USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

BGP Vortex: Update Message Floods Can Create Internet Instabilities

Felix Stöger, *ETH Zurich;* Henry Birge-Lee, *Princeton University;*
Giacomo Giuliari, *Mysten Labs;* Jordi Subira-Nieto and Adrian Perrig, *ETH Zurich*

https://www.usenix.org/conference/usenixsecurity25/presentation/stoeger

This paper is included in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.

BGP Vortex: Update Message Floods Can Create Internet Instabilities

Felix Stöger
felix.stoeger@inf.ethz.ch
ETH Zurich

Henry Birge-Lee
birgelee@princeton.edu
Princeton University

Giacomo Giuliari
giacomo@mystenlabs.com
Mysten Labs

Jordi Subira-Nieto
jordi.subiranieto@inf.ethz.ch
ETH Zurich

Adrian Perrig
adrian.perrig@inf.ethz.ch
ETH Zurich

Abstract

The Border Gateway Protocol (BGP), while essential for Internet connectivity, faces many stability and convergence challenges in today's evolving routing ecosystem.

In this paper, we present the discovery of the *BGP Vortex*, a configuration where just three legitimate BGP UPDATE messages can trigger persistent instability. We demonstrate that this vulnerability can be weaponized as an attack vector, potentially causing widespread Internet connectivity issues through router overload and forwarding loops. Crucially, a BGP Vortex cannot be prevented by existing security mechanisms such as BGPSEC or RPKI, because the protocol messages involved are legitimate. All major router implementations we could experiment with are susceptible to this threat.

At its root, the BGP Vortex is caused by standards-compliant BGP extensions—BGP Communities in this case—that allow the modification of route preferences for traffic engineering purposes. Therefore, to aid the mitigation of this attack as well as its potential future variations, we propose a framework to determine which BGP extensions are problematic, and which are safe to deploy. Our findings highlight the need to carefully balance network operators' traffic engineering capabilities with routing stability requirements.

1 Introduction

The Border Gateway Protocol is a critical component of the Internet infrastructure, as it performs inter-domain routing among autonomous systems (ASes). BGP populates the border routers' inter-domain forwarding tables, enabling global Internet connectivity. Given BGP's central role in connecting the world, its reliability and resilience to attacks are paramount.

One key issue for BGP is routing *convergence*, whereby the Internet's routing tables settle into a stable and correct forwarding state after a transient period of instability. During instability, routing loops and incorrect routes can appear, leading to dropped traffic and router overload; therefore, rapid convergence is essential. While BGP does not, in the general case, provide any convergence guarantee, Gao and Rexford show in their seminal work [23] that the economic relationships between Internet service providers (ISPs) shape their routing policies in such a way that convergence can be assured in practice. Yet, the Internet routing ecosystem has since evolved, and today ISP operators have a large suite of tools to perform *traffic engineering*, shaping their ingress and egress traffic beyond what the simple peering relationships studied by Gao and Rexford allow.

In our efforts to study BGP's and BGPSEC's convergence properties and research on enhancing their scalability and security, we have discovered that such traffic engineering tools can be detrimental—if not fatal—to BGP routing convergence, with dire consequences to the Internet ecosystem as a whole. Specifically, in this paper we show how a sequence of just three BGP UPDATE messages[1] can induce persistent instability and prevent route convergence for the announced prefixes. We call this unfortunate configuration a *BGP Vortex*, as the ASes involved are forced to indefinitely forward BGP UPDATE messages to each other.

The discovery that a sequence of three standards-compliant messages can cause persistent instability is already a sufficient cause for concern. Unfortunately, we also demonstrate that the BGP Vortex can be used as a powerful attack vector. Under the right conditions, a malicious AS can in fact craft multiple independent BGP Vortices, compounding their effect. Since the adversary can use its own prefixes and does not rely on hijacking, unfortunately neither BGPSEC nor RPKI can prevent a BGP Vortex attack.

The effects of a BGP Vortex attack can be severe: The resulting flood of UPDATE messages creates a variety of problems that can cause wide-spread connectivity outages. Even under normal circumstances, routers can reach processing limits, after which they may drop incoming UPDATEs.[2]

[1] A BGP UPDATE message announces an IPv4 or IPv6 address range, called a *prefix*, and contains the list of ASes on the path to that prefix. Through such UPDATE messages, BGP discovers routes or paths to destinations.

[2] This problem caused, e.g., the Rogers outage [36], which was caused by

This problem is exacerbated by high-end routers with limited processing capacity due to underpowered CPUs for general computation such as the BGP routing process [6, 19]. Overloaded routers can drop or ignore incoming BGP UPDATE messages, leading to outages for a given destination if no other path is available, or, even worse, if a BGP WITHDRAW message is dropped, traffic toward the affected destination will continue to be routed to a potentially failed link [41]. Further, forwarding tables can be left in inconsistent states across different border routers, which often results in intermittent forwarding loops. Forwarding loops prevent reachability and congest links, with a doubly-detrimental effect.

Our ethical analysis of the BGP Vortex threat shows that BGP Vortices can indeed be triggered in the wild, and that with only 21 ASes an adversary could construct up to 340 vortex configurations—which are then multiplied by the number of prefixes the adversary can announce—possibly creating floods of thousands of updates per second. In controlled lab experiments, we show that all major router software we could test—Cisco XRv9000, BIRD, and FRR— can fall into persistent instability due to a BGP Vortex, and that an adversary can inject multiple IP Prefixes into a BGP vortex to delay convergence, overload routers, and cause data-plane outages.

As we point out in Section 3.5, mitigating the BGP Vortex is challenging, as the injected messages are legitimate BGP UPDATE messages. We identify the root cause of the problem in the traffic engineering tools—BGP communities, in this case—that allow an AS to modify the local preference of its routes in neighboring ASes. We propose a classification of BGP communities, considering which ones can be safely adopted to avoid attacks such as the BGP Vortex.

In conclusion, we make the following **contributions:**

- We discover the BGP Vortex, a persistent instability in the Internet routing process that can be triggered with just three legitimate BGP UPDATE messages, even though all autonomous systems' policies are Gao-Rexford compliant;
- we demonstrate that BGP Vortices can occur in the wild, and that an adversary could leverage them to create UPDATE floods;
- in an isolated testbed, we measure the disruptive power of BGP Vortex attacks: Even in a small topology with 7 ASes, where convergence should be almost instantaneous, the attack can delay the propagation of UPDATES for tens of seconds, causing complete outages for the targeted prefixes;
- finally, we propose an analysis of the causes of the BGP Vortex, and derive a set of recommendations to prevent the occurrence of BGP Vortex and similar instabilities in the Internet.

core router overloading, or the recent global Microsoft Cloud outage [35].

2 Background

The Border Gateway Protocol (BGP), specified in RFC4271 [42], is the de-facto standard inter-domain routing protocol, that enables networks to exchange reachability information for IP prefixes across the Internet. BGP operates between autonomous systems (ASes), which are independently administered networks that collectively form the Internet's infrastructure.

