Kirin: Hitting the Internet with Distributed BGP Announcements

Lars Prehn
MPI-INF
lprehn@mpi-inf.mpg.de

Pawel Foremski
IITiS PAN / DomainTools
pjf@iitis.pl

Oliver Gasser
IPinfo / MPI-INF
oliver@ipinfo.io

ABSTRACT

The Internet is a critical resource in the daily life of billions of users. To support the growing number of users and their increasing demands, operators continuously scale their network footprint—e.g., by joining Internet Exchange Points (IXPs)—and adopt relevant technologies—such as IPv6—which provides a vastly larger address space than its predecessor.

In this paper, we revisit prefix de-aggregation attacks in the light of these two changes and introduce Kirin—an advanced BGP prefix de-aggregation attack that announces millions of IPv6 routes via thousands of IXP connections to overflow the memory of routers within remote ASes. Kirin's highly distributed nature allows it to bypass traditional route-flooding defense mechanisms, such as per-session prefix limits or route flap damping.

We analyze Kirin's theoretical feasibility by formulating it as a mathematical optimization problem, test for practical hurdles by deploying enough infrastructure to perform a micro-scale Kirin attack using 4 IXPs, and validate our assumptions via BGP data analysis, real-world measurements, and router testbed experiments. Despite its low deployment cost, we find that Kirin may inject lethal amounts of routes into the routers of thousands of ASes.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; • **Networks → Denial-of-service attacks**; **Control path algorithms**.

KEYWORDS

BGP, IPv6, DDoS

ACM Reference Format:
Lars Prehn, Pawel Foremski, and Oliver Gasser. 2024. Kirin: Hitting the Internet with Distributed BGP Announcements. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '24), July 1–5, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3634737.3657000

1 INTRODUCTION

The Internet is an indispensable resource for communication, trade, commerce, education, and entertainment in today's world. Over the past years, the Internet has only become more important in people's everyday life, as the reliance of many societies on the Internet has increased with the COVID-19 pandemic [12–14, 42, 76].

ASIA CCS '24, July 1–5, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0482-6/24/07
https://doi.org/10.1145/3634737.3657000

To counter IP address exhaustion, the IPv6 protocol was designed over 20 years ago [32]. In December 2023, 20% of websites are IPv6-ready [122], a third of Autonomous Systems (ASes) announce IPv6 routes [63], and 40% of users access Google via IPv6 [56]. Yet, the additional capabilities provided by IPv6 come with new threats, e.g.: targeted probes can find home routers in the vast IPv6 address space [50, 108]; privacy mechanisms can be defeated and devices can be tracked over time [107]; even a single device using legacy IPv6 addressing can foil all privacy extension efforts [109].

In addition to these attacks on the data plane, IPv6 introduces new challenges for the control plane. Its vast address space raises questions about the scalability of the Internet's standard interdomain routing protocol, the Border Gateway Protocol (BGP). Some large networks own /19 IPv6 prefixes, each of which contains *half a billion* possible /48 subprefixes that can reliably propagate over BGP. As routers have limited amounts of memory, such a large number of subprefixes, if announced, would exhaust the memory of many routers deployed on the Internet. BGP incidents, even as simple as fat-finger mistakes, have a long history of causing major, far-ranging problems to the global Internet [35].

In this paper, we revisit BGP flooding attacks in the light of recent technology developments, and describe a distributed attack named Kirin—short for **K**illing **I**nternet **R**outers in **I**Pv6 **N**etworks. We argue that: *with today's connectivity opportunities, attackers may circumvent per-session prefix limits by distributing unique prefix announcements across enough sessions.* In particular, our contributions can be summarized as follows:

- **Feasibility:** Our main contribution is that we show that today's interconnection platforms (i.e., IXP peering LANs) may provide enough sessions to effectively overwhelm routers within various target ASes while adhering to the prefix limit of each individual session. Our feasibility analysis in Section 4 combines mathematical models, standard route-propagation assumptions, and real-world connectivity data to provide a deep dive into the conditions and cost at which a large-scale, distributed prefix de-aggregation attack becomes feasible: we analyze the required number of (1) targeted ASes, (2) joined peering LANs, (3) contracted transit providers, and (4) bi- and multi-lateral peers.
- **Practicability:** To validate Kirin's practicability, we (1) tested router responses in a lab environment, (2) obtained the resources and deployed the infrastructure needed to perform a micro-scale Kirin attack, (3) performed micro-scale route propagation experiments using the before-mentioned infrastructure and the PEERING testbed, and (4) analyzed the routing ecosystem's redistribution characteristics using data from public route collectors. Our tests showed that Kirin's required resources can be obtained at a low cost (less than 500 EUR) and in a short time (few weeks), that there are no technical hurdles in setting up the attack, that different router types can (partially) crash after exceeding their

(RIB or FIB) memory, and that the routing ecosystem is unlikely to substantially hinder the attack once started.
- **Defense & Notification:** We extensively discuss possible defense mechanisms (see Section 7), provide an open-source implementation for one of them [45], and describe the two-stage vulnerability notification campaign we carried out (see Section 8).

2 BACKGROUND AND RELATED WORK

BGP is the routing protocol that makes the Internet work, where ASes announce and redistribute reachability information between each other according to routing policies [37]. When an AS receives an announcement, it consists of an IP prefix and a path of ASes to traverse, thus the term *route* refers to a prefix-AS path pair.

Routers. Routers establish BGP sessions over TCP between each other. Each router holds a routing information base (RIB) that contains all currently reachable routes—for IPv4 and IPv6 separately. For each prefix, a router determines its best path from all alternatives, and installs it in forwarding information base (FIB). The FIB is then used to quickly retrieve the next hop to which the router forwards a packet. To maximize performance, the FIB is often stored in specialized memory types, such as TCAM or DRAM, which is a scarce resource due to its comparatively high cost. This fact has previously been exploited for theoretical stress attacks on BGP [33].

Route Propagation in Theory. Once a router determines the best-path for a given prefix, it may redistribute it to BGP neighbors. Whether a route is redistributed to a certain neighbor is determined by applying egress filter rules, which express abstract policies that represent a network's business incentives. In 2001, Gao and Rexford [49] first categorized the business relationships between ASes and identified three redistribution patterns: (1) transit relationships, where customers pay transit providers to forward traffic, (2) peering relationships, where two ASes achieve mutual benefits by exchanging traffic at no cost, and (3) sibling relationships, where two ASes appear logically separate, but are operated by the same organization, hence resulting in "arbitrary" redistribution patterns. Based on these categories, ASes only redistribute routes that result in monetary benefit. While ASes would redistribute routes from their customers to all neighbors (as the customer ultimately pays for the delivered traffic), they would not forward routes from peers to other peers or transit providers (as the peer would not pay for resulting maintenance or transit costs).

Route Propagation in Practice. While these abstract relationship models still hold today [51, 67, 75], they are partially superseded by more nuanced relationships [36, 53], e.g., partial transit, paid peering, or hybrid relationships. Besides business incentives, the propagation behavior of an AS can be influenced by e.g., route reputation (filter routes using block lists [2, 29, 117]), aggregation strategy (limit routing table growth or provide partial default routes [69, 74, 103]), or remote signaling (influence propagation using BGP communities [10, 118]).

Propagation Timing. There are multiple factors that determine *when* a router propagates a route, e.g., Minimal Route Advertisement Interval (MRAI), during which BGP updates are aggregated. After the MRAI timer expires, only the active best-path is propagated, which reduces the effects of route flaps, i.e., routes that generate many updates as they rapidly shift between many configurations [41, 52]. Another widely deployed mechanism is Route Flap Damping (RFD) [57, 92], where a BGP session keeps a penalty counter for each prefix. The counter is incremented for each received update and decremented at fixed time intervals. If the counter exceeds the "suppress" threshold, the router starts to dampen the prefix—i.e., withdraw it from all neighbors and stop redistributing updates—until the counter decreases below the "reuse" threshold.

Path Exploration. A router may enter a "path exploration" period due to BGP withdrawals. When an origin AS completely withdraws a prefix, remote ASes start receiving many withdrawal messages from different paths, which are spread in time due to different propagation timings of routers along a path. If a router knows multiple paths for a prefix and receives a withdrawal for its current best path first, then it installs another path as its new best path, and sends an update to its neighbors. If a router knows N paths for a prefix, it may repeat this cycle $N - 1$ times in the worst case, before it finally redistributes the withdraw message: i.e., it "explores" all alternative paths before it fully withdraws the prefix. Path exploration is considered in route propagation experiments and has been studied extensively [4, 78].

Internet eXchange Points (IXPs). Over the last decade, peering has increasingly gained importance [11]. IXPs allow their members to establish BGP sessions cost-effectively with other members on top of peering LANs, i.e., layer-2 switching infrastructures bound to specific geographic locations [3]. Many IXPs provide route servers to further facilitate peering: using a single BGP session, an IXP member can exchange routes with all other connected ASes [101]. As of April 2023, there are more than 700 active IXPs worldwide [91], some of which provide routes for more than half of the Internet via over 1000 ASes [11, 94]. The reachability benefits provided by IXPs make remote participation attractive. Nowadays, "remote peering," i.e., connecting to a peering LAN through a virtual connectivity provider, has become the norm rather than an exception [18, 80, 88].

Topology Blindness. While IXPs are highly popular and have been shown to enable hundreds of thousands of interconnections, most of these interconnections are invisible to the existing BGP monitoring platforms [3, 94]. These platforms in total operate more than 50 route collectors that receive routing updates from more than 600 feeding ASes, but in general they miss many peering links, as those often do not propagate to any feeding AS [3, 8, 55, 89].

Route Aggregation & Filtering. To reduce routing table size, some ASes perform route aggregation, i.e., they summarize multiple more-specific routes into a single aggregate route, which they propagate further [46, 47, 69, 74]. Furthermore, operators often configure their routers to ignore too-specific routes. Especially those with CIDR sizes more specific than /24 and /48—for IPv4 and IPv6, respectively—are commonly filtered [115, 119].

Comparison to Previous Work. While the option to de-aggregate a prefix has been well-known in the operator community for multiple decades, academic literature on the issue is limited.

Chang et al. experimentally investigated the response of 3 commercial grade routers to large BGP routing tables in 2002 [20]. The authors found substantial differences in how routers respond and highlighted that the BGP graceful restart capability could alleviate the effects of BGP malfunctions on IP routing. A deliberate attack

and its impact on the Internet was outside the scope of that paper. Yet, similar to Ceasar et al. [15], the authors advocate for the use of prefix limits on BGP sessions. The operator community largely shares this sentiment as prefix de-aggregation often exacerbates the impact of route leaks [44, 93].

In 2013, Schuchard et al. first characterized the concept of prefix de-aggregation attacks for IPv4 [112]. While they describe the same underlying idea, in comparison with Kirin, the paper does not consider various practical details: (1) they assume that the attack is executed by major transit networks with rich peering fabrics; (2) they assume that an AS can obtain enough address space via squatting (illegitimately announcing unused address space) and that filters against squatting are negligibly deployed; and (3) they assume that typical maximum prefix limits range between tens of thousands of prefixes and the full routing table size. While our work builds upon the same simple idea, it actively addresses these real-world issues, ultimately rendering the attack practically feasible: (1) based on discussions with network operators, we assert that prefix limits are widely deployed and usually range between hundreds to a few thousand prefixes on peering sessions; (2) we leverage IPv6 as an enabler to source millions of legitimately allocated—and hence unfiltered and even RPKI-valid—prefixes; and (3) we make use of remote peering providers, VPS (virtual private server) providers, and IXPs to assemble thousands of sessions allowing arbitrary actors to execute Kirin at a minimal cost. Thus, besides a theoretical feasibility analysis, we evaluate the interlinking parts of our improved attack model in practice and on real-world data.

