,707	7%
[LR, LP, LP]	8,683,968	6%
Inferred	108,030,704	100%
[LP, LP]	14,325,960	13%
[LR, LP, LP]	8,675,841	8%
[SR, LP]	5,169,427	5%
Full T1	101,024,444	100%
[LR, LP]	14,246,024	14%
[LR, LP, LP]	8,978,175	9%
[SR, LP, LP]	5,163,786	5%
Full T1 + large ISP lock	69,638,282	100%
[LR, LP]	9,473,820	14%
[LR, LP, LP]	5,899,779	8%
[SR, LP]	3,310,842	5%
Inferred + large ISP lite	8,005,724	100%
[LR]	2,537,276	32%
[LR, TP]	1,281,620	16%
[LR, SP]	653,167	8%
Full T1 + large ISP lite	5,215,232	100%
[LR]	2,386,597	46%
[LR, SP]	679,076	13%
[SR]	412,399	8%
Full T1 + large ISP both	4,649,828	100%
[LR]	2,023,579	44%
[LR, SP]	584,124	13%
[SR]	407,661	9%

Table 5: Analyzing leak segments by UCLA classes transited.

Scenario	Transited AS Statistics											
	Segment Length			Tier 1s			Large ISPs			Small ISPs		
	average	std. dev	% paths	average	std. dev	% paths	average	std. dev	% paths	average	std. dev	
No filters	4.4	1.8	27%	0.2	0.4	89%	2.3	1.6	40%	0.7	0.8	
Inferred	4.6	1.8	7%	0.5	0.2	98%	2.5	1.5	45%	0.8	0.9	
Full T1	4.7	1.8	0%	0.0	0.0	98%	2.6	1.5	46%	0.8	0.9	
Full T1 + large ISP lock	4.8	2.1	0%	0.0	0.0	98%	2.7	1.6	50%	0.9	1.0	
Inferred + large ISP lite	2.9	1.0	35%	0.3	0.5	72%	0.6	0.5	42%	0.8	1.0	
Full T1 + large ISP lite	2.7	1.1	0%	0.0	0.0	74%	0.6	0.5	47%	1.0	1.1	
Full T1 + large ISP both	2.7	1.1	0%	0.0	0.0	71%	0.6	0.5	49%	1.0	1.1	

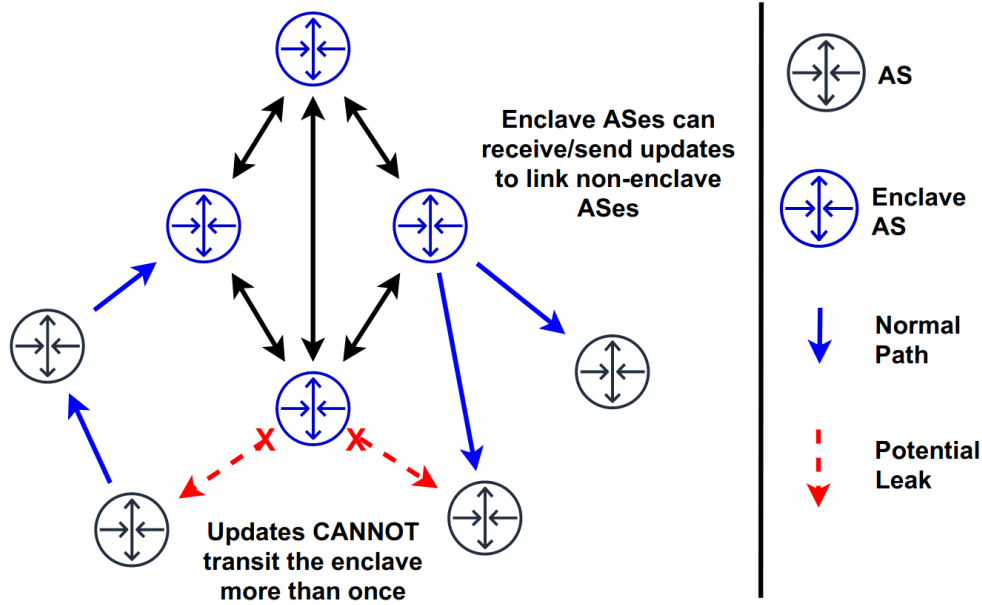


Figure 22: The "core enclave" concept - routes only permitted to transit the enclave once.

Table 6: Route Collectors in Corelock Study.