In BGP, each AS advertises routes to its neighbors. A route consists of a destination IP prefix and its associated AS path—the sequence of ASes through which the advertisement has propagated. When an AS receives a route advertisement, it is processed according to AS-local routing policies. This processing includes assigning a local preference value to the route and performing a best-path computation to determine if the new route should replace the currently installed route for that destination. When an AS updates its preferred route, it appends itself onto the route's AS-path and propagates this new route to adjacent ASes.

BGP Convergence Properties. In general, BGP does not inherently guarantee convergence, and careful design of routing policies is required to ensure routing stability and convergence. Early work on BGP convergence behavior describe unstable routing policies and topology configurations—so-called "gadgets"— that cause oscillations and prevent convergence, thus leading to reduced end-to-end performance and loss of reachability [27, 49].

However, seminal work by Gao and Rexford proposed practical constraints on BGP policies that prevent the construction of undesirable gadgets and thus guarantee convergence [23]. The authors classify AS relationships in the Internet into three categories: (i) customer-provider, where the provider offers paid transit services to its customers, (ii) peering, where ASes route traffic originating from their respective customers free of charge, and (iii) backup, which are used in case of failure. The model further assumes that routes received from peers or providers are not redistributed to other peers or providers. That is, an AS does not provide transit services to its providers or peers.

In this model, convergence is guaranteed if an AS in its best-path computation strictly prefers routes received from customers over routes received from peers, and does not strictly prefer routes received from peers over routes received from customers. In practice, this is achieved by ensuring $\texttt{LOCAL\_PREF}_{\texttt{CUST}} > \texttt{LOCAL\_PREF}_{\texttt{PROV}}$ and $\texttt{LOCAL\_PREF}_{\texttt{CUST}} \geq \texttt{LOCAL\_PREF}_{\texttt{PEER}}$.

While the model and the policy rules are believed to be widely upheld as they align with business interests of ASes, new and more nuanced AS relationships like hybrid relationships, partial transit, and indirect peering have been developed [25].

BGP Communities. BGP communities, introduced in RFC1997 [33], are a mechanism that allows operators to attach additional metadata to BGP routes, thus enabling more complex routing policies. A BGP route can carry multiple community values, which can either be: standardized "well-known" communities specified in RFC1997 and follow-on RFCs like RFC1998 [13], or custom communities that ASes can freely define. Importantly, many ASes accept communities from customers that change the export behavior or local preference of those customers' routes [14]. These communities are often non-standard and are documented in the routing policies of networks that support them. By attaching these behavior-altering BGP communities to their prefix, customers can impact the propagation of their prefix through the Internet.

3 The BGP Vortex

In this section, we introduce the BGP Vortex, a phenomenon in which three interconnected ASes get trapped in a state of persistent route oscillations. These oscillations not only overload routers of ASes in the Vortex, but they cause a surge of route advertisements that is disseminated across the internet and possibly overloads further routers of ASes not part of the Vortex. We begin with a high-level overview of a BGP Vortex, and a discussion on the motivations and required capabilities for an adversary to create a BGP Vortex. Next, we provide a detailed functional description of a BGP Vortex. We conclude by examining why existing BGP stability mechanisms do not protect against BGP Vortex.

3.1 BGP Vortex: A High-Level Overview

The BGP Vortex is, in essence, a setting with three interconnected ASes that are in a state of persistent route oscillations. The oscillations are triggered when each AS receives a specially crafted route advertisement containing the local preference lowering and selective NOPEER BGP communities for an IP prefix. These BGP advertisements induce persistent circulation of further route advertisements among the three ASes without convergence. These circulations cause a continuous surge of BGP route advertisements to be propagated across the Internet. Notably, the BGP route advertisements that trigger the oscillations are fully compliant with the BGP specification. Since a BGP Vortex can be triggered by a single misconfigured or compromised BGP speaker, it represents a significant threat to Internet routing stability and it is ripe for exploitation. We now describe how a BGP Vortex can be generated in an adversarial setting, the motivations and capabilities needed to do so, and a detailed functional description of a BGP Vortex. This section concludes with a discussion on why existing BGP stability mechanisms are insufficient to prevent BGP Vortices.

3.2 Threat Model

We consider an adversary aiming to disrupt data plane connectivity by inducing delays and routing inconsistencies in the BGP control plane. The adversary has control over a single BGP speaker, for example as a malicious network operator or through exploitation of router vulnerabilities [4, 5, 7]. The compromised speaker can be located anywhere in the Internet, provided the local topology allows for creating a BGP Vortex as explained below.

The adversary is limited to advertising routes for prefixes legitimately owned by the compromised speaker's AS. All operations it performs comply fully with the BGP specification [42]—we explicitly exclude capabilities like sending malformed packets or hijacking prefixes owned by other ASes. This constraint makes this attack particularly stealthy, as it uses only standard BGP operations that are inconspicuous.

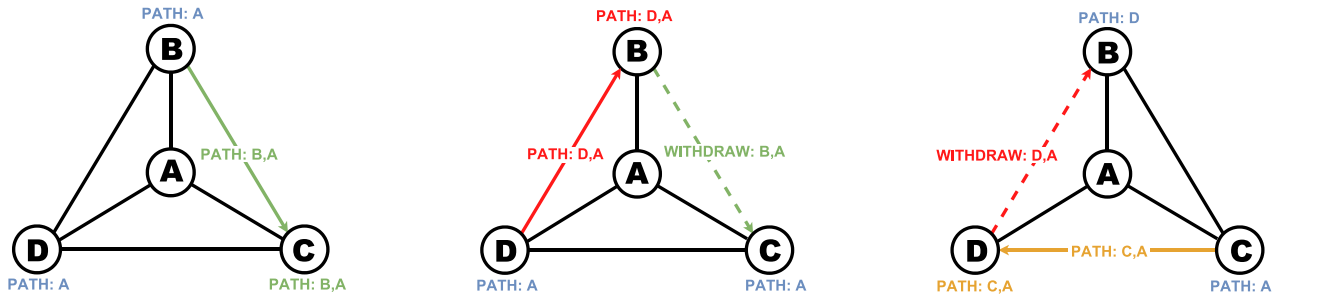
3.3 BGP Vortex Network Topology

A BGP Vortex, illustrated in Fig. 1, consists of three interconnected ASes: **B**, **C**, and **D**, that are interconnected as peers, forming a triangle. They are providers to an additional AS **A**, that is a customer of all three ASes and contains the adversary-controlled BGP speaker. The three provider ASes support both the local preference lowering and selective NOPEER communities. We assume these providers follow standard routing practices that comply with the Gao-Rexford constraints [23] described in Section 2, in particular that routes received from peers are only redistributed to customers. Additionally, like most ASes, they allow customers to advertise a sizable number of prefixes (commonly up to 100).