3 KIRIN: OVERVIEW

In essence, Kirin is simple and ostensibly obvious: the attacker introduces enough new IP routes to overflow the FIB and/or RIB of the BGP routers within victim ASes. After that, the attacker simultaneously withdraws all previously established routes, which triggers the path-exploration phenomenon that leads to a flood of update messages, impacting the performance of routers.

The idea that routers may crash due to memory constraints is not new: many operators already reported crashed routers when the IPv4 routing table reached 512K and 768K routes [1, 38]. Nowadays, high-end devices from major router vendors support ≈ 2–4M routes in total in their FIB: Cisco's Catalyst 8200 and 8500 platforms can store between 800k and 4M routes (depending on the exact model and its respective DRAM storage [25, 26]), Arista's FlexRoute Engine can store up to 2.5M total routes [7], and Juniper's PTX10001 platform can handle 2M total routes [123].

However, it is the new context and the availability of novel techniques that, we believe, re-enable a well-known attack to be successfully executed today, by anyone, and with a limited budget. Although there are various roadblocks built into the routing ecosystem to prevent the exploitation of the FIB/RIB overflow issue, Kirin uses various observations and tricks to maneuver these roadblocks.

3.1 Threat Model

Our threat model, which was already introduced in a similar form by Schuchard et al. [112], focuses on highly connected ASes with legitimate BGP speakers that act maliciously. The goal of our adversary is to fill the FIB or RIB within a remote router to the point where it fully exhausts the available memory using millions of prefix announcements. Hereby, the adversarial AS is not limited to transit ASes; we demonstrate in §4 that even stub ASes are capable of reaching this goal. In fact, we show in §6.1 that an adversary can start without any resources or infrastructure and yet is able to perform Kirin within less than a month and at a cost bearable for individuals. Notably, an AS may either intentionally decide to become an adversary (and explicitly assemble the required infrastructure) or may be forced in this role by an outside entity that compromised various BGP routers or a global route controller.

While an adversary's router can only send BGP messages to the direct neighbors it established sessions with, it relies on those genuine peers to redistribute these messages according to common BGP policies. Further, our adversary may potentially ignore best common routing practices, yet must assume that all other ASes may implement them.

3.2 Accountability

Following our attack model, Kirin can be detected and the attacker can be identified.[1] Similar to BGP hijackers, it might take repeated incidents to hold attackers legally accountable [21] as blame can be averted for a single incident by attributing it to "fat finger" mistakes. In fact, there are multiple (likely accidental) de-aggregation events each year [115]. Hence, we believe that accountability might not be a strong enough deterrent for resourceful attackers to stage Kirin.

3.3 Attack Incentives

Kirin aims to temporarily disrupt the communication between ASes. Our cast of potential bad actors ranges from individuals to state-level actors. Politically-motivated groups or individuals could use Kirin for "hacktivism" campaigns by disconnecting service-hosting ASes during important events as a sign of protest (effectively benefiting from its accountability). Economically motivated individuals may influence the stock price of public companies by sabotaging live demos of products that depend on Internet connectivity (e.g., micro-services hosted by cloud providers). State-level actors may use Kirin to retaliate against economic sanctions, impair critical infrastructure for military operations, or launch a hybrid attack against certain nations—especially those that rely on a single ISP to connect to the Internet [48, 104].

3.4 Enablers & Prerequisites

To enable a successful Kirin attack, several factors are necessary, which will be elaborated on in the following.

Per-Session Max-Prefix Limits. The most common approach to prevent the announcement of too many routes is to set a maximum number of accepted prefixes for each BGP session. Upon hitting this limit, the session may produce a warning, might be capped (stop accepting updates for new prefixes), or can be dropped entirely [27]. Because this approach requires only per-session state, it is simple to implement and requires no cooperation—two key factors that pushed today's wide deployment. Kirin attempts to respect per-session limits by distributing a dedicated set of prefixes to each of *many* BGP sessions: no single prefix is shared between any two

[1] Even if the attacker announces routes with forged AS paths, its direct neighbors can detect the forgery and trace back the attacker.

sessions. Using this strategy transforms the goal of announcing millions of routes into a session-hunting challenge. We further explore this relation in Sections 4 and 6. During our experiments we find IP transit and IXP operators to be permissive about increasing the prefix limits when inquired. One major transit provider stated they do not impose prefix limits on IP transit links; another stated they allow the limit that we set ourselves in the Internet Routing Registry (IRR).

Instant and Cheap BGP Peering. ASes no longer need their own physical connection to establish peering [88]. Remote peering at IXPs is an established reality, and a recent study found that already over 10% of members of major IXPs are remote [80]. Commercial services allow for instantly establishing peering links with dozens of significant IXPs, cloud operators, and data centers [81, 98, 100]. Furthermore, prompt provision of VMs with IXP peering sessions has never been easier: e.g., a VM with NL-IX peering costs under 30 EUR per month [64], and a VM with BGP IP transit costs just *a few* USD per month [121]. Moreover, while carrying out our experiments for this paper, we found it is easy to obtain *free* IPv6 transit—foremost from Hurricane Electric (HE), a major Internet operator, who actively seeks to establish bi-lateral sessions with new IXP members. We also inquired a few major operators and found the cost of a BGP peering port with IP transit would cost around 100–300 USD per month, depending on location and bandwidth.

IPv6. IPv6's address space vastly exceeds that of IPv4. As a consequence, Internet operators also handle much larger IP prefixes, e.g., ARIN's allocation policy states that an ISP should never receive less than a /32 IPv6 prefix allocation [6]. Given that the smallest IPv6 prefix that reliably propagates over BGP is a /48 [95, 115, 119], bad actors could split a typical IPv6 prefix into *many* more subnets than a typical IPv4 prefix. A /29 IPv6 prefix, for example, can source more than 1M unique more-specific[2] prefixes. In general, if C is the difference between the smallest propagating CIDR size (typically a /48) and the parent prefix length, an attacker can source up to $2^{C+1} - 1$ unique routes.

Accessible Internet Resources. It is relatively easy to obtain an AS number and a large IPv6 prefix. A relatively cheap way is to use services of a *sponsoring* LIR, who proxies a request for resources to a RIR (e.g., Securebit [113]). LIR operators can *lease* their allocated IP space, e.g., some offer /29 prefixes with a free trial, which is enough to launch Kirin [97]. Another essentially free (yet illegal) method for malicious attackers is *squatting*, a method in which non-announced Internet resources allocated to an unrelated organization are used [87]. Finally, it is also possible to become a regular LIR and gain direct access to legit and large IPv6 allocations. For example, as of 2023, becoming a RIPE member costs around 2600 EUR and allows for /29 IPv6 allocations without providing justification [85, 86].

Ineffective Route Aggregation. Given that we source all prefixes from the same continuous address space, a wide deployment of route aggregation would limit our attack potential. To overcome this challenge, Kirin only announces non-aggregatable prefix combinations to each neighbor and may also alternate its origin AS.

Circumventable Filtering. While it is hard to enter millions of route-objects into IRR databases, many providers nowadays also accept routes with valid ROAs. As ROA entries allow for CIDR ranges, an adversary may enter a single ROA with CIDR sizes /29–/48, wait for it to propagate, and then would pass, e.g., the route filtering checks of HE [61].

[2]E.g., a /46 prefix can source 7 routes in total: 1x /46, 2x /47, and 4x /48.

3.5 Collateral Damage via Path Exploration

While Kirin fills the FIB/RIB of victim ASes, it does so by announcing millions of routes globally that eventually need to be withdrawn from the Internet again. If a global route gets fully withdrawn, the path-exploration phenomenon may produce a burst of updates (see §2 for details).

Given that Kirin triggers this phenomenon simultaneously for millions of prefixes, it "accidentally" generates a distributed update flooding attack. Given that some ASes use route flap damping to ignore these announcements and stop the redistribution, it is hard to provide realistic estimates on the number of produced updates at each AS. In the worst case, an AS that knows N paths for a prefix may produce $N - 1$ updates during path-hunting, if the best-path-choice and withdraw order are aligned. The minimum and maximum number of generated updates are multiple orders of magnitude apart, not allowing for any insights without prior knowledge, hence we leave a more detailed analysis as future work.

4 THEORETICAL FEASIBILITY ANALYSIS

In this section, we theoretically analyze Kirin's feasibility in two scenarios: (1) the adversary obtains (potentially costly) transit from a few providers and (2) the adversary obtains as many (virtually cost-free) bi-lateral and multi-lateral peerings as possible. While, in reality, an adversary may use both of these scenarios simultaneously, examining them independently allows us to keep our analysis reasonably simple while still obtaining deep insights into Kirin's cost-benefit trade-off. Further, we assume that an adversary only establishes a single (virtual) port via a single method and service provider at each peering LAN.

We start this section by clearly stating the assumptions we make about route redistribution (§ 4.1) and the data sources that we build our analysis upon (§ 4.2). We then define the cost-benefit trade-offs for the first and second scenario as ILP problems (§ 4.3 and § 4.4) and finally discuss our analysis results (§ 4.5).

4.1 Assumptions & Definitions

Routing Policies and Assumptions. The policies that underpin today's inter-domain routing mostly follow economical incentives [5]. In particular, we assume that: (1) if an AS receives a route from a customer, it forwards the route to all neighbors; (2) if an AS receives a route from a settlement-free peer or a provider, it forwards the route to customers only; (3) an AS will always forward a route by the above rules to maximize its economical gain.

Assumptions 1 and 2 are known as the Gao-Rexford redistribution model [49], and are the standard assumptions in the field of AS relationship inference [43, 53, 66, 67, 75]. Assumption 3 has frequently yet implicitly been used for simulating route propagation [70, 83, 125]. These assumptions do not always capture the real-world behavior perfectly—e.g., see complex relationships [53] or non-economic incentives [51])—yet their frequent appearance

renders them as reasonable abstractions. Based on these assumptions, Luckie et al. introduced the notion of the *customer cone*, i.e., the set of all direct and indirect customers of an AS [75]. While they introduced multiple methods to calculate this set, we choose the one that only uses routes which the AS forwarded to its peers and providers, as this yields more stable and realistic results. By recursively applying our three assumptions, one arrives at these high-level statements: (1) routes sent to a peer will eventually reach all ASes in the peer's customer cone; (2) routes sent to a transit provider will eventually reach all ASes globally (even if sometimes a route will not propagate because it is filtered or a certain AS only wants the default route from their provider).

Provider Funnel & Funneling Degree. In this paper, we introduce the concept of *provider funnel* PF_T as the set of all recursively added providers for a given target AS T. We use the example in Figure 1 to further illustrate this concept. In our example, T is multi-homed to two direct providers—P_1 and P_2. Neither P_1 nor P_2 are Tier1 ASes, so they also rely on different transit providers P^* and I to reach certain parts of the Internet. When P^* announces a route to P_1, P_1 likely forwards this route to T. Even though P^* and T share no direct connection, P^* is an indirect provider of T.

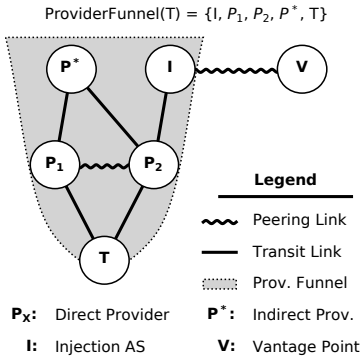

Figure 1: Provider funnel example.

When executing Kirin, our vantage point V has connections to ASes within T's provider funnel. As these ASes redistribute our routes so they ultimately reach T, we call them *injection ASes*. Moreover, as V might maintain multiple BGP sessions to I (e.g., at different IXPs), we further define an *injection session* as a unique BGP session to an injection AS.

Finally, we call the number of ASes in PF_T the *funneling degree* of T and denote it as FD_T. Note that we include T in its provider funnel, i.e., $PF_T = \{P_1, P_2, P^*, I, T\}$. We use the term *restricted funneling degree* FD_T^S to refer to the size of the provider funnel when only considering ASes in the set S, i.e., $FD_T^S = |PF_T \cap S|$.