Corelock Experiment Collectors				
Collector Name	Location	Service	RIR	Region
rrc01	LINX/LONAP (UK)	RIS	RIPE	Europe
rrc19	NAPAfrica (South Africa)	RIS	AFRINIC	Africa
rvsg	SG1 Equinix (Singapore)	RouteViews	APNIC	Asia/Pacific
rveqix	Equinix (US)	RouteViews	ARIN	North America
rvsp	PTT Metro/NIC.br (Brazil)	RouteViews	LACNIC	Latin America/Caribbean

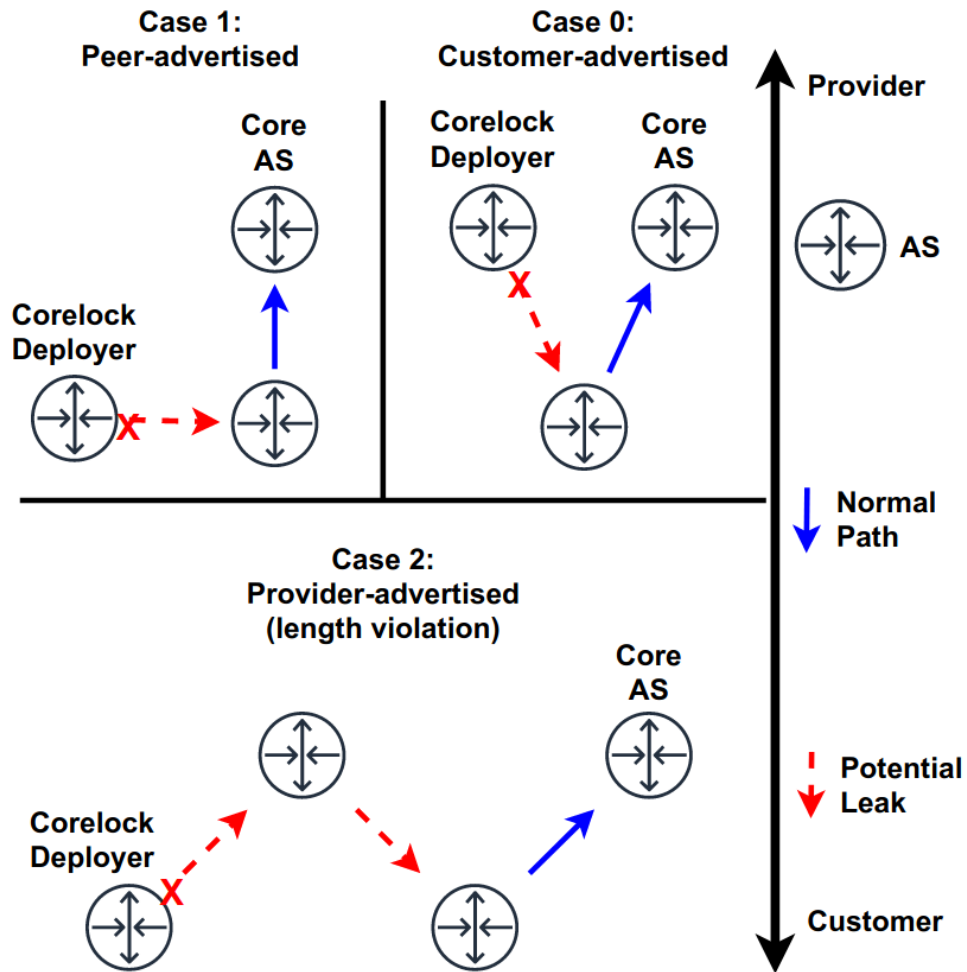


Figure 23: Corelock: RIB-analysis based filtering.