3.4 Selective NOPEER and Local Preference Lowering Induce Persistent Oscillations.

The adversary forms the BGP Vortex among ASes **B**, **C**, and **D** by sending each AS a route advertisement for a prefix owned by the AS of the adversary's BGP speaker. These advertisements contain two specially chosen BGP communities: (i) the selective NOPEER community, which restricts ASes **B**, **C**, and **D** to only redistribute the adversary's route to one of their two peers in the BGP Vortex, and (ii) the local preference lowering community, which causes them to consider the route to be less preferred in their BGP decision process. By reducing the local preference value, this community tag ensures that each AS prefers an indirect route through their peer over the direct route from the adversary.

These communities create a dependency cycle where each AS continuously toggles between the direct route from the adversary and an indirect route through one of its peers. When an AS receives a route from its peer, it prefers and installs this route over its direct route due to the local preference lowering community. However, this causes the AS to withdraw its



(a) AS **B** redistributes its direct route to AS **C**. AS **C** prefers it over the direct route and installs it in its routing table.

(b) AS **D** redistributes its route to AS **B**. AS **B** prefers this route and withdraws its previous direct route from AS **C**, causing it to fall back to its direct route.

(c) AS **C**, having fallen back to its direct route, redistributes it to AS **D**. AS **D** prefers this route and withdraws its previous direct route from AS **B**.

Figure 1: Visualization of the three main stages of a BGP Vortex.

previous route from its other peer, which then falls back to its direct route and redistributes it, thus continuing the cycle. We now provide a step-by-step explanation of this process in which we show how the adversary configures the communities and how this leads to persistent route oscillations among the ASes of the BGP Vortex.

Setup Phase. Initially, none of the provider ASes (**B**, **C**, and **D**) have a route for the adversary's prefix. AS **A** advertises a route for its prefix to all three providers, including the local preference lowering and selective NOPEER communities. The selective NOPEER communities are set to enforce clockwise redistribution:

- AS **B** is instructed to redistribute the route only to AS **C**.
- AS **C** is instructed to redistribute only to AS **D**.
- AS **D** is instructed to redistribute only to AS **B**.

Each provider installs the direct route from AS **A** in their routing tables, as they have no alternative routes.

Step-by-Step Oscillation Mechanism.

1. **AS B Redistributes to AS C**
   AS **B**, following the policies and communities, redistributes the route received from customer AS **A** to its peer AS **C**. Due to the local preference lowering community, AS **C** prefers this new route from AS **B** over its direct route from AS **A** and updates its routing table accordingly (Fig. 1a).
2. **AS D Redistributes to AS B**
   Next, AS **D** redistributes its direct route to AS **B**, as permitted by the selective NOPEER community. AS **B** receives this route and, due to the local preference lowering community, prefers it over the route from AS **A**. AS **B** updates its routing table, withdraws the previous (direct) route from AS **C** (Fig. 1b), however it does not redistribute the new route since routes from peers are not redistributed to other peers according to our Gao-Rexford assumption. AS **C**, upon receiving the withdrawal, reverts to its direct route from AS **A**.
3. **AS C Redistributes to AS D**
   AS **C**, now using the direct route from AS **A**, redistributes it to AS **D**. AS **D** prefers this route over its existing one due to the local preference lowering community and updates its routing table, withdrawing its previous route from AS **B** (Fig. 1c). AS **B**, upon receiving the withdrawal, reverts to its direct route to AS **A**.

Persistent Oscillations This process creates a loop of route advertisements and withdrawals among the provider ASes, which continues indefinitely:

- AS **B** redistributes the direct route to AS **C**, causing AS **C** to prefer the route from AS **B** and withdraw from AS **D**.
- AS **D** redistributes to AS **B**, causing AS **B** to prefer the route from AS **D** and withdraw from AS **C**.
- AS **C** reverts to the direct route and redistributes it to AS **D**, causing AS **D** to prefer the route from AS **C** and withdraw from AS **B**.
- AS **B** reverts to the direct route from AS **A** and the cycle repeats.

By exploiting the BGP communities and the providers' routing policies, the adversary causes the providers to continuously prefer routes from their peers over the direct route from AS **A**, leading to persistent oscillations and a surge of BGP UPDATE messages.

3.5 Evading BGP Stability Mechanisms

In this section, we discuss why existing mechanisms designed to prevent oscillations and surges of BGP UPDATEs are either ineffective or insufficiently deployed to mitigate our attack. We analyzed the three most widely known mechanisms: route aggregation, MinRouteAdvertisementInterval (MRAI), and Route Flap Damping (RFD).

Route Aggregation. Route aggregation aims to reduce BGP control plane traffic by combining multiple route advertisements with identical path attributes into a single advertisement. However, this mechanism is easily circumvented by ensuring each route has unique path attributes, which can be achieved through common traffic engineering practices such as AS-path prepending. Since these techniques are widely used by network operators for legitimate purposes, their use will not raise suspicion.

MRAI Timers. The MinRouteAdvertisementInterval (MRAI) timer, specified in RFC4271 [42], is a rate-limiting mechanism that restricts how frequently a BGP speaker can send route advertisements or withdrawals to each neighbor. Commercial BGP implementations by Cisco [3] and Nokia [8], and open-source implementations Quagga [1] and FRR [21], implement MRAI using a single per-neighbor timer. In such designs, all route advertisements and withdrawals to a neighbor are subject to the same timer—they are buffered until the timer expires before being forwarded. While this approach simplifies implementation, it can significantly delay network convergence since critical route updates may be buffered behind less important ones.

The choice of appropriate MRAI timeout values significantly impacts route stability and network convergence times. While RFC4271 [42] initially recommended a 30-second timeout, this value has since been deprecated in favor of leaving the choice to network operators and vendors. Network operators largely prioritize faster convergence over routing stability, with a survey by Gill et al. [24] finding that over 90% disable MRAI timeouts entirely. This trend is reflected in major implementations: Juniper JunOS [30] and FRR [21] disable MRAI by default, Cisco IOS disables it for BGP sessions inside VRFs [3], and BIRD [20] does not implement it at all. While MRAI could theoretically mitigate the BGP Vortex attack by slowing down route oscillations, its limited deployment and strong operator opposition make it an impractical defense mechanism.

Route Flap Damping. Route Flap Damping (RFD) is a mechanism designed to reduce the impact of route flapping by damping BGP route advertisements for unstable IP prefixes. Under RFD, a BGP session maintains a penalty score for each destination prefix. This penalty increases additively when advertisements or withdrawals are received and decays exponentially over time. When a prefix's penalty exceeds a configured suppress-threshold, the prefix is damped— withdrawn from all neighbors and not redistributed until its penalty falls below a reuse-threshold.