4.2 Data Sources & Processing

We estimate funneling degrees using two inputs: (1) the number of sessions that each AS has with each peering LAN and (2) the provider funnel for each AS.

Estimating Peering LAN Sessions. On 2022-09-09, we generated a snapshot of EURO-IX's IXP database [39]. We further obtained a PeeringDB snapshot for that day from CAIDA's daily archive [17]. While the EURO-IX data set does not contain a direct reference to the IXP, it contains the PeeringDB identifier for each co-location facility, which allowed us to merge the `(peering LAN, ASN, IPv6 address)` triplets we extracted from both data sources. The obtained data describes 24k sessions via 725 peering LANs.

Estimating IPv6 Provider Funnels. While CAIDA publishes provider-peer-determined customer cones on a monthly basis [16], this data comes with two problems: (1) it is only available for IPv4 and (2) it only uses data from public route collectors which miss substantial portions of the AS topology. Hence, we generate this data set (and most of the required tooling) from scratch.

We first extract all IPv6 routes from public route collector data via BGPStream on 2022-09-09 (including routes from all RIB snapshots and update messages). Next, we add routes from 130 IPv6 route servers of 11 IXPs—e.g., DE-CIX, LINX, and IX.br—including both primary and (potentially multiple) secondary servers. All of these route servers provide a public Alice-lg looking glass utility [31] that has a back-end API allowing for obtaining all IPv6 routes received from their peers. We automated the querying process and obtained the IPv6 routes of all route servers throughout 2022-09-09.

To estimate AS relationships, we utilize the publicly available ASRank script [16]. We modify the script to tailor it towards the IPv6 ecosystem [54]. We use the previously collected IPv6 routes and a list of route server ASNs—that we obtained by selecting ASNs with the "Route Server" network type within our PeeringDB snapshot—as input to the modified ASRank script, which leads to the inference of 247K peering links and 32K transit links. Finally, we convert the IPv6 paths and business relationships into peer-provider-determined customer cones [84]. To calculate provider funnels, we inverted these customer cones, i.e., we checked for each AS in which other AS' customer cone it appears.

4.3 ILP Formulation: Transit Scenario

Now that we obtained the required data sets, we can formalize Kirin's resource needs and attack potential. In our first scenario, we assume that the adversary chooses multiple transit providers and then joins peering LANs to establish additional sessions with the chosen providers. As discussed in § 4.1, we assume that routes announced to a transit provider propagate globally. As every prefix reaches each AS globally, we can focus on the number of sessions that can be obtained by using P_{max} providers and connecting to L_{max} peering LANs.

Sets. Let A be the set of all IPv6-enabled ASes and L be the set of all peering LANs.

Parameters. Let $\omega_{a,l}$ denote the number of unique sessions that can be established with AS $a \in A$ at peering LAN $l \in L$. We can then build the following session matrix:

$$S = \begin{pmatrix} \omega_{a_1,l_1} & \omega_{a_2,l_1} & \cdots & \omega_{a_{|A|},l_1} \\ \omega_{a_1,l_2} & \omega_{a_2,l_2} & \cdots & \omega_{a_{|A|},l_2} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{a_1,l_{|L|}} & \omega_{a_2,l_{|L|}} & \cdots & \omega_{a_{|A|},l_{|L|}} \end{pmatrix}$$

We further provide the parameters $L_{max} \in \mathbb{N}$ and $P_{max} \in \mathbb{N}$ that reflect the maximum number of peering LANs and providers that can be chosen.

Variables. We first introduce a binary decision matrix D that contains a binary decision variable $d_{a,l}$ for each $\omega_{a,l}$ that denotes whether provider $a \in A$ at peering LAN $l \in L$ is part of the solution. Further we introduce two sets of binary decision variables that help us to realize our constraints: CL contains a variables cl_l for each $l \in L$ that determines whether the adversary has to connect to peering LAN l while CP contains a variable cp_a for each $a \in A$ that determines whether a is chosen as a transit provider

ILP Problem Formulation. Given S, L_{max}, and P_{max}, our goal is to chose a set of providers and a set of LANs such that we can obtain the maximum number of sessions, i.e.,

$$\text{maximize} \quad \sum_{l \in L} \sum_{a \in A} \omega_{a,l} * d_{a,l}$$

To ensure that only L_{max} LANs and P_{max} ASes are chosen, we add the following two constraints:

$$\text{wrt.} \quad \sum_{l \in L} cl_l \leq L_{max}$$
$$\sum_{a \in A} cp_a \leq P_{max}$$

Next, we need to make sure that $d_{a,l}$ is always 0 whenever either cl_l or cp_a are 0—if a LAN/AS is not chosen, its entire line/row should only contain zeros. If both, cl_l and cp_a, are set to 1, we want $\omega_{a,l}$ to be arbitrarily large (the more sessions can be obtained, the better). To represent this circumstance we introduce a "large enough" number, B, and formulate the following constraints:

$$\forall a \in A: \quad \sum_{l \in L} \omega_{a,l} * d_{a,l} \leq cp_a * B$$
$$\forall l \in L: \quad \sum_{a \in A} \omega_{a,l} * d_{a,l} \leq cl_l * B$$

For our calculations, we set $B = 10^{10}$ which is multiple orders of magnitude larger than the sum over all entries in the session matrix S. Using this ILP formulation, we can now calculate the maximum number of sessions that can be obtained for at most P_{max} providers when connecting to at most L_{max} peering LANs.

4.4 ILP Formulation: Peering Scenario

In our second scenario, we assume that the adversary chooses multiple settlement-free peers as injection ASes and then joins peering LANs to establish additional sessions with them. This case differs from the previous one, as routes are no longer propagated globally but rather only into the customer cone of the injection AS. We reuse the notation from § 4.3.

While we already defined the funneling degree, FD_a, of an AS $a \in A$ in § 4.1, we need to extend this concept to incorporate the number of sessions that can be established with the injection ASes. We can calculate the session-multiplied funneling degree (SMFD), $f^P_{a,l}$, for AS a using only injection ASes in $I \subset A$ that are present at peering LAN l:

$$f^I_{a,l} = \sum_{i \in I} \omega_{i,l} \cdot \mathbb{1}_{PF_a}(i)$$

where $\mathbb{1}_Y(x)$ represents the indicator function that returns 1 if $x \in Y$ and otherwise 0.

Parameters. After calculating $f^I_{a,l}$ for each (peering LAN, ASN)-pair, we build the matrix F as our first parameter:

$$F = \begin{pmatrix} f^I_{a_1,l_1} & f^I_{a_2,l_1} & \cdots & f^I_{a_{|A|},l_1} \\ f^I_{a_1,l_2} & f^I_{a_2,l_2} & \cdots & f^I_{a_{|A|},l_2} \\ \vdots & \vdots & \ddots & \vdots \\ f^I_{a_1,l_{|L|}} & f^I_{a_2,l_{|L|}} & \cdots & f^I_{a_{|A|},l_{|L|}} \end{pmatrix}$$

We also provide the parameters $R \in \mathbb{N}$ and $N \in \mathbb{N}$ and a set of potential injection ASes, I. R describes the required SMFD to count an AS as *fully affected*, and N describes the required number of fully affected ASes.

Variables. We add two binary decision variables, $d_l \in \{0,1\}, l \in L$ and $c_a \in \{0,1\}, a \in A$; d_l determines whether the adversary should participate at peering LAN l while c_a tracks whether the current LAN selection introduce a session-multiplied funneling degree of at least R for AS a.

ILP Problem Formulation. Given I, F, N, and R, our goal is to minimize the resources—i.e., the number of peering LANs with which we have to connect—needed to perform the Kirin attack, i.e., our objective function is:

$$\text{minimize} \quad \sum_{l \in L} d_l$$

Every valid solution should have a least N fully affected ASes. Hence, we first add this constraint:

$$\sum_{a \in A} c_a \geq N$$

Next, we want to assure that the combined SMFD (across all chosen LANs) of an AS is larger than R for at least N many ASes. Here, we utilize the fact that at least N many c_a variables are set to 1 (by the previous condition) while all other are set to 0. When we multiply R by c_a we effectively generate a switch that either does nothing or conditions the session-multiplied funneling degree of a to be larger than R. As the described condition works only for a single AS, we have to add it once for each AS:

$$\forall a \in A: \quad \sum_{l \in L} d_l * f^I_{a,l} \geq Rc_a$$

Notably, this formulation does not incentivize the ILP solver to arrive at the solution with the largest number of set c_a variables—each solution that sets at least N of them is seen as equally good.

4.5 Analysis & Results

After we formulated our models, we can now run an ILP solver with varying input parameters to explore Kirin's cost-benefit trade-off.

Implementation and Execution. We implement the ILP program using Python's PuLP library [96] configured to use the CBC C++ solver [28] and time out after three hours (i.e., return the current best, potentially sub-optimal solution). We refine sub-optimal solutions whenever possible, i.e., when an optimal run with stricter requirements produced a better objective value than a sub-optimal run, we reuse the results from the optimal run.[3]

[3] E.g., when you need X peering LANs to affect 1000 ASes, you do not need more than X to affect 900 with otherwise identical configuration.

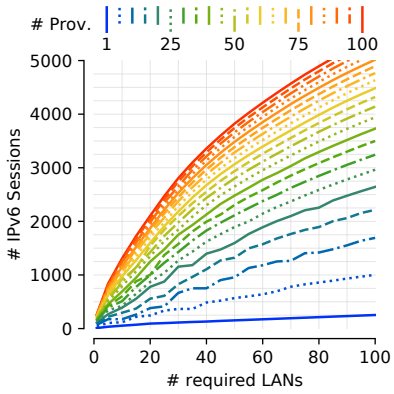

Figure 2: Transit Scenario: trade-off landscape.

We solve the ILP problem defined in § 4.3 for L_{max} and P_{max} values between 1 and 100 and obtain the maximum number of sessions that can be established using each pair. Figure 2 shows different lines for the number of transit providers (P_{max}), the number of peering LANs (L_{max}) on the x-axis, and the resulting number of obtainable sessions on the y-axis.

4.5.1 Transit Scenario. We first observe that we can establish more than a thousand transit sessions by choosing 20 providers and join 25 peering LANs. Given the many possibilities to remotely connect to a peering LAN as well as the cheap (in fact, often free) IPv6 transit options, deploying such an infrastructure is not a major hurdle. If each sessions allows us to send 1000 prefixes (which is not uncommon for transit sessions), this setup would already allow us to inject 1M routes into the global routing table.

We further observe that we need to contract at least 35, 45, and 60 transit providers while joining at least 40, 60, and 80 peering LANs to establish 2000, 3000, and 4000 sessions via just a single port per peering LAN, respectively. While certainly harder to achieve, these scenarios are not out of reach, e.g., for state-backed adversaries.

4.5.2 Peering Scenario. We solve the problem defined in § 4.4 for different SMFDs (R), fully affected ASes (N), and three sets of injection ASes (I). We choose I_{all} to be the set of all IPv6-enabled ASes, which corresponds to setting up a bi-lateral peering link with each AS that participates at a peering LAN. While this connectivity setup is unrealistic for new and small ASes, it provides us with a lower bound for the number of needed peering LANs. Then, we choose restricted sets of injection ASes, i.e., a scenario in which the adversary convinces a limited number of ASes to setup bi-lateral peering. In this scenario, choosing peers with large customer-cones and many sessions is the most ideal; hence, we rank ASes by the product of their customer-cone size and their total session count across all peering LANs and then choose the top 5 and top 20 ASes to represent the injection sets I_5 and I_{20}, respectively.