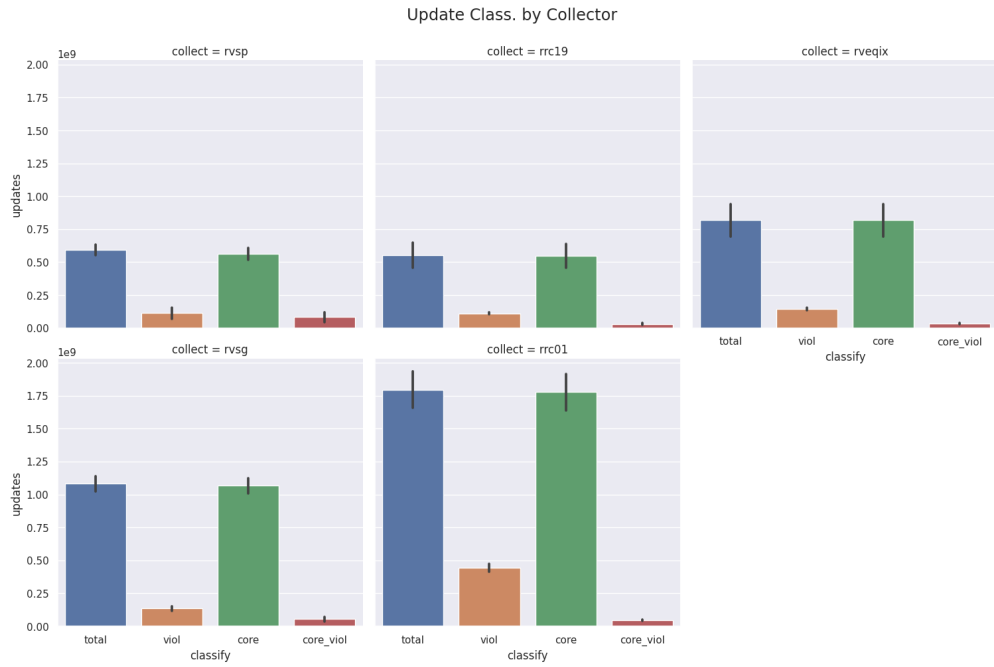


Figure 24: Corelock update classification by regional collectors, 11/20-01/21. Error bars show std. dev.

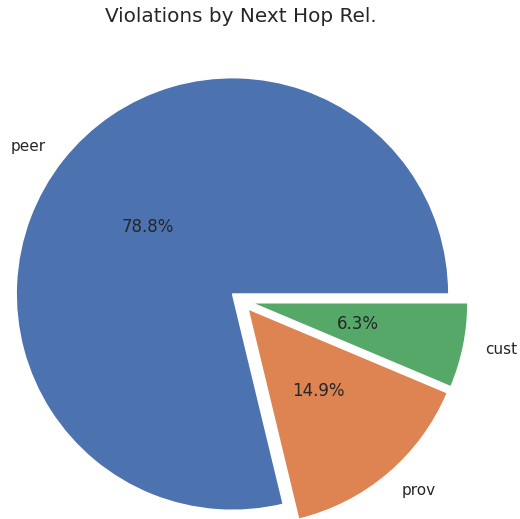


Figure 25: Violations by CAIDA inference of next-hop relationship, 11/20-01/21, RouteViews Equinix.

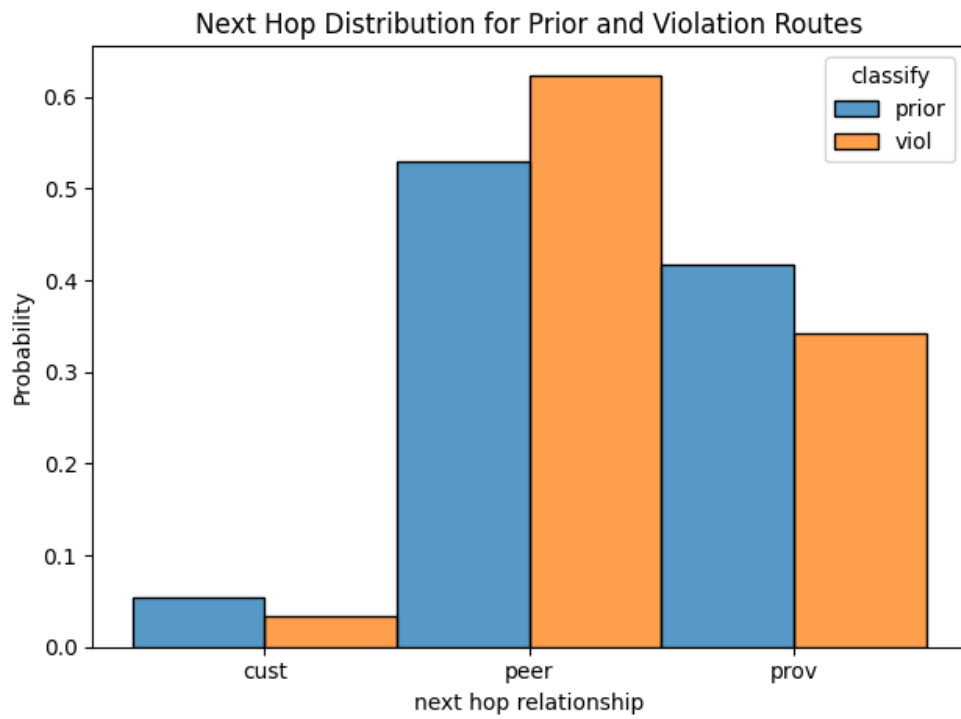


Figure 26: Protector next hop relationship, violation vs. prior routes (subset), 11/20-01/21, RouteViews Equinix.