The configuration of RFD presents significant operational challenges. Prior work has shown that suboptimal parameter choices can impair network convergence by inadvertently suppressing legitimate route advertisements [34]. These challenges have led to evolving recommendations over time, with RIPE even recommending to disable RFD entirely [15, 18, 38, 39, 45]. While recent RIPE and IETF guidelines recommend less aggressive damping parameters in favor of disabling RFD, its deployment remains limited. Prior work has found a lower bound of only approximately 8% of ASes using RFD, with most using deprecated parameters. Major router vendors, including Cisco [2], Juniper [9], Nokia [8], as well as open-source router implementations FRR [21] and Quagga [1], disable RFD by default. FRR and Quagga explicitly discourage its use, while BIRD does not implement the feature at all.

4 BGP Vortices in the Wild

In previous sections, we discussed the theory behind BGP Vortices, their potential as attack vectors, and how existing mitigations are insufficent to stop them effectively. Yet, BGP Vortices require a specific set of topological conditions to be viable as an attack vector: Three ASes supporting the right BGP communities must be peering with each other, and they must be providers to the adversary's AS.

In this section, therefore, we investigate the prevalence of such conditions in the Internet, and show that there are indeed many sets of ASes in the Internet that satisfy the preconditions for BGP Vortices. Further, we ethically probe these ASes with BGP route advertisement messages, showing that an adversary would be able to trigger these BGP Vortices in practice.

4.1 Statically Identifying BGP Vortices

As a first step, we analyze the Internet topology to identify the presence of BGP Vortices.

Methodology. To find BGP Vortices, we need to (i) find ASes that support the selective NOPEER and local preference lowering communities, and (ii) find sets of three ASes among these that are peering with each other.

Unfortunately, information on the BGP communities supported by ASes in the Internet is scarce. We use the data collected by a previous publication [14], which lists the communities supported by the top 30 ASes. [3] Out of these, 21

[3] The top 30 selected as per as-rank.caida.org, at the time of publication of the survey (March 2019).

ASes support both the selective NOPEER and local preference lowering communities. Then, we leverage this information and run a simple algorithm on the CAIDA AS relationship graph [4] to identify the presence of BGP Vortices: For every AS supporting the selective NOPEER and local preference lowering communities, we check whether it has two AS peer that also support these communities and that peer with each other. If so, we have identified a BGP Vortex. We finally count the number of unique BGP Vortices identified in this way, as well as the number of Vortices that each AS is a part of. This latter measure is a proxy for the potential amplification that an adversary may achieve in targeting a single AS, as the adversary may trigger all Vortices the AS is part of in parallel.

Results. Among the 21 candidate ASes we identified, we find 340 viable BGP Vortices. For comparison, there are 1119 peering triangles between the 30 top-tier ASes, meaning that BGP Vortices are present in 30.4 % of all possible peering triangles in this subset of the Internet topology.

Then, we consider individual ASes, and count the number of BGP Vortices they can be a part of. We find that ASes are contained in 58 Vortices in the median, with one AS being part of as many as 86; only one AS in not part of any BGP Vortex.

Although this analysis is constrained by the publicly available information on supported BGP communities, and therefore considers only a small subset of well-connected top-tier ASes, the results suggest that BGP Vortices are not a rare occurrence in the Internet.

4.2 Ethically Probing Vulnerability of Tier-1 ASes

We confirmed the behavior required to initiate a BGP Vortex (in an ethical manner) among three Tier-1 ASes. To do this, we confirmed the Tier-1 providers' support for each of the individual communities required to initiate the BGP Vortex separately but did not use all of the communities at the same time. We made BGP announcements from the cloud provider Vultr for our own prefix space tagged with the relevant communities and then confirmed the behavior of the communities by observing route propagation in public looking glasses.

The BGP Vortex we confirmed was made up of the ASes: NTT (AS 2914), Arelion (AS 1299, fka. Telia) and Lumen (AS 3356, fka. Level3). The communities we used were found in public routing guides [12, 37, 46]. We confirmed support for the following communities at these ASes:

- NTT: Lower local preference below peer (2914:450), No export Level3 (65500:3356)
- Arelion: Lower local preference below peer (1299:50), No export NTT (1299:2529 and 1299:5529 used concurrently)
- Lumen: Lower local preference below peer (3356:70), No export Arelion (65000:1299)

Utilizing these communities in combination would create a BGP Vortex between these providers. To avoid introducing instability into the global routing system, we did not use all of these communities concurrently and confirmed their behavior one-by-one. However, we did use all of the local preference lowering communities together and confirmed that this lead to an unstable equilibrium that was dependent of the order of BGP advertisements. This instability combined with the selective NOPEER communities (that we confirmed functioned separately) introduces the osculating behavior seen in our BGP Vortex.

4.3 Update Floods in the Customer Cone

Having established that BGP Vortices are indeed present in the Internet, and that they can be triggered in practice with just three route advertisement messages, we now consider their reach and impact on the Internet topology.

Even though a BGP Vortex is composed of three ASes, the advertisements generated by the BGP Vortex are propagated further into the network, and therefore can cause high route churn in ASes that are not directly part of the BGP Vortices. In particular, ASes in the *customer cone* of the BGP Vortices are bound to receive all the advertisements generated by their providers under attack.

Therefore, we study how many ASes are in the customer cone of the BGP Vortices we identified in the previous analysis, and how many BGP route advertisements they would receive in case the adversary would trigger one or more Vortices in parallel.

Methodology. We define a metric to quantify the amount of churn generated by BGP Vortices on customer ASes.

Recall that in a BGP Vortex, ASes keep oscillating their routes—generating advertisement and withdrawal messages for the prefix the adversary is advertising—indefinitely. These BGP messages are generated in a cycle, where each AS sends one advertisement and one withdrawal message for every revolution of this cycle. We define the *period* of the Vortex as the time between two identical routing states of the Vortex, i.e., the time it takes for each of the three ASes to update their routes and then cycle back to the initial (unstable) state.

If we assume, for simplicity, that the processing and propagation of these BGP messages are similar across all BGP Vortices, we then have that the "period" of the BGP Vortices is constant. Thus, each AS produces exactly one BGP advertisement message per period, for each BGP Vortex it is part

[4] The CAIDA AS Relationships Dataset, March 1st 2024 version, https://www.caida.org/catalog/datasets/as-relationships/

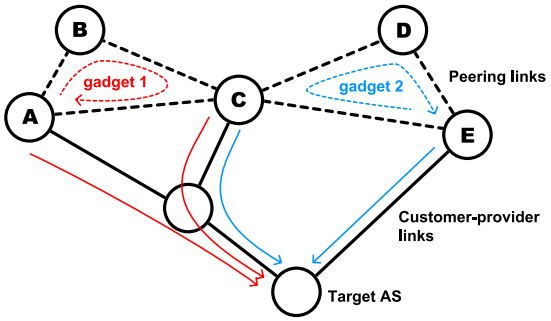


Figure 2: Example of the number of route advertisements per period received by a customer of two BGP Vortices. Each of the ASes in the two BGP Vortices (A-B-C, C-D-E), generates one route advertisement message per period. These messages are then propagated to customers, reaching the target AS. Since C is part of both Vortices, it generates two advertisement messages per period. The solid arrows indicate the route advertisements received by the target AS.