Figure 3 shows the resulting trade-off landscapes for I_{all} (left), I_{20} (middle), and I_5 (right). Each subplot shows the number of fully affected ASes (N) on the x-axis, different curves for the minimal required session-multiplied funneling degree (R), and the resulting minimal number of required peering LANs on the y-axis. The I_{all} subplot shows that if an adversary could establish bi-lateral peering connections to all ASes at IXP LANs, connecting to a single (or few) peering LAN(s) is sufficient to generate $R = 600$ for 8000 (and probably more) ASes. If the adversary can only establish peering with the injection ASes in I_{20} or I_5, it is realistic to connect to enough peering LANs to introduce $R = 200$ for 5000+ ASes, yet further increasing the required session-multiplied funneling degree might become a substantial obstacle.

While a real adversary would realistically arrive at a setup somewhere between I_{all} and I_{20}, properly representing the full spectrum of possibility, which is probably highly dependent on case-by-case, non-technical aspects (e.g., access to the right contacts, marketing, justification of need, "prestige" in the operator community, etc.), goes beyond the scope of this paper. Yet, our analysis shows that running Kirin *solely* based on peering connections—which often have max-prefix limits of ≈100—seems unrealistic. This insight is further substantiated by our experiments in § 6.2.3 which show that announcements via bi-lateral peering sessions do not necessarily propagate to all ASes within a peer's customer cone, which means that our calculated SMFDs are likely overestimates.

4.5.3 Discussion & Feasibility. While it is unlikely that an adversary acquires enough sessions via bi-lateral peering alone, we demonstrated it is possible to get thousands of sessions from various transit providers[4]. Notably, our analysis took a very conservative approach for estimating the session count. In reality, an adversary could use 5, 10, or even more different VPS and remote peering providers simultaneously to establish multiple ports at each peering LAN, which would provide a linear multiplication factor to the number of sessions that can be established. Hence, a highly motivated adversary could potentially end up with 10k+ sessions, most of which capable to reach a substantial portion of the IPv6 routing ecosystem. Even if each session would be tightly limited to 100 prefixes, such a setup could still produce an increase of 1M prefixes; hence, we conclude that performing Kirin is clearly feasible.

Stability of Results. We repeated the same analyses for January 1, 2021 and April 2, 2022, producing figures with identical structure, yet overall minimally lower values. At all three points in time, our takeaways remained the same.

5 TESTING ROUTER BEHAVIOR

As the "512k day" in August 2014 (as well as its successors) received substantial media coverage [1, 38], router vendors are well aware of the possibility and potential impact of exceeding a router's available RIB or FIB memory. In this section, we examine how routers react to a large number of announced non-aggregatable IPv6 routes.

We perform an evaluation in our testbed with one popular enterprise router—the Juniper MX5 [68]— and one virtual version of a popular core router—the Cisco Virtual Router XRv9k [24]. We use ExaBGP [40], a stateless BGP speaker, to quickly announce a large number of routes from a measurement machine to each of the two routers and assess the impact of those announcements over time. We devise two different scenarios for our experiments: (1) the best-case scenario (from the victim's perspective), where each route contains the shortest possible AS path (i.e., a single AS, resulting in a path length of 1) and no BGP communities attached

[4]While we recommend that IXPs should track session development within the peering LAN over time, this recommendation has economical and operational requirements (e.g., the ability to collect, store, analyze, and correlate vast amounts of traffic data) and, to the best of our knowledge, is currently not widely applied in the wild.

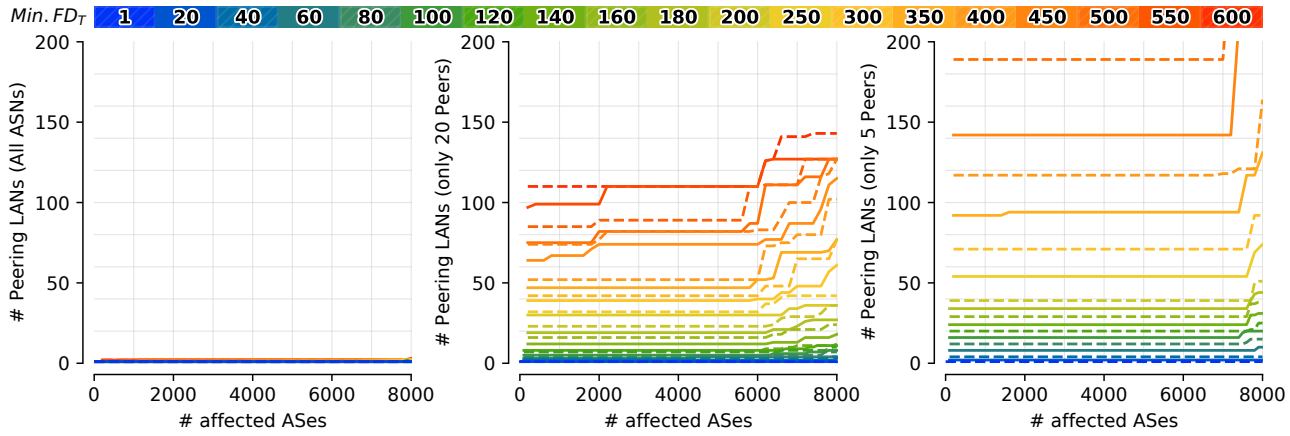

Figure 3: Peering Scenario: trade-off landscape for I_{all} (left), I_{20} (middle), and I_5 (right).

at all; (2) the worst-case scenario, where each route contains the longest possible AS path and the maximum number of large BGP communities[5]. For both AS numbers as well as BGP communities we choose 32 bit values to maximize the impact on router memory. For the hardware and the virtual router we use a minimal default configuration whenever possible. The Juniper MX5 does not have any prefix limit configured by default, while the Cisco Virtual Router XRv9k has a default prefix limit of 524,288 for IPv6 [23]. We increase XRv9k's prefix limits for our experiments. Note that these limits do not make Kirin infeasible: in fact, they can be circumvented by announcing prefixes over multiple sessions. While we continuously announce new routes via ExaBGP, we monitor the resource usage of the system under test.

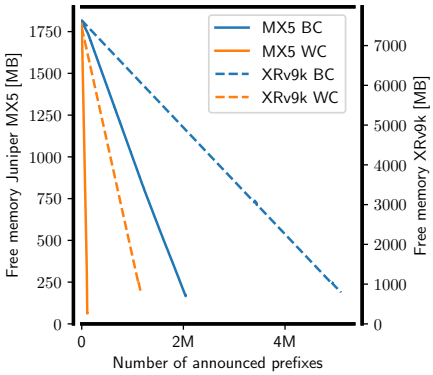

Figure 4: Juniper MX5 and Cisco XRv9k memory exhaustion for best-case (BC) and worst-case (WC) scenarios.

Juniper MX5 In Figure 4 we show the results of our memory exhaustion experiments for the Juniper MX5 router. In the best-case scenario, the router accepts ~2.04 million prefixes, before running out of memory. In the worst-case scenario this number drops to 109k prefixes—which is substantially lower than the current number of all announced IPv6 prefixes (164k) [60]. Once the router's memory is exhausted it will trigger an out-of-memory exception, which results in the BGP routing process being killed. This results in a core dump of the routing process[6], a complete loss of all established BGP sessions, and a purge of all entries in the RIB and FIB.

Cisco Virtual Router XRv9k We show the results for Cisco Virtual Router XRv9k also in Figure 4. In the best-case, the virtual router accepts slightly more than 5 million prefixes before running out of memory. In the worst-case, it only accepts around 1.16 million prefixes. The virtual Cisco router deploys different levels of memory alerts [106]. (1) a *minor alert* is triggered at 85% memory occupancy which leads to rejection when trying to establish new eBGP sessions, whereas already established sessions are not affected. (2) a *severe alert* is raised at 90% memory usage and at that point the BGP daemon shuts down already established eBGP sessions until the memory threshold becomes minor. The daemon shuts down BGP sessions with the lowest percentage of best paths selected (# best paths from peer/# total paths from peer). (3) a *critical alert* will be triggered at 95% memory usage, which leads to a shutdown of all established BGP sessions. In our experiments we trigger all of these alerts sequentially, leading to a complete shutdown of all established BGP sessions.

5.1 Theoretical Lower Bound Memory Usage

We can also calculate the lower bound RIB memory usage of our worst-case announcements as follows:

$$MEM = (PREFIX\_SIZE + (255 \times ASN\_SIZE) + (255 \times COMM\_SIZE)) \times NUM\_PFX$$

Assuming a prefix size of 16 bytes for the IPv6 prefix and 1 byte for the IPv6 prefix length, an ASN size of 4 bytes, and a BGP large community size of 12 bytes, we get a lower bound of $MEM = 4097 \times NUM\_PFX$, i.e. every worst-case prefix needs at least 4kB of RIB memory. Given that today's core routers (e.g., Cisco ASR 9000, Juniper MX960, or Arista 7280CR2K) have RIB memory of 32 or 64 GB, a large number of worst-case prefixes can still bring a router with lots of memory to its knees: 8M prefixes—which can be obtained from de-aggregating a /26—suffice to fill up 32 GB. Finally, as the IPv4 Internet is approaching the 1M route threshold [62] and

[5]The maximum possible AS path length and number of BGP communities that can be sent with ExaBGP is 251, even though the BGP [99] and BGP large communities [58] specifications allow even longer path attributes.

[6]Interestingly, the core file can be so large that it leads to the `/var` directory on the router becoming full, which can not be written to anymore, unless cleaned manually.

with the increasing deployment of technique such as RPKI [116], fewer attacker routes are needed to fill up a router's RIB memory.

Takeaway 1: *Enterprise routers can be overwhelmed with only ≈ 100k announcements, whereas core routers can at least handle around 1M. In the worst case, a route needs ≥4KB router memory to be stored.*

6 REAL-WORLD EXPERIMENTS

Due to ethical concerns as well as economical and social consequences, we can not simply perform a large-scale attack on the Internet. In order to get to a proof-of-concept without inducing any harm, we perform multiple micro-scale experiments that provide interlocking insights into the viability of different attack parts.

6.1 Obtaining Resources and Connectivity

We state in Section 3.4 that it is fairly easy to (1) receive the resources needed to execute the proposed attack, (2) join multiple IXP peering LANs, and (3) establish additional sessions to large transit providers. Below we report on our experience in building and operating a proof-of-concept network capable of performing a micro-scale Kirin attack at negligible cost.

Internet Resources. We obtained an AS number (AS39282) and a few IPv6 address blocks (2a10:cc47:100::/40, 2a0e:b107:e80::/44, and 2a10:2f00:15d::/48) through a sponsoring LIR (Securebit), at a total cost of 270 EUR (valid for 1 year). It took only a few days from requesting these resources until obtaining them for use on the Internet. **Takeaway 2:** *It is possible to obtain ASNs and IP prefixes in a matter of days and at cost bearable for individuals.*

Peering LANs. We built our proof-of-concept network using 2 VMs with IXP access—in Frankfurt (via vServer.site) and Dusseldorf (via Securebit). This allowed us to directly access all route servers and peering LANs of 4 medium-to-large IXPs: DEC-IX, NL-IX, KleyReX, and LocIX. In total, we paid an initial setup fee of 160 EUR and a monthly operating fee of 60 EUR. It took a day until we connected to the first IXP and a few weeks until we connected to the last IXP. **Takeaway 3:** *IXP connectivity providers let new ASes quickly join many peering LANs at a small cost.*

Transit Sessions. We used Hurricane Electric (HE, AS6939) as our main transit provider, as it is one of the most important IPv6 networks [65]. Surprisingly, HE reached out to us about setting up bilateral peering sessions at our IXPs—with a free IPv6 transit option—before we even knew the IXP on-boarding process finished. Additionally, we obtained a VM in Amsterdam from Vultr (AS20473), which provides BGP transit to its customers at no additional cost. We paid no setup fee and a monthly operating fee of 5 USD. The VM was available in a few minutes. **Takeaway 4:** *It is possible to instantly get cheap IP transit.*

Prefix Limits. After finding out our sessions have low prefix limits, we asked if our providers could raise them. As a result, in less than 24h, most operators increased the limits by an order of magnitude without asking for explanation. Other operators stated they could arbitrarily raise the limits given a reasonable justification. **Takeaway 5:** *Increasing prefix limits is often a matter of asking.*

6.2 Propagating Announcements

We now take a closer look at the routing ecosystem. In particular, we analyze the correctness of the claims we made earlier in Sections 3 and 4. We use the infrastructure described in the previous subsection and the PEERING testbed to run real-world experiments for a limited number of ASes and contrast our findings with insights from the information captured by route collector projects.