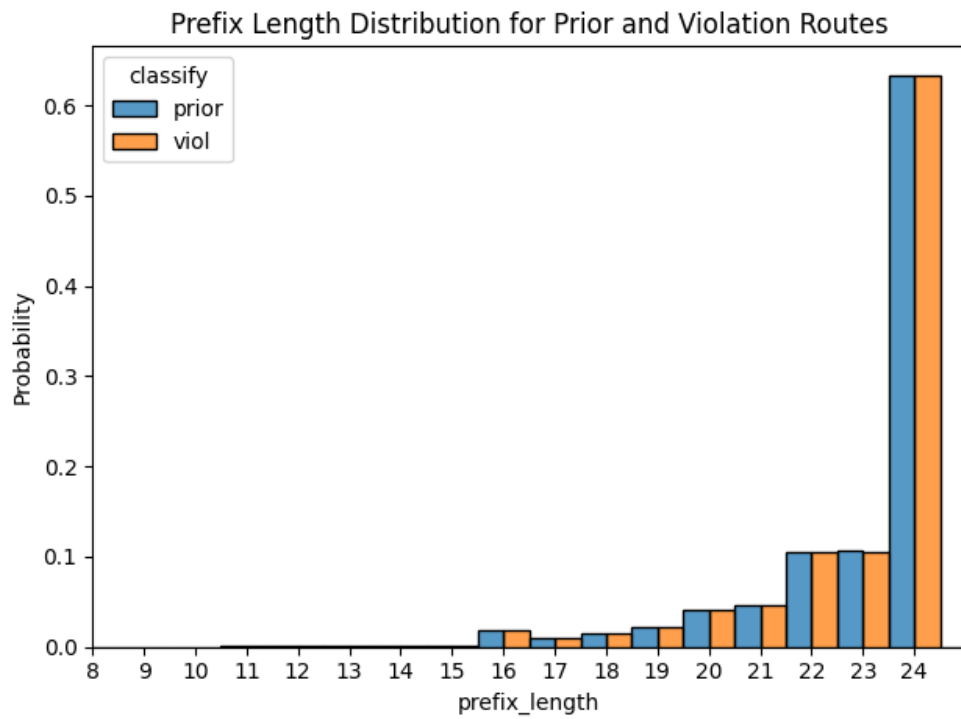


Figure 27: Prefix length, IPv4 violation vs. prior routes (subset), 11/20-01/21, RouteViews Equinix.