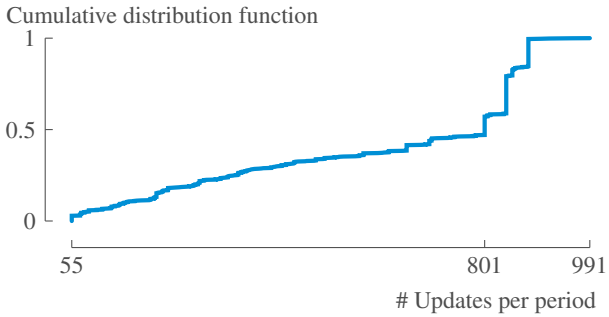


Figure 3: The distribution of the number of updates per period seen by customer ASes, if all 340 BGP Vortices are triggered in parallel (one prefix per Vortex). The min, median, and max number of updates per period are highlighted.

of, and for each prefix the adversary advertises in the Vortex. This simplification allows us to describe the route churn experienced by and AS in terms of *advertisements per period*. This concept is illustrated in Fig. 2. In practice, ASes may be more or less reactive to route advertisements, leading to BGP Vortices with different periods.

Our goal is then to compute the number of route advertisements per period seen by a customer AS of each of the 340 BGP Vortices we identified in Section 4.1. We thus perform the following steps:

- First, consider the set of 21 origin ASes, O, supporting the selective NOPEER and local preference lowering communities. Using the CAIDA AS relationship graph, we compute the customer cone of each AS i, $i \in O$.
- Then, we annotate each AS c with the set $\mathcal{P}_c \subset O$ of origin ASes which have AS c in their customer cone. E.g., in Fig. 2, target AS has $\mathcal{P}_T = \{A, C, E\}$.
- Finally, we define g_i as the number of BGP Vortices each origin AS i is part of, and use it to compute the number of route advertisements per period seen by each customer AS in different scenarios. The key observation is that each as in the customer cone of an origin AS i, will see *at least* g_i UPDATE messages originating from AS i per period.[5] Then, we compute a lower bound on the total number of UPDATEs per period, p_c, seen by each customer AS c as:

$$p_c = \sum_{i \in \mathcal{P}_c} g_i. \quad (1)$$

In Fig. 2, ASes A and E are part of one BGP Vortices each, and AS C is part of two Vortices. Thus, if both Vortices are activated, the target AS receives four updates per period.

Results. Of the 76 564 ASes present in the CAIDA dataset, 73 743 (more than 96 %) are in the customer cone of at least one of the 21 ASes we identified as possible BGP Vortex targets. This is unsurprising, since these ASes are selected from the top 30 ASes in the AS rank.

Yet, this increases the potential impact of an attack based on BGP Vortices: If all 340 BGP Vortices are triggered in parallel, a very large fraction of the ASes in the Internet will see a significant increase in the number of UPDATE messages they receive. Figure 3 shows the distribution of the number of updates per period seen by customer ASes in this scenario: The median is 801 updates per period, with a minimum of 55 and a maximum of 991.

Finally, note that these results only consider one prefix per Vortex. The adversary can multiply the number of updates per period simply by announcing more prefixes on every BGP Vortex. Then each new prefix will be propagated in parallel, easily achieving a 100 × increase[6] in the number of updates seen by customer ASes. Since each prefix can induce up to ≈ 1000 updates on each AS, victims may receive tens of thousands of updates per period.

[5] Some ASes may see the same update message multiple times, because of different paths that converge onto them.However, ASes will likely discard these duplicate messages. Therefore, we conservatively consider the lower bound of update messages received. We further assume that the ASes in the CAIDA map are in the default-free zone, and therefore receive updates for all prefixes from their neighbors.

[6] The 100 × figure here is mentioned because, in most standard peering agreements, announcing up to 100 prefixes is allowed without extra paperwork. Yet, a motivated adversary can circumvent this 100 prefix limit to achieve multiple times this factor, for example by using multiple upstreams or peering in several locations.

Discussion. To put the results above in perspective, a recent report [28] shows that, in 2024, the APNIC R&D Center AS (AS 131072) received around 200 000 BGP updates per day, or 2.3 per second.[7] Thus, the fact that a single BGP Vortex attack, based only on 21 ASes, can induce tens of thousands of updates *per period* highlights the potential impact a BGP Vortex attack can have on the global routing system. Clearly, then, the practical impact of the abstract results described above depends on many factors, but most importantly:

- the length of the period of the BGP Vortex, which determines how quickly the updates are generated; and
- the ability of the border routers of ASes to process and propagate the volume of generated updates. Updates may be dropped because the router ingress queues are overwhelmed, or because, in some router configurations, conflicting updates for the same prefix may be dropped in favor of the most recent one.

To concretely answer the first question on the length of the period, we anticipate here a result from the following experiment section: A BGP vortex composed of three FRR routers can oscillate with a period of 0.15 seconds (6.6 periods per second), while a BGP Vortex composed of BIRD routers—which are geared towards immediate announcement propagation—oscillates with a period between 0.025 and 0.0227 seconds (40 periods per second). Multiplied by the number of updates per period, this leads to, in theory, between thousands and hundreds of thousands of updates per second. In the following sections, we further explore these two metrics in a variety of deployments with real router implementations.

4.4 Topology Location of the Adversary

Multi-homing ASes Many ASes in the CAIDA topology have three or more providers making them candidates to launch the BGP Vortex. We found that out of all 77251 ASes in the CAIDA AS-relationship data set [8], 17976 of them have more than 2 providers, and 2410 ASes have at least 3 of the 21 vulnerable ASes as providers. Many ASes multihome to improve connectivity and redundancy, providing resilience against network outages. Even smaller ASes commonly multihome—for instance, our affiliated institution's network provider maintains three tier-1 upstreams. Historically, multihoming was a requirement for obtaining an independent ASN [48], making widespread multihoming an expected characteristic of the Internet topology.

Additionally, while we present the attack in a multihomed setting for clarity, direct multihoming to three vulnerable providers is not strictly required. First, an adversary with control at the AS level could register new providers for the purpose of launching this attack (an adversary could even register a new ASN for this purpose as well). Second, an adversary that has fewer than three providers can potentially launch the BGP Vortex using transitive communities, as detailed in Appendix A.