6.2.1 Setup Specifications. We make use of the proof-of-concept network that we built in the previous subsection to produce IPv6 route announcements. Besides the thousands of (implicitly gained) multilateral peering sessions via route servers, our network only has few direct sessions (most of which connect to HE). To improve our coverage of large IPv6 transit providers and, thereby, improving the generalizability of our results, we also utilize the PEERING testbed [110, 111]. The PEERING testbed is a research network that allocates resources (i.e., ASNs and prefixes) to submitted and accepted project proposals. It has 207 direct IPv6 sessions to 150 different networks distributed across 9 physical locations as well as dedicated IPv6 sessions to 12 route servers at 5 IXPs. All announcements from the PEERING testbed were originated from AS 47065 and sourced from the 2804:269c:10::/44 IPv6 address block. In addition to the standard project capabilities we received the additional capability to announce BGP communities that control the redistribution behavior of the connected route servers.

Announcement Schedules. We announced a dedicated /48 IPv6 prefix via each session. As we control fewer unique /48 prefixes than we have sessions, we first organize the sessions into groups and then reuse the same prefixes across groups (but not within each group). To substantially reduce the likelihood that two successive groups are influenced by each other (e.g., as the first one triggers Route Flat Damping), we adopt a two hour announcement schedule—we announce all prefixes within a group, then wait 30 minutes for route convergence, then withdraw all prefixes, and then wait another 90 minutes before repeating the cycle with the next group. While, e.g., MRAI timers [52] or similar update minimization techniques may introduce few minutes of delay to the propagation of our announcements, we have to wait additional 60 minutes in the last step to ensure that accidentally triggered Route Flap Damping penalties expire [57] and can hence no longer influence the next group of announcements.

Routing Information. We utilize the route collector projects RIPE RIS and Routeviews as our vantage points. In total, they operate 47 IPv6-enabled route collectors that connect to 305 full-feed ASes via 555 IPv6 sessions. For our analysis, we utilize all available RIB snapshots at 2022-09-26, 00:00 UTC+0 using the BGPStream tool.

6.2.2 Route Aggregation. In this first experiment, we announce pairs of aggregatable routes via all our transit providers, i.e., HE at our infrastructure and 7 different transit providers at the PEERING testbed. We repeat this experiment twice. The first time, we announce two consecutive prefixes (i.e., A:B:C::/48 and A:B:C+1::/48) via each session. As both routes are entirely identical, a transit network may decide to aggregate these two routes and only redistribute the resulting /47 route that covers both announcements. The second time, we announce a /47 covering prefix and the /48 subprefix with the same network address (i.e., we announce A:B:C::/47 and A:B:C::/48, but not A:B:C+1::/48). In this scenario, a transit AS may decide to not redistribute the more-specific /48 route given that the AS path is identical. While we see all announcements propagate

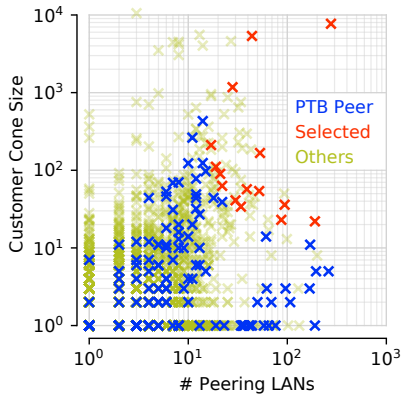

Figure 5: PEERING testbed peers: customer cone vs. peering LANs.

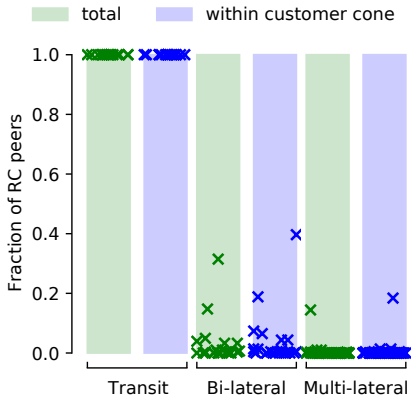

Figure 6: Redistribution behavior of different session types.

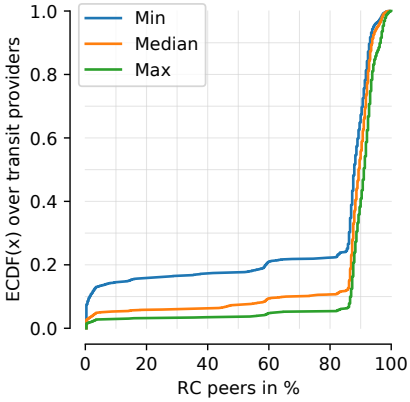

Figure 7: Redistribution for transit providers of single-homed ASes.

globally (i.e., each prefix is seen by at least 95% of all route collector peers), we see no signs of aggregation.

Analysis. When an AS aggregates a route, it may leave up to three clues in BGP messages. First, AS paths may consist of AS sequences and AS sets [99]. A set is generated whenever two routes with different AS paths are aggregated; they represent a summary of the non-matching parts of the two initial AS paths. If an AS aggregates a route and generates no AS set during this process, it should add the `ATOMIC_AGGREGATE` attribute to the message. Finally, an AS may set the `AGGREGATOR` field to indicate that it produced this route aggregate. We searched all IPv6 routes seen by the route collectors for these three hints and display our findings in Table 1. While we observe that 72 % of prefixes have at least one path with an aggregation hint, we only observe 11 % of paths and 10 % of routes with aggregation hints; hence, we believe that only few ASes actively perform route aggregation. While we did not find any signs of route aggregation during our own experiments, an adversary could also make routes less aggregatable by announcing neither neighboring nor covering prefixes to the same neighbor[7], and alternating the origin AS[8]. **Takeaway 6:** *While aggregation is a theoretical challenge, it is rare in practice and can be circumvented.*

	Routes	Paths	Prefixes
Total	58.2M	13.9M	223K
AS set	12K (0%)	10K (0%)	57 (0%)
ATOM.	4.2M (7%)	1.0M (7%)	161K (72%)
AGGR.	5.1M (8%)	1.3M (9%)	16K (6%)
Any Hint	6.4M (10%)	1.6K (11%)	162K (72%)

Table 1: Results of aggregation analysis.

6.2.3 Route Redistribution. Next, we want to analyze whether our assumptions for the route propagation behavior are accurate. While the number of transit providers for both testbeds is limited, applying our schedule to all bi-lateral and multi-lateral peers connected to the PEERING testbed would require extensive amounts of time; hence, we select a smaller set of important ASes.

Tested Networks. The importance of a network for our attack depends on two metrics: the number of sessions we can establish with it and the number of networks it redistributes our announcements to. Figure 5 shows the customer cone size (y-axis) against the number of peering LANs to which a network is connected (x-axis) as a scatter plot for all networks with PeeringDB entries. We mark networks that connect to the PEERING testbed in blue ("PTB Peer") or red ("Selected") and all other networks in green ("Others"). As both dimensions are equally important to Kirin, we select the 15 PEERING peers with the highest harmonic mean of customer cone size and number of potential sessions.

Experiment. Figure 6 shows the fraction of route collector peers (y-axis) reached by /48 announcements via each of the three different session types (on the x-axis). We calculate this fraction twice: once relative to all IPv6 route collector peers (green, "total") and once relative to the peers within the customer cone of the neighbor to which we announced the prefix (blue, "within customer cone"). We can first verify that announcements towards transit providers always propagated globally and that announcements via multi-lateral peers barely propagates at all. Yet, contrary to our assumption, not a single bi-lateral peering sessions redistributed our prefixes into even half of its customer cone. Hence, as noted in the section, we likely over-estimated the achievable funneling degrees in § 4.4.

Analysis. To further test the validity of our transit propagation assumption, we analyze the public BGP data. After removing path-prepending [79], we select all prefixes for which all paths have the same first-hop AS, i.e., that were announced via a single transit provider. Figure 7 shows the minimal, median, and maximal propagating route for each of these transit providers as an ECDF. We observe that for 80 % of transit providers every route propagates globally (i.e., to more than 80 % of route collector peers), while for 89 % and 94 % at least the median and best route propagated globally, respectively. **Takeaway 7:** *While bi- and multi-lateral peers do not necessary redistribute into their entire customer cones, announcing to a transit provider leads to global redistribution.*

[7] While not necessarily generalizable, such a mapping can be generated for our (and more relaxed) scenarios—i.e., distributing around 1M prefixes onto 10k sessions with 100 prefixes each—by skipping the largest CIDR size and then greedily picking the prefix with the largest CIDR size that fulfills both of the outlined conditions.

[8] This would require the aggregator to introduce an AS set into the path, which is rare in practice (see Table 1) and actively discouraged by the operators [73].

7 DISCUSSION

Targetability & Collateral damage. While we introduced Kirin as a global attack, BGP has many mechanisms that allow an adversary to steer route redistribution. Many transit providers allow customers to directly decide which neighbors their routes are redistributed to by attaching specific BGP community attributes [10, 19, 118]. In addition, the adversary may also "poison" the AS path to prevent certain ASes accepting the route. The poisoning method uses cyclic route filters implemented by most routers: if adversary A forges a route with path AXA and this route propagates to X, X is likely to drop it [71]. As the Internet's routing hierarchy has flattened drastically over the last decade, it is likely that a combination of these two mechanisms could be sufficient to steer routes towards most regional networks. Yet, even if the adversary succeeds in steering the majority of the attack towards a single AS, the increase in routes at intermediate ASes should still be noticeable, providing an opportunity to detect the attack and limit the redistribution.

Detection & Mitigation. Kirin is easily detectable due to the large number of routes newly introduced into the IPv6 routing table. Even operators who do not monitor actively could detect Kirin by checking Twitter notifications from the IPv6 routing table size bot or the BGPStream bot [22, 30]. Once some operator detected the attack and shared its origin ASes via some high-visibility operator mailing list such as NANOG, Kirin can be mitigated by employing ingress filters for the covering prefix and origin ASes. Once filters are added, routers no longer import any routes related to the attack, which should prevent them from running out of memory and also normalize the CPU load. Kirin's attack duration is effectively limited to how quickly network operators can coordinate the mitigation efforts—a time that we hope to reduce by raising awareness via this paper and our carefully designed disclosure process.

Traceability & Repercussions. Kirin's resources can easily be traced to the RIRs that allocated them and, from there, could be directly accountable to a specific person or organization. While this seems like a large issue, there are no real sanctions or direct repercussions for "routing vandalism." Bitcanal illustrates this issue nicely: besides loosing some "reputation" via call-outs from researchers and operators [114, 120], it continued to hijack the resources of other ASes over multiple years, until Spamhaus added all related ASNs to their "Don't route and peer" list [117].

7.1 Potential Defense Mechanisms

While Kirin can be mitigated quickly, we would like to entirely prevent it from being feasible. Based on its distributed nature, there is no simple solution that fully prevents the attack; however, there are multiple technical and non-technical mechanisms that may increase the attacker's requirements and limit Kirin's impact, e.g., using router control plane firewalls [45, 124].