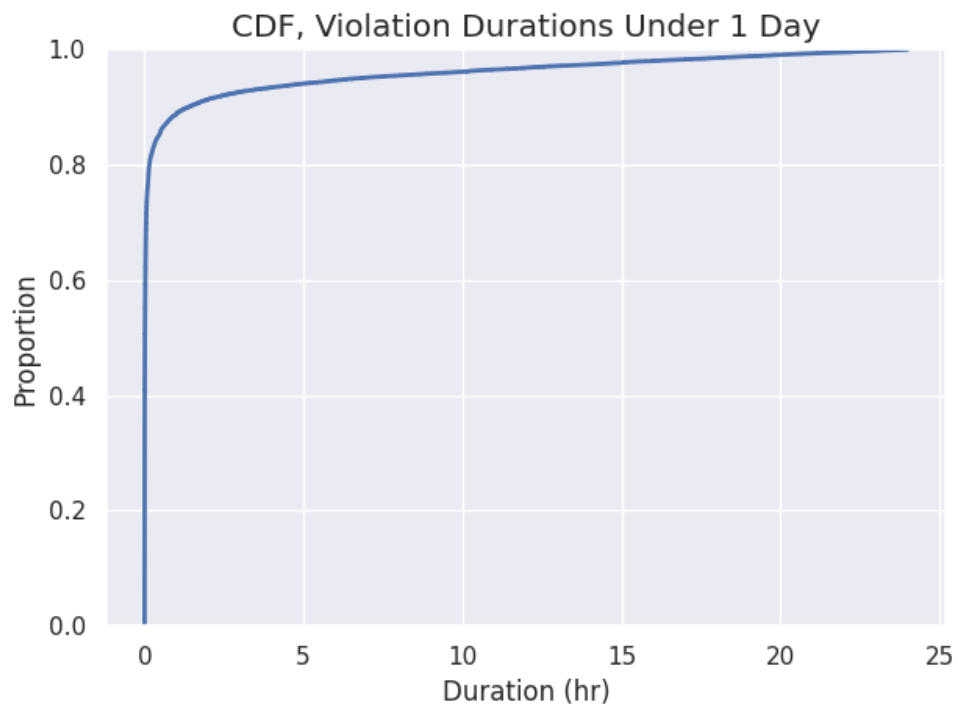


Figure 28: Duration of violations resolved under 24 hours (when observable), 11/20, RouteViews Equinix.

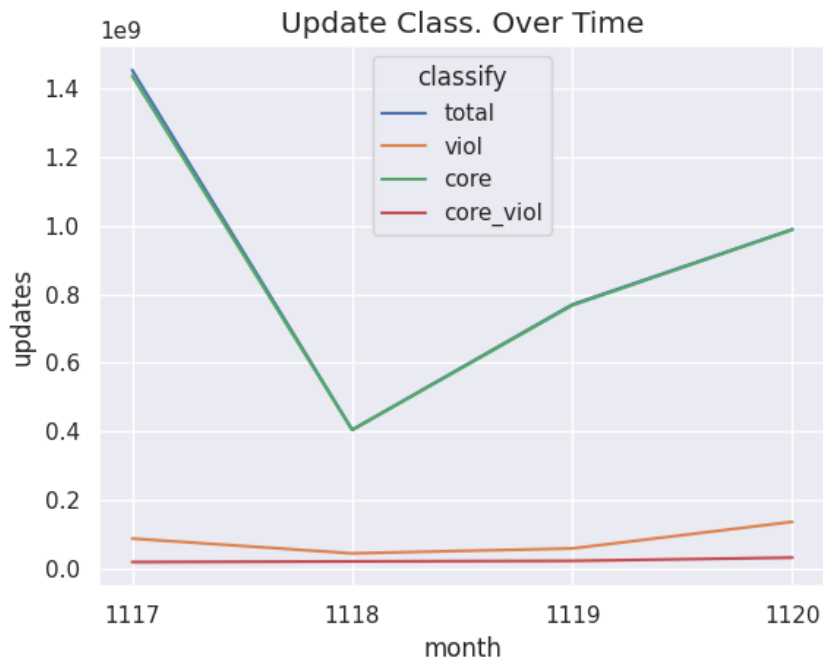


Figure 29: Corelock update classification, November, 2017-2020, RouteViews Equinix.

Vita

Benjamin "Tyler" McDaniel is a computer scientist and software engineer with a broad technical skillset. His eclectic professional background revolves around the design, implementation, and analysis of complex distributed systems.

Tyler's PhD research focuses on the "glue" that binds thousands of disparate networks into a single global Internet - interdomain routing. These networks communicate using BGP, a dated protocol that lacks security controls to protect the Internet's infrastructure from misconfiguration or attack. In fact, single remote networks can (and sometimes do) mistakenly publish incorrect routing information, called route leaks, that spread regionally or even globally. Solutions have been challenged historically by the decentralized nature of the Internet and the capability/incentive disparities between networks. Tyler's work, published in top cybersecurity venues, advocates for simple and deployable security systems that empower the largest "core" networks, acting in their own interest, to protect the Internet at large from route leaks.

Concurrent with his PhD research, Tyler worked as a staff high performance computing engineer at Oak Ridge National Laboratory on national security and leadership-scale physics simulation projects, including QMCPACK. Most of his effort was directed toward ASGarD, a new software package that targets high-dimensional PDEs (like those arising in nuclear fusion) with an adaptive sparse-grid discretization atop a discontinuous Galerkin framework of arbitrary high-order. ASGarD seeks to overcome the historical infeasibility of fully-kinetic solutions to these problems with a novel mathematical approach and scalability-centric design. While software products are the primary artifacts of these efforts, Tyler's work at Oak Ridge has appeared in computational physics, linear algebra, and high-performance computing journals and conferences.

Tyler has accepted an engineering position with Amazon Web Services External Security Services following his PhD.