Use of non-stub ASes The attack works regardless of whether the adversary controls a router in a stub or non-stub AS and whether the ASes in the BGP Vortex have additional connections. This is because in both cases, the adversary can selectively send BGP announcements only to those providers needed to form the vortex, irrespective of any additional neighboring ASes. Moreover, although an AS in the vortex may receive advertisements originating from a BGP Vortex through other BGP neighbors, these advertisements are unlikely to disrupt the cycling of the vortex. This is because the Gao-Rexford routing policies prevent announcements originating in the BGP Vortex (consisting of Tier-1 ASes) from being redistributed back into the vortex over customer to provider and peer to peer sessions. Additionally the BGP announcements that fuel the vortex are stable and our results in Section 5 show that stable BGP announcements are the least likely to be interrupted by the vortex.

5 Experimental Evaluation

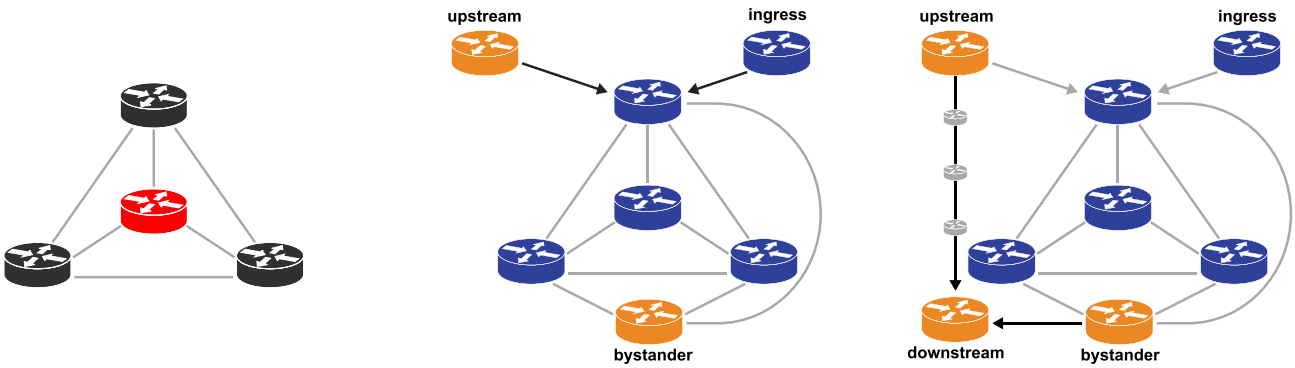
In this section, we demonstrate the susceptibility of widely deployed commercial routers to persistent route oscillations caused by a BGP Vortex, and we show how just a single such Vortex can significantly impact control plane convergence, router processing load, and data plane availability. We present the following four key findings: (i) all tested router implementations are vulnerable to persistent route oscillations caused by BGP Vortex, (ii) even a single BGP Vortex can delay legitimate route advertisements by up to 40 seconds, significantly impacting network convergence, (iii) the surge of route advertisements generated by a BGP Vortex can overwhelm routers, causing their BGP instances to not be able to keep up with incoming route advertisements, and (iv) the induced routing inconsistencies can cause outages in the data plane. These findings highlight that BGP Vortices pose a significant threat to Internet routing stability.

5.1 Experiment Infrastructure

We built a controlled lab environment to evaluate the impact of a BGP Vortex on routing stability. Our testbed allows us to measure how a BGP Vortex affects network convergence time, router load, and data plane availability in a repeatable manner. While real ASes can consist of many BGP speakers, we model each AS by a single BGP speaker since our focus is on the inter-AS propagation of BGP routes rather than AS-internal dynamics. However, investigating potential intra-AS

[7] Anecdotally, this is a high number of updates, as this research AS is peering with as many other ASes as possible to provide a good observation point for Internet research.

[8] The CAIDA AS Relationships Dataset, April 1st 2025 version, https://www.caida.org/catalog/datasets/as-relationships/



(a) Network topology used to verify that common router implementations are susceptible to persistent oscillations induced by the BGP Vortex. Three ASes (dark gray routers) form the BGP Vortex, while a fourth AS (red router) acts as an adversary-controlled BGP speaker.

(b) Network topology for evaluating route propagation delays. The *upstream-AS* advertises routes that traverse the BGP Vortex (blue routers running BIRD) to reach the *bystander-AS* (orange routers running FRR). This setup allows measuring the latency impact of BGP Vortex on route propagation.

(c) Network topology for evaluating data plane availability impact of the BGP Vortex. The *downstream-AS* has two paths to reach the *upstream-AS*: a path through both *bystander-AS* and BGP Vortex, and a direct path which we simulate to have three on-path ASes (gray routers) using AS-path prepending.

Figure 4: Network topologies used to evaluate the impact of a BGP Vortex: base topology demonstrating router susceptibility (a), topology measuring route propagation delays (b), and topology with two routes showing impact on data plane connectivity (c).

effects of a BGP Vortex is an interesting direction for future work.

Our testbed consists of two complementary setups. The first testbed utilizes the Cisco Modeling Labs 2 (CML2) network modeling platform to confirm that the Cisco XRv9000 Carrier-Grade virtual router indeed exhibits persistent oscillations in a BGP Vortex. However, since CML2 is designed for functional testing of network topologies rather than performance measurements, we built a second testbed as an overlay network in DigitalOcean's Frankfurt datacenter to measure the performance impact of a BGP Vortex. This second setup provisions a virtual machine (VM) running Ubuntu 22.04.2 LTS for each AS in the experiments. For each experiment, the ASes were configured to run one of two widely deployed open-source routers: FRR (version 9.0.1), and BIRD (version 2.0.8). VMs running BIRD had two dedicated vCPUs and 16GB RAM, while VMs running FRR had four dedicated vCPUs and 8GB RAM.[9]

We collect three types of measurements:

- TCP dumps to analyze BGP route advertisements at the network level;
- Router logs to measure processing delays within routers;
- iperf3 traces to detect data plane outages.

Our evaluation results are derived from over 500GB of raw experimental data which we analyzed using standard networking and data analysis toolkits.

[9]BIRD's BGP implementation is single-threaded, while FRR's BGP implementation uses two CPU cores: one dedicated to the BGP state machine and another to BGP I/O operations [21].

5.2 Routers are Susceptible to BGP Vortices

First, we verified that widely deployed router implementations are indeed susceptible to persistent route oscillations induced by a BGP Vortex. We tested three router implementations commonly deployed in production environments: Cisco XRv9000 Carrier-Grade virtual routers, FRR (version 9.0.1), and BIRD (version 2.0.8). Using our testbed, we were able to verify that a BGP Vortex comprising any of these routers indeed persistently oscillates.

Methodology. To investigate the routers' susceptibility to persistent route oscillations, we constructed a BGP Vortex, shown in Fig. 4a with four distinct setups: homogeneous setups using only Cisco XRv9000, FRR, and BIRD routers, as well as a heterogeneous setup combining FRR and BIRD routers.