Dynamic Yet Tight `Max-Prefix` Limits. Transit providers should introduce dynamically growing yet tight per-session limits on their eBGP sessions. We recommend to allow customers and peers to announce at most 1.5x the number of prefixes they announced the previous day. Similarly, the IPv6 routing table currently grows $<50k$ new prefixes per year [60]; hence, we further recommend to allow a maximum daily increase of at most few thousand prefixes on transit sessions. Automatic imports of max-prefix limits from, e.g., PeeringDB should be checked and not be allowed to surpass a certain predefined limit—otherwise adversaries could enter arbitrary high numbers and abuse the prevention automation.

Per-Origin and Per-Block Prefix Limits. We recommend transit providers to stop redistribution once too many routes are announced within the same covering prefix or by the same origin AS. Although covering prefixes would optimally be determined via the daily RIR delegation files, counting on a /29 or /32 basis might be easier to implement. As of April 2024, the AS with most announcements is AS11172 (with $>6k$ IPv6 routes), and the BGP prefix with the most sub-prefixes is 2409:8000::/20 (with $>11k$ more-specifics). As implementing these limits on each router is costly and may still be insufficient if routers receive unique route sets, we recommend introducing these limits on a route reflector.

Open-Source Implementation. In order to bolster adoption of the mechanism proposed above, we provide its implementation under bgpipe, a BGP reverse proxy and firewall [45]. The contributed stage "limit" supports per-session, per-IP block, and per-AS origin IP prefix limits. If a peer exceeds a limit, our implementation can either drop the BGP session entirely (hard limit), or prevent new prefixes from being announced over that session unless the already announced prefixes that contributed towards that limit are withdrawn first, thus making space for new prefixes (soft limit).

Tight Resource Monitoring & Filtering. We recommend transit providers to monitor the number of established sessions—especially if their peering policy is fully open or they operate an automated session establishment service. If they automatically generate filter lists from third-party data sources (e.g. RPKI [77], IRR [82], or Team CYMRU [29]), we recommend them to carefully monitor the resulting filter size: rapid increase in the number of acceptable prefixes may reveal preparation for a Kirin attack. Further, we recommend transit providers to only redistribute what is correctly registered and avoid loose filtering, i.e., do not assume that more-specific versions of route objects or ROA records are valid by default. While this will not directly prevent the attack, it will increase the effort on the adversary's side to register the resources correctly.

Delayed Propagation of Unfamiliar Routes. The concept behind Pretty Good BGP (PGBGP) [70] is to avoid propagation of anomalous routes, not seen in a window of historical data. Thus, the use of previously unseen routes is delayed, with the hope of identifying and neutralizing any attacks in the meantime. In the context of Kirin, if the attacker used a hijacked prefix, PGBGP could stop the attack from propagating (but note that tracking prefix history needs memory). On the other hand, as Kirin does not need IP prefix hijacking, the attacker can use a large pool of previously unannounced addresses. Thus, we suggest modifying the PGBGP concept to also delay accepting *new* prefixes (i.e., not contained in already propagated, larger address blocks).

8 ETHICAL CONSIDERATIONS

The Kirin attack has the potential to cause serious harm. Hence, we discuss ethical concerns and how we dealt with them below.

Real-world Experiments. We performed a theoretical evaluation of Kirin's potential impact, and assessed the behavior of various BGP implementations in a non-Internet lab environment. Since the Internet is a dynamic system–and the issue of deaggregation is

well-known—it might not be susceptible to the attack in practice. Hence, we also conducted real-world micro-scale experiments.

Harm-benefit analysis. When designing our experiments, we followed [72, 90] to mitigate potential harm to the Internet. This includes a harm-benefit analysis after assessing the theoretically possible impact. To understand potential harm we define scenarios, their probabilities, and consequences. The following scenarios can occur: (1) network operators are made aware of attack potential, (2) malicious actors are made aware of attack potential, (3) our attack causes issues for the network or network devices. Scenario (1) is likely to occur, since we actively reached out to network operators, notified subscribers of operator mailing lists, and the wider public. Scenario (2) is somewhat likely to occur, as malicious actors might be seeing our outreach efforts or observe our experiments in public route collectors data [102, 105]. Scenario (3) is unlikely, as we keep our experiments at micro-scale. Scenario (1) would be beneficial, as increased awareness in the networking community makes successful, large-scale Kirin attacks less likely. The consequences of Scenarios (2) and (3) can be considered harmful and highly harmful, respectively. Given that beneficial Scenario (1) is more likely than harmful Scenarios (2) and (3), we decide to conduct a micro-scale, real-world Kirin experiment. We weighted that our methods are generally known in the community, and that potentially malicious actors may independently develop our scaling methods for prefix deaggregation. At the same time, the networking community considers existing techniques—like per-session prefix limits—sufficient to mitigate the threat, and is unlikely to consider Kirin unless it is practically demonstrated. To demonstrate the feasibility and thus increase the chances of Scenario (1), we decided to conduct a micro-scale experiment using only 500 and 20 prefixes using our Vultr and PEERING testbeds, respectively. Given the size of the IPv6 routing table (>200k prefixes), we believe 500 prefixes (<0.3% of that) to be well inside the daily IPv6 table size churn. We limited the duration of the announcements, made them unlikely to trigger route flapping, and fully withdrawn them after completing the experiments.

Independent Reproduction by Unknown Third-Party. Six days after we conducted our experiments—which did not cause noticeable load at an independent AS we operate as well—we observed an unknown entity that replicated our experimental setup announcing over 8,000 prefixes from a single /32 prefix via Vultr. This caused noticeable load on the independent AS we operate and was noticed in the operator community. We hence decided to accelerate the initial disclosure process we had planned. Furthermore, it demonstrates that threat actors are actively monitoring the global routing table. Researchers conducting experiments for potential vulnerabilities in the routing ecosystem *must* consider that even micro-scale experiments may reveal attack opportunities to third-parties. This leads to substantial problems when an "attack opportunity" is well-known in the community, yet is currently not considered "exploitable enough" [34].

Disclosure Schedule. After the independent third-party potentially replicated our experiments on a substantially larger scale, we immediately launched a two-stage notification process. While a coordinated vulnerability disclosure process [59] would have been preferred–to have more time to discuss with operators *why* this well-known vector is a higher threat *now*—we opted for this path due to the actions of the unknown third-party around 2022/10/5 [9].

- **Private Disclosure Phase (2022/10/11–19).** We first disclosed the details of our attack via a whisper-network of well-connected Tier-1 network operators and IXPs. In this process, we distributed the document enclosed in Figure 9. This process included 8 major IXPs, 20 Tier-1 ASes, and 7 major content providers. We followed-up the initial notification with a clarifying statement, highlighting that an independent third-party potentially already executed the attack on a larger scale. We received the feedback that this clarification made the severity of the problem apparent.
- **Public Disclosure (2022/10/20 and onward).** After sufficient reaction time and no signals to further delay the disclosure, we publicly disclosed our findings via 13 different operator mailing lists (including NANOG and RIPE Routing WG), as well as blog posts, public talks, and social media platforms.

During our disclosure phases, we continuously discussed our findings with network operators, integrated their experiences, and assisted them in deploying prevention mechanisms whenever possible. From private e-mail exchanges, we know that at least two Tier-1 ASes, three cloud providers, and various smaller networks actively configured prevention mechanisms against Kirin.

9 SUMMARY

In this paper, we presented Kirin, an attack that overwhelms BGP routers by globally distributing millions of IPv6 routes via thousands of distributed sessions. We demonstrated that Kirin can bypass traditional prevention mechanisms via its distributed nature and showed that its required infrastructure and resources can be obtained swiftly and at a cost bearable even for single individuals. We tested our assumptions in lab experiments, real-world measurements, and by analyzing passive routing data. Finally, we launched a disclosure campaign to notify network operators and expedite the deployment of prevention mechanisms.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for providing constructive feedback. We also thank Tobias Fiebig and Sebastian Becker for their invaluable help during the responsible disclosure phase.

REFERENCES

[1] Emile Aben. 2019. 768k Day. Will it Happen? Did it Happen? https://labs.ripe.net/author/emileaben/768k-day-will-it-happen-did-it-happen/.
[2] AbuseIPDB. 2022. making the internet safer, one IP at a time. https://www.abuseipdb.com/.
[3] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. 2012. Anatomy of a large European IXP. In *SIGCOMM*.
[4] Eatedal A Alabdulkreem, Hamed S Al-Raweshidy, and Maysam F Abbod. 2014. Using a fight-or-flight mechanism to reduce BGP convergence time. In *ComNet*.
[5] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. 2015. Investigating interdomain routing policies in the wild. In *IMC*.
[6] ARIN. 2022. Number Resource Policy Manual. https://www.arin.net/participate/policy/nrpm/#6-5-2-1-size.
[7] ARISTA. 2022. FlexRoute Engine IP Forwarding - Network Efficiency. https://www.arista.com/en/solutions/flexroute-engine-ip-forwarding.
[8] Todd Arnold, Jia He, Weifan Jiang, Matt Calder, Italo Cunha, Vasileios Giotsas, and Ethan Katz-Bassett. 2020. Cloud provider connectivity in the flat internet. In *IMC*.
[9] BGP6-Table. [n. d.]. Weekly BGP table movement. https://twitter.com/bgp6_table/status/1579562700392103937.
[10] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. 2019. Sico: Surgical interception attacks by manipulating bgp communities. In *CCS*.

[11] Timm Böttger, Gianni Antichi, Eder L Fernandes, Roberto di Lallo, Marc Bruyere, Steve Uhlig, and Ignacio Castro. 2018. The elusive internet flattening: 10 years of IXP growth. *CoRR* (2018).

[12] Timm Böttger, Ghida Ibrahim, and Ben Vallis. 2020. How the Internet reacted to Covid-19: A perspective from Facebook's Edge Network. In *IMC*.

[13] Xander Bouwman, Victor Le Pochat, Pawel Foremski, Tom Van Goethem, Carlos H Gañán, Giovane CM Moura, Samaneh Tajalizadehkhoob, Wouter Joosen, and Michel Van Eeten. 2022. Helping hands: Measuring the impact of a large threat intelligence sharing community. In *31st USENIX Security Symposium (USENIX Security 22)*. 1149–1165.

[14] Francesco Bronzino, Nick Feamster, Shinan Liu, James Saxon, and Paul Schmitt. 2021. Mapping the digital divide: before, during, and after COVID-19. In *ICCIIP*.

[15] Matthew Caesar and Jennifer Rexford. 2005. BGP routing policies in ISP networks. *IEEE network* 19, 6 (2005), 5–11.

[16] CAIDA. [n. d.]. AS Relationships, serial-1. https://publicdata.caida.org/datasets/as-relationships/.

[17] CAIDA. 2022. PeeringDB. https://catalog.caida.org/dataset/peeringdb.

[18] Ignacio Castro, Juan Camilo Cardona, Sergey Gorinsky, and Pierre Francois. 2014. Remote peering: More peering without internet flattening. In *CoNEXT*.

[19] R. Chandra, P. Traina, and T. Li. 1996. *BGP Communities Attribute*. RFC 1997.

[20] Di-Fa Chang, Ramesh Govindan, and John Heidemann. 2002. An empirical study of router response to large BGP routing table load. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. 203–208.

[21] Richard Chirgwin. 2018. BGP hijacker booted off the Internet's backbone. https://www.theregister.com/2018/07/11/bgp_hijacker_booted_off_the_internets_backbone/.

[22] Cisco. [n. d.]. BGPStream. https://twitter.com/bgpstream.

[23] Cisco. 2015. Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide, Release 5.3.x . https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r5-3/routing/configuration/guide/b_routing_cg53xasr9k/b_routing_cg53xasr9k_chapter_010.html.

[24] Cisco. 2018. Cisco IOS XRv 9000 Router Data Sheet. https://www.cisco.com/c/en/us/products/collateral/routers/ios-xrv-9000-router/datasheet-c78-734034.html.

[25] Cisco. 2022. Cisco Catalyst 8200 Series Edge Platforms Data Sheet. https://www.cisco.com/c/en/us/products/collateral/routers/catalyst-8200-series-edge-platforms/nb-06-cat8200-series-edge-plat-ds-cte-en.html.