We configured each router to support the necessary local preference lowering and selective NOPEER communities as described in Section 3.4. In particular, we configured the selective NOPEER community to prevent redistribution of routes inside the BGP Vortex to counter-clockwise peers, and the local preference lowering community to reduce the accompanying route's local preference below that of a peer-advertised route. These policies were implemented using route-policies for XRv9000, route-maps for FRR, and filters for BIRD.

Further, we measure the *period* of the BGP Vortex by measuring the time between two identical routing states across the three target ASes. We only measure the period in FRR and BIRD, as the Cisco XRv900 routers are running in emulation, and therefore their performance is not comparable.

Results. In all tested BGP Vortex setups, the routers persistently oscillate the route injected by their shared customer. This validates that the router implementations under scrutiny are indeed susceptible to the BGP Vortex.

Regarding the period, a BGP vortex composed of three FRR routers oscillates in the experiment with a period of 0.15 seconds (6.6 periods per second), while a BGP Vortex composed of BIRD routers oscillates with a period between 0.025 and 0.0227 seconds (40 periods per second).

5.3 BGP Vortices Delay Network Convergence

Next, we investigated the impact a BGP Vortex has on network convergence by measuring the delay a legitimate BGP advertisement incurs as it traverses (i) ASes caught in the Vortex, and (ii) ASes adjacent to the Vortex but within its customer cone, meaning in the customer cone of the ASes composing the vortex.

Methodology. To measure the route propagation delays through a BGP Vortex and its customer cone, we configure our testbed to include a BGP Vortex and three additional ASes outside the Vortex, as seen in Fig. 4b:

- *upstream-AS*: configured as provider of one AS in the Vortex. It sends out legitimate route advertisements.
- *bystander-AS*: a customer of all ASes in the Vortex.
- *ingress-AS*: injects both a full Internet route table and a live stream of BGP updates into the Vortex.

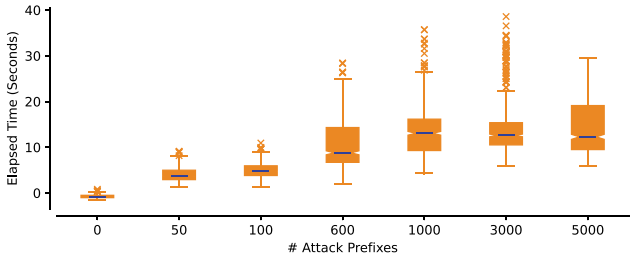
This topology allowed us to measure the increase in delay for legitimate BGP routes advertised by the *upstream-AS* as they (i) traverse through the BGP Vortex, and (ii) are processed by the *bystander-AS*. The addition of the *ingress-AS* accurately models the control plane load experienced by BGP speakers in the Internet core. We obtained a live BGP stream by peering with the Vultr AS (AS20473) in Frankfurt.

The ASes in the BGP Vortex and the *ingress-AS* deployed BIRD, while the remaining ASes deployed FRR. This configuraiton reflected an adversary who aims to to create a BGP Vortex with the lowest period by choosing ASes deploying the fastest routers implementations.

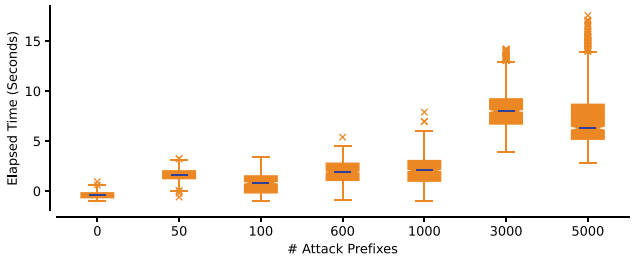
The experiment began with a 5-minute warmup period to allow the full Internet routing table from the *ingress-AS* to fully propagate and any buffers of routers in the Vortex to fill. Subsequently, the *upstrea-AS* advertised 50 legitimate prefixes with a two-second interval. We waited another 60 seconds for the system to settle before stopping the experiment.

We performed packet captures at the *upstream-AS* and the *bystander-AS*, and configured FRR to log processed BGP route advertisements. Subtracting the time when a route advertisement was sent by the *upstream-AS* from the time it arrived at the *bystander-AS* yields the propagation delay in the BGP Vortex. By subtracting the time a route advertisement arrived at the *bystander-AS* from when it was logged in the FRR's BGP log, we computed the processing time on the *bystander-AS*.

We repeated the experiment while varying two parameters: (i) Number of routes oscillating in the Vortex: 0 (baseline) to 5000; and (ii) Number of CPU cores available to the *bystander-AS*'s FRR instance.



(a) 1 CPU core available to *bystander-AS*' FRR instance.



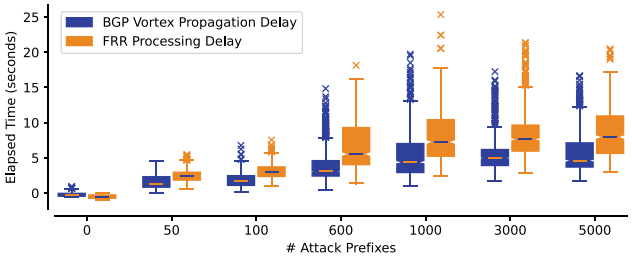
(b) 2 CPU cores available to *bystander-AS*' FRR instance.

Figure 5: End to end delay for legitimate route advertisements: propagating through the BGP Vortex and processing by *bystander-AS*' FRR instance when it is limited to (a) one CPU core or (b) two CPU cores.

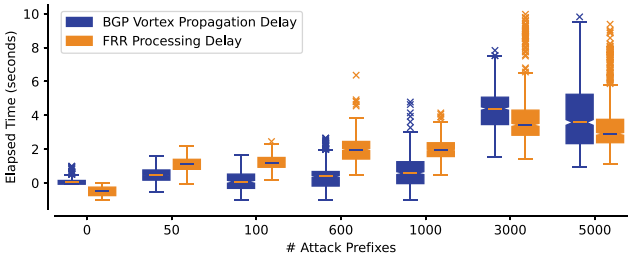
Results. The measurements demonstrate substantial end-to-end delays, that vary based on computational resources assigned to the bystander-AS (see Fig. 5): with a single CPU core available to the *bystander-AS*' FRR instance, route propagation delay reached up to 40 seconds, while the results with two CPU cores show an improved but still considerable delay of up to 17.5 seconds. This delay comprises two main components as seen in Fig. 6: (i) the propagation delay inside the BGP Vortex, and the processing delay on the *bystander-AS*' FRR instance. Both distributions are highly positively skewed with increasing number of routes oscillating in the Vortex. This indicates that ASes in the Vortex running BIRD and the *bystander-AS* running both fail to process route advertisements with predictable delays.

While the addition of a second CPU core to the FRR instance in the *bystander-AS* improves route processing delay, this improvement should be interpreted with caution.[10] Firstly,

[10] Attempting the experiment with three CPU cores showed only insignifi-



(a) 1 CPU core available to *bystander-AS*' FRR instance.