[26] Cisco. 2022. Cisco Catalyst 8500 Series Edge Platforms Data Sheet. https://www.cisco.com/c/en/us/products/collateral/routers/catalyst-8500-series-edge-platforms/datasheet-c78-744089.html.

[27] Cisco. 2022. Configuring the BGP Maximum-Prefix Feature. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/25160-bgp-maximum-prefix.html.

[28] COIN-OR Foundation. 2022. Cbc. https://github.com/coin-or/Cbc.

[29] TEAM CYMRU. 2022. What Is a Bogon, and Why Should I Filter It? https://team-cymru.com/community-services/bogon-reference/.

[30] Darren O'Connor. [n. d.]. BGP6-Table. https://twitter.com/bgp6_table.

[31] DE-CIX. [n. d.]. Alice-LG - Your friendly looking glass. https://github.com/alice-lg/alice-lg.

[32] S. Deering and R. Hinden. 1998. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard). https://www.rfc-editor.org/rfc/rfc2460.txt

[33] Wenping Deng, Peidong Zhu, Xicheng Lu, and Bernhard Plattner. 2010. On Evaluating BGP Routing Stress Attack. *J. Commun.* 5, 1 (2010).

[34] Constanze Dietrich, Katharina Krombholz, Kevin Borgolte, and Tobias Fiebig. 2018. Investigating system operators' perspective on security misconfigurations. In *CCS*.

[35] Doug Madory. 2023. A Brief History of the Internet's Biggest BGP Incidents. https://www.kentik.com/blog/a-brief-history-of-the-internets-biggest-bgp-incidents/.

[36] DrPeering. 2022. The Art of Peering: The Peering Playbook. http://drpeering.net/white-papers/Art-Of-Peering-The-Peering-Playbook.html.

[37] J. Durand, I. Pepelnjak, and G. Doering. 2015. BGP Operations and Security. RFC 7454 (Best Current Practice). https://www.rfc-editor.org/rfc/rfc7454.txt

[38] Marco Hogewoning Emile Aben. 2014. 512K-mageddon? https://labs.ripe.net/author/emileaben/512k-mageddon/.

[39] Euro-IX. 2022. IXPDB. https://www.euro-ix.net/.

[40] Exa Networks. [n. d.]. ExaBGP on GitHub. https://github.com/Exa-Networks/exabgp.

[41] Alex Fabrikant, Umar Syed, and Jennifer Rexford. 2011. There's something about MRAI: Timing diversity can exponentially worsen BGP convergence. In *INFOCOM*.

[42] Anja Feldmann, Oliver Gasser, Franziska Lichtblau, Enric Pujol, Ingmar Poese, Christoph Dietzel, Daniel Wagner, Matthias Wichtlhuber, Juan Tapiador, Narseo Vallina-Rodriguez, Oliver Hohlfeld, and Georgios Smaragdakis. 2021. A Year in Lockdown: How the Waves of COVID-19 Impact Internet Traffic. *CACM* 64 (June 2021). Issue 7.

[43] Guoyao Feng, Srinivasan Seshan, and Peter Steenkiste. 2019. UNARI: an uncertainty-aware approach to AS relationships inference. In *CoNEXT*.

[44] Jérôme Fleury, Tom Strickx, and Martin J. Levy. 2022. Anatomy of a route leak. https://ripe84.ripe.net/presentations/36-Cloudflare-Anatomy-of-a-Route-Leak-RIPE84-1.pdf.

[45] Pawel Foremski. 2024. bgpipe: BGP reverse proxy and firewall. https://github.com/bgpfix/bgpipe/.

[46] Roque Gagliano, Eduardo Grampin, Javier Baliosian, Xavier Masip-Bruin, and Marcelo Yannuzzi. 2009. Understanding IPv4 prefix de-aggregation: Challenges for routing scalability. In *2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops*. IEEE, 107–112.

[47] Julien Gamba, Romain Fontugne, Cristel Pelsser, Randy Bush, and Emile Aben. 2017. BGP Table Fragmentation: what & who?. In *Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*.

[48] Alexander Gamero-Garrido, Esteban Carisimo, Shuai Hao, Bradley Huffaker, Alex C Snoeren, and Alberto Dainotti. 2022. Quantifying Nations' Exposure to Traffic Observation and Selective Tampering. In *Passive and Active Measurement: 23rd International Conference, PAM 2022, Virtual Event, March 28–30, 2022, Proceedings*. Springer, 645–674.

[49] Lixin Gao and Jennifer Rexford. 2001. Stable Internet routing without global coordination. *ToN* 9, 6 (2001).

[50] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the expanse: understanding and unbiasing IPv6 hitlists. In *Proceedings of the Internet Measurement Conference 2018*. 364–378.

[51] Phillipa Gill, Michael Schapira, and Sharon Goldberg. 2013. A survey of interdomain routing policies. *CCR* 44, 1 (2013).

[52] Rajvir Gill, Ravinder Paul, and Ljiljana Trajković. 2012. Effect of MRAI timers and routing policies on BGP convergence times. In *IPCCC*.

[53] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and KC Claffy. 2014. Inferring complex AS relationships. In *IMC*.

[54] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and Kc Claffy. 2015. IPv6 AS relationships, cliques, and congruence. In *PAM*.

[55] Vasileios Giotsas, Shi Zhou, Matthew Luckie, and Kc Claffy. 2013. Inferring multilateral peering. In *CoNEXT*.

[56] Google. [n. d.]. IPv6 Adoption. https://www.google.com/intl/en/ipv6/statistics.html.

[57] Caitlin Gray, Clemens Mosig, Randy Bush, Cristel Pelsser, Matthew Roughan, Thomas C Schmidt, and Matthias Wahlisch. 2020. BGP beacons, network tomography, and Bayesian computation to locate route flap damping. In *IMC*.

[58] J. Heitz (Ed.), J. Snijders (Ed.), K. Patel, I. Bagdonas, and N. Hilliard. 2017. BGP Large Communities Attribute. RFC 8092 (Proposed Standard). https://www.rfc-editor.org/rfc/rfc8092.txt

[59] Allen D Householder, Garret Wassermann, Art Manion, and Chris King. 2017. *The cert guide to coordinated vulnerability disclosure*. Technical Report.

[60] Geoff Houston. [n. d.]. IPv6 CIDR REPORT for 10 Oct 22. https://www.cidr-report.org/v6/as2.0/.

[61] Hurricane Electric. [n. d.]. Hurricane Electric Route Filtering Algorithm. https://routing.he.net/algorithm.html.

[62] Geoff Huston. [n. d.]. AS131072 IPv4 CIDR Report. https://www.cidr-report.org/as2.0/.

[63] Geoff Huston. 2021. AS131072 IPv6 BGP Table Data. https://bgp.potaroo.net/v6/as2.0/index.html.

[64] iFog GmbH. [n. d.]. IXP Access. https://ifog.ch/en/ip/ixp-access-vm-amsterdam.

[65] Siyuan Jia, Matthew Luckie, Bradley Huffaker, Ahmed Elmokashfi, Emile Aben, Kimberly Claffy, and Amogh Dhamdhere. 2019. Tracking the deployment of IPv6: Topology, routing and performance. *Computer Networks* 165 (2019).

[66] Yuchen Jin, Colin Scott, Amogh Dhamdhere, Vasileios Giotsas, Arvind Krishnamurthy, and Scott Shenker. 2019. Stable and Practical {AS} Relationship Inference with ProbLink. In *NSDI*.

[67] Zitong Jin, Xingang Shi, Yan Yang, Xia Yin, Zhiliang Wang, and Jianping Wu. 2020. Toposcope: Recover as relationships from fragmentary observations. In *IMC*.

[68] Juniper. [n. d.]. Features supported on MX5 in Junos OS 21.2R3. https://apps.juniper.net/home/mx5/features.

[69] Costas Kalogiros, Marcelo Bagnulo, and Alexandros Kostopoulos. 2009. Understanding incentives for prefix aggregation in BGP. In *ReArch*.

[70] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. 2006. Pretty good BGP: Improving BGP by cautiously adopting routes. In *ICNP*.

[71] Ethan Katz-Bassett, David R Choffnes, Ítalo Cunha, Colin Scott, Thomas Anderson, and Arvind Krishnamurthy. 2011. Machiavellian routing: improving internet availability with bgp poisoning. In *HotNets*.

[72] Erin Kenneally and David Dittrich. 2012. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *Available at SSRN 2445102* (2012).

[73] W. Kumari and K. Sriram. 2011. *Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP*. BCP 172. RFC Editor.

[74] Franck Le, Geoffrey G Xie, and Hui Zhang. 2011. On route aggregation. In *CoNEXT*.

[75] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. 2013. AS relationships, customer cones, and validation. In *IMC*.

[76] Andra Lutu, Diego Perino, Marcelo Bagnulo, Enrique Frias-Martinez, and Javad Khangosstar. 2020. A characterization of the COVID-19 pandemic impact on a mobile network operator traffic. In *IMC*.

[77] MANRS. [n. d.]. RPKI Week. https://www.manrs.org/resources/events/rpki-week/.

[78] Z Morley Mao, Randy Bush, Timothy G Griffin, and Matthew Roughan. 2003. BGP beacons. In *IMC*.

[79] Pedro Marcos, Lars Prehn, Lucas Leal, Alberto Dainotti, Anja Feldmann, and Marinho Barcellos. 2020. AS-Path Prepending: there is no rose without a thorn. In *IMC*.

[80] Fabricio Mazzola, Pedro Marcos, Ignacio Castro, Matthew Luckie, and Marinho Barcellos. 2022. On the Latency Impact of Remote Peering. In *PAM*.

[81] Megaport. [n. d.]. Data Centre Interconnect. https://www.megaport.com/services/datacentre-interconnect/.

[82] Merit. [n. d.]. Internet Routing Registries. https://www.irr.net/.

[83] Reynaldo Morillo, Justin Furuness, Cameron Morris, James Breslin, Amir Herzberg, and Bing Wang. 2021. ROV++: Improved Deployable Defense against BGP Hijacking.. In *NDSS*.

[84] Lucas Müller, Matthew Luckie, Bradley Huffaker, Kc Claffy, and Marinho Barcellos. 2019. Challenges in inferring spoofed traffic at IXPs. In *CoNEXT*.

[85] RIPE NCC. 2022. Billing, Payment and Fees. https://www.ripe.net/participate/member-support/payment.

[86] RIPE NCC. 2022. How to Request an IPv6 Allocation. https://www.ripe.net/manage-ips-and-asns/ipv6/request-ipv6/how-to-request-an-ipv6-allocation.

[87] Eugenio Nerio Nemmi, Francesco Sassi, Massimo La Morgia, Cecilia Testart, Alessandro Mei, and Alberto Dainotti. 2021. The parallel lives of Autonomous Systems: ASN Allocations vs. BGP. In *IMC*.

[88] George Nomikos, Vasileios Kotronis, Pavlos Sermpezis, Petros Gigis, Lefteris Manassakis, Christoph Dietzel, Stavros Konstantaras, Xenofontas Dimitropoulos, and Vasileios Giotsas. 2018. O peer, where art thou? Uncovering remote peering interconnections at IXPs. In *IMC*.

[89] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. 2009. The (in) completeness of the observed Internet AS-level structure. *ToN* 18, 1 (2009).

[90] Craig Partridge and Mark Allman. 2016. Ethical considerations in network measurement papers. *CACM* 59, 10 (2016).

[91] PCH. 2023. Internet Exchange Directory. https://www.pch.net/ixp/dir.

[92] Cristel Pelsser, Olaf Maennel, Pradosh Mohapatra, Randy Bush, and Keyur Patel. 2011. Route flap damping made usable. In *PAM*.

[93] Alin C. Popescu, Brian J. Premore, and Todd Underwood. 2004. Anatomy of a Leak: AS9121. https://www.youtube.com/watch?v=l_UCneZm6dE.