(b) 2 CPU cores available to *bystander-AS*' FRR instance.

Figure 6: Delay breakdown for legitimate route advertisements: propagation through the BGP Vortex (purple) and processing by *bystander-AS*' FRR instance (orange) when it is limited to (a) one CPU core or (b) two CPU cores.

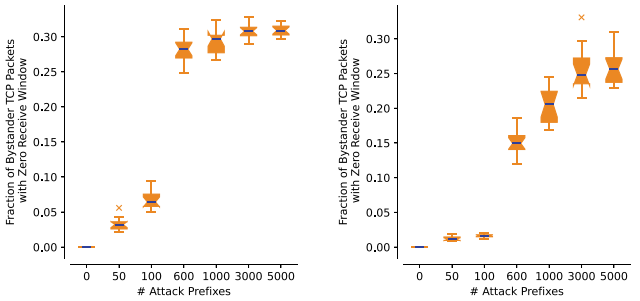
the BGP state machine is difficult to parallelize, thus likely limiting the possibility to scale by adding additional CPU cores as a mitigation strategy against BGP Vortex. Moreover, our lab environment provided idealized conditions for routers that deviate significantly from production deployments. In operational networks, routers have to manage a complex set of critical tasks in parallel: maintaining peering sessions with hundreds of BGP peers, RPKI verification, IGP protocols (OSPF/IS-IS), MPLS/RSVP-TE, implementing sophisticated route filtering policies, enforcing access control mechanisms, handling firewall rules, and collecting telemetry data. The difficulty to scale the BGP state machine and the extensive set of additional tasks not modeled in our controlled lab environment suggests that the performance improvements we saw by adding additional CPU resources represent an optimistic upper bound.

5.4 BGP Vortices Overload Routers

While analyzing packet captures for the previous experiment, we found that up to 33% of TCP packets sent by the *bystander-AS* to ASes in the BGP Vortex had a receive window size of zero. This indicates that the *bystander-AS*' FRR instance failed to read from its TCP receive buffers at the rate with which they were filled with (mostly bogus) route advertisements sent by ASes in the BGP Vortex.

cant improvements.

Methodology. From the packet captures collected in the previous experiment, we extracted all TCP packets sent by the *bystander-AS* to the routers in the BGP Vortex. Out of the extracted TCP packets, we computed the fraction which had a receive window size of zero.



(a) 1 CPU core available to *bystander-AS*' FRR instance

(b) 2 CPU cores available to *bystander-AS*' FRR instance

Figure 7: Fraction of TCP Packets sent by the *bystander-AS* that have a TCP receive window size of 0.

Results. Our analysis shows a significant increase in the fraction of TCP packets with a receive window size of zero (see Fig. 7). In the baseline scenario with no routes oscillating in the BGP Vortex, we observe no TCP packets with zero receive window size. With routes oscillating in the BGP Vortex, the fraction of zero-window packets increases drastically to more than 30% in both single-core and dual-core FRR configurations.

The surprisingly high fraction of TCP zero receive window sizes, despite increasing the number of CPU cores available to FRR, is worrisome since it may indicate an underlying I/O bottleneck. While extra cores can reduce route processing delays (see Section 5.3), this fails to address any underlying I/O bottleneck between kernel socket buffers and the user-space BGP implementation. How such a potential bottleneck could be exploited is an interesting research direction which we intend to explore in the future.

5.5 BGP Vortices Cause Data Plane Outages

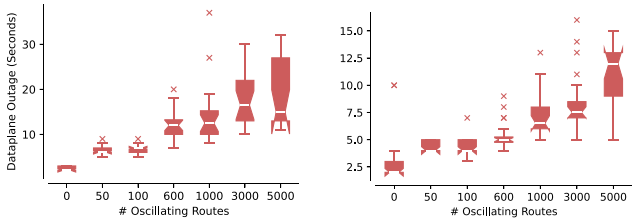
Finally, we studied how BGP Vortices can cause data plane outages during transient events such as route changes by delaying the propagation of critical BGP route updates.

Methodology. To measure the impact of a BGP Vortex on data plane availability during route changes, we extended then network topology from our previous experiment with a *downstream-AS*, which was configured as a customer of the *bystander-AS* and *upstream-AS*. The *downstream-AS* had two network paths to reach the *upstream-AS* as shown

in Fig. 4c: a path through the *bystander-AS* and BGP Vortex, and a direct path. We simulated the direct path as having three intermediate ASes by using AS-path prepending at the *upstream-AS*. This makes any route advertisement from the *upstream-AS* sent via the direct link appear longer (and thus less preferred) than routes through the *bystander-AS*. The *downstream-AS* thus always prefers routes using the path through the *bystander-AS*.

In the experiment, we simulated a route change at the *upstream-AS* and measured the duration of transient data plane unavailability between the *downstream-AS* and *upstream-AS* caused by the BGP Vortex. The experiment begins with a 5-minute warmup period similar to Section 5.3, during which the *upstream-AS* sends a route advertisement via both paths which are received by the *downstream-AS*. We then simulate a route change at the *upstream-AS*: the *upstream-AS* prepends itself twice to the route advertised through the BGP Vortex and *bystander-AS*, while simultaneously installing a firewall rule to drop incoming data plane traffic arriving via this path. This change makes the route using the direct link appear shorter and thus more preferred to the *downstream-AS*. However, the *downstream-AS* only learned of this route change once the corresponding advertisement sent by the *upstream-AS* had traversed the BGP Vortex and *bystander-AS*. Until then, any data plane traffic sent to the *upstream-AS* by the *downstream-AS* is still routed via the *bystander-AS* and subsequently dropped by the firewall rule. To measure data plane connectivity, we establish a continuous iperf3 UDP session between the *downstream-AS* and *upstream-AS*. We measure data plane unavailability as the time between the route change and when the *downstream-AS* switches to the direct path.

We repeated the measurements, varying two parameters: (i) Number of routes oscillating in the Vortex: 0 (baseline) to 5000; (ii) Number of CPU cores available to the *bystander-AS*' FRR instance.



(a) 1 CPU core available to *bystander-AS*' FRR instance

(b) 2 CPU cores available to *bystander-AS*' FRR instance

Figure 8: Duration of data plane outages in seconds.

Results. The results from this experiment, shown in Fig. 8, are consistent with and support the findings from Section 5.3. Notably, we observed an increase in the duration of transient data plane unavailability over the baseline (without any oscillating routes) of up to 37 seconds when the *bystander-AS*'s FRR instance was restricted to one CPU core, and 17 seconds for two CPU cores. These measurements are in agreement with the 40 second and 17.5 second increase in end-to-end delays we recorded in our earlier experiment. Until the *downstream-AS* had received and processed the advertisement for the route change, traffic generated by the iperf3 session was continuously routed via the *bystander-AS* and dropped by the firewall rule. This demonstrates that backup links fail to protect against transient data-plane outages due to critical route advertisements