[94] Lars Prehn, Franziska Lichtblau, Christoph Dietzel, and Anja Feldmann. 2022. Peering Only? Analyzing the Reachability Benefits of Joining Large IXPs Today. In *PAM*.

[95] Lars Prehn, Franziska Lichtblau, and Anja Feldmann. 2020. When wells run dry: the 2020 IPv4 address market. In *CoNEXT*.

[96] pulp documentation team. 2022. Optimization with PuLP. https://coin-or.github.io/pulp/.

[97] RapidSeedbox. 2022. IPv6 Address for Rental. https://www.rapidseedbox.com/ipv4-rental#pricing.

[98] IX Reach. [n. d.]. Remote peering. https://www.ixreach.com/services/remote-peering/.

[99] Y. Rekhter (Ed.), T. Li (Ed.), and S. Hares (Ed.). 2006. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard). https://www.rfc-editor.org/rfc/rfc4271.txt

[100] RETN. [n. d.]. Remote IX. https://retn.net/products/remote-ix.

[101] Philipp Richter, Georgios Smaragdakis, Anja Feldmann, Nikolaos Chatzis, Jan Boettger, and Walter Willinger. 2014. Peering at peerings: On the role of IXP route servers. In *IMC*.

[102] RIPE NCC. [n. d.]. RIPE Routing Information Service (RIS). https://www.ripe.net/ris/.

[103] Nils Rodday, Lukas Kaltenbach, Italo Cunha, Randy Bush, Ethan Katz-Bassett, Gabi Dreo Rodosek, Thomas C Schmidt, and Matthias Wählisch. 2021. On the Deployment of Default Routes in Inter-domain Routing. In *TAURIN*.

[104] Joey Roulette. 2023. SpaceX curbed Ukraine's use of Starlink internet for drones. https://www.reuters.com/business/aerospace-defense/spacex-curbed-ukraines-use-starlink-internet-drones-company-president-2023-02-09/.

[105] RouteViews Project. [n. d.]. University of Oregon RouteViews Project. https://www.routeviews.org/.

[106] Routing-Bits blog. [n. d.]. Low Memory Handling. https://routing-bits.com/2011/08/21/low-memory-handling/.

[107] Erik Rye, Robert Beverly, and Kimberly C Claffy. 2021. Follow the scent: defeating IPv6 prefix rotation privacy. In *IMC*.

[108] Erik C Rye and Robert Beverly. 2020. Discovering the IPv6 Network Periphery. In *PAM*.

[109] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. 2022. One Bad Apple Can Spoil Your IPv6 Privacy. *CCR* 52 (June 2022). Issue 2.

[110] Brandon Schlinker, Todd Arnold, Italo Cunha, and Ethan Katz-Bassett. 2019. PEERING: Virtualizing BGP at the Edge for Research. In *CoNEXT*.

[111] Brandon Schlinker, Kyriakos Zarifis, Italo Cunha, Nick Feamster, and Ethan Katz-Bassett. 2014. Peering: An as for us. In *CoNEXT*.

[112] Max Schuchard, Christopher Thompson, Nicholas Hopper, and Yongdae Kim. 2013. Peer pressure: Exerting malicious influence on routers at a distance. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 571–580.

[113] Securebit AG. 2022. Internet resources. https://www.securebit.ch/internet/resources.

[114] Security Boulevard. [n. d.]. Notorious 'Hijack Factory' Shunned from Web. https://securityboulevard.com/2018/07/notorious-hijack-factory-shunned-from-web/.

[115] Khwaja Zubair Sediqi, Lars Prehn, and Oliver Gasser. 2022. Hyper-specific prefixes: gotta enjoy the little things in interdomain routing. *CCR* 52, 2 (2022).

[116] Job Snijders. [n. d.]. RPKI's 2022 year in review — growth and innovation. https://blog.apnic.net/2023/01/18/rpkis-2022-year-in-review-growth-and-innovation/.

[117] SPAMHAUS. 2022. The Spamhaus Don't Route Or Peer Lists. https://www.spamhaus.org/drop/.

[118] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. Bgp communities: Even more worms in the routing can. In *IMC*.

[119] Stephen Strowes. 2019. Visibility of IPv4 and IPv6 Prefix Lengths in 2019. https://labs.ripe.net/author/stephen_strowes/visibility-of-ipv4-and-ipv6-prefix-lengths-in-2019/.

[120] Cecilia Testart, Philipp Richter, Alistair King, Alberto Dainotti, and David Clark. 2019. Profiling BGP serial hijackers: capturing persistent misbehavior in the global routing table. In *IMC*.

[121] Vultr. [n. d.]. Announce Your IP Space. https://www.vultr.com/features/bgp/.

[122] W3Techs. [n. d.]. Usage statistics of IPv6 for websites. https://w3techs.com/technologies/details/ce-ipv6.

[123] Xantaro. 2022. Juniper Networks PTX10001 tested as a Peering / Edge Router. https://www.xantaro.net/en/tech-blogs/juniper-ptx10001-test-peering-edge-router/.

[124] Ying Zhang, Z Morley Mao, and Jia Wang. 2007. A Firewall for Routers: Protecting against Routing Misbehavior. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*. IEEE, 20–29.

[125] Zheng Zhang, Ying Zhang, Y Charlie Hu, and Z Morley Mao. 2007. Practical defenses against BGP prefix hijacking. In *CoNEXT*.

A PRIVATE DISCLOSURE NOTIFICATION

Figure 9 shows the initial email that we sent out in the private disclosure notification. In Figure 8 we show the follow-up email highlighting why the attack can cause serious harm and has already been run on a larger-scale by an unknown third party.

```
Dear colleagues,

we received some feedback that the message we provided you with is simply
stating the obvious, and noticed an important piece of information missing:

Note, that we conducted experiments with a limited (<=500 prefixes) test-setup
around the 29th of September. On the 5th of October an entity unknown to us
replicated our experiments via AS20473 with around 8k prefixes, already causing
noticeable load but yet staying below the potential of this technique. We hence
assume that our technique is by now known--not only commonly known in the
community but potential attackers being consciously aware--to third parties,
which is why we are sending out these notifications for something technically
well known. We plan to notify the wider networking community in one week from
now.

With best regards,
<blanked>
```

Figure 8: Follow-up email text of private disclosure notification.

Dear <Person>,

I'm a researcher at <blanked> in <blanked> and received your contact from <Person>, who believes that you might be the right contact at <Company> for the following issue:

We started the private disclosure process for an IPv6-based routing attack discovered in a research collaboration between the <blanked> in <blanked> and <blanked>. We'd highly appreciate your valuable insights and hope you join our efforts in globally deploying effective prevention mechanisms. To keep the Internet and its users safe, it is important to keep the attack details confidential until prevention mechanisms are in place; we count on you not to publicly share this information prior to the public disclosure, which we currently plan for Wednesday, 19th October 2022.

What is the problem?
Routers either crash, drop sessions, or behave in other unintended ways when their FIB or RIB runs out of memory. While newer routers can store up to 4M prefixes, many ASes still run (at least some) older hardware that may only be able to store 1M routes or even less. TL;DR: We found an attack that allows an adversary to introduce very quickly more than 1M new and unique IPv6 prefixes into the global routing table and is only preventable with the help of major transit networks and IXPs. If, afterwards, these prefixes also get withdrawn simultaneously, the resulting path-hunting behavior additionally results in a massive flooding attack.

How does the adversary even obtain 1M unique prefixes?
After obtaining a /29 address block from any of the RIRs (e.g., RIPE this does not even require need-based justification) the adversary announces every possible /48, /47,... /29 route leading to the announcement of 1.048.575 unique routes---if C is the difference between the minimal propagating CIDR size, /48, and the CIDR size of the address block from which an attacker sources routes, the adversary can announce up to 2^(C+1) - 1 unique routes, e.g., a /46 block can source seven routes in total: one /46 route, two /47 routes, and four /48 routes.

Don't we have per-session prefix limits that prevent such attacks?
If the average per-session limit is X, an adversary 'simply' has to distribute its routes via 1M/X many sessions, i.e., per-session limits do not eliminate the issue, they only transform it into a session-hunting challenge. During our real-world experiments and discussions, we noticed that while many ASes set tight (often 100-500 prefixes) per-session limits on their peering sessions, it's less common that ASes on either side of a transit session enforce prefix limits.

Why does ROV not protect us from this attack?
It is possible to set a single ROA entry that specifies that the /29 prefix can be announced with CIDR sizes up to /48. If the adversary generates such a ROA and waits some days for it to propagate to all validating ASes, each of the more than 1M prefixes would be a valid announcement.

How can an adversary even get hundreds or thousands of sessions?
The idea is that remote peering providers and VPS providers (e.g., Vultr) enable the adversary to quickly and cheaply 'click together' (virtual) ports at many (think 20+) different peering LANs. The adversary obtains transit by picking providers that also establish transit sessions over peering LANs (Hurricane Electric being the prime example), many bi-lateral peering sessions via openly/aggressively peering networks (that can be identified via, e.g., PeeringDB), and additional (less effective) sessions via multi-lateral peering with Route Servers. Surprisingly, while it would be hard to assemble enough sessions with just one port at each peering LAN (yet eventually doable), this limitation does not exist in reality; while certain providers directly allow clicking multiple ports for a single peering LAN, there are also multiple providers---this allows the adversary to obtain a 5X to 10X factor for its session counts by establishing multiple sessions to each neighbor (in fact each port of each neighbor).

Do these routes even propagate far enough?
TL;DR: yes. As a rule of thumb: The routes announced via transit sessions usually propagate globally, routes announced to bi-lateral peers usually propagate into the peer's customer cone, and routes announced via multi-lateral peering usually propagate only to the peer's regional customers. As part of our research, we analyzed the propagation behavior and found that an adversary that combines announcements via all three peering types can inject lethal amounts of IPv6 routes into routers of 8k+ ASes, i.e., yes, enough of these routes propagate far enough.

Don't ASes along the path aggregate the individual routes?
While some ASes do aggregate routes, it is possible to launch the attack in such a way that routes can not be aggregated: the adversary would have to choose the prefixes in each session in such a way that neither two consecutive prefixes nor a prefix and its covering prefix are announced via the same session and/or neighbor. To be extra safe, the adversary could switch between multiple origin ASNs for the announcements or use path-poisoning to alter a route's AS path.

What can IXPs do to help prevent the attack?
Ensure that your route servers have tight prefix limits and that they only accept a small number of sessions from each participant.

If applicable, monitor your members' session acquisition behavior (e.g., by looking for BGP-session related packets in the peering LAN's traffic data) to identify potential adversaries early.

What can transit providers do to help prevent the attack?
Introduce dynamically growing yet tight per-session limits on all of your sessions. Allow, e.g., customers and peers to announce at most 1.3x the number of prefixes they announced yesterday. Similarly, the IPv6 routing table currently grows at a rate of <50k new prefixes per year; hence, one could limit the maximum daily growth to, e.g., at most 10k prefixes.

Closely monitor the number of sessions that other ASes establish with you---especially if your peering policy is fully open or you employ a fully automated session establishing service.

Given that the attack model is highly distributed, the best position to install protection mechanisms is your route reflectors, as they often have a complete view of the globally redistributed routes. If possible, implement the following two limiters:

(i) ensure that you only accept and redistribute a certain number of routes per origin AS

(ii) ensure that you only accept and redistribute a certain number of more-specific routes for each assigned address block.

(iii) accept only what is correctly registered. Do not allow an automatic "or longer" for any registered prefix. This will not prevent the attack but add more effort on the attackers' side to register the resources correctly.

(iv) monitor your generated filter size. A simple check on the number of acceptable prefixes can reveal the preparation of such an attack.

If you have any further questions, please don't hesitate to contact me!

Figure 9: Private disclosure notification email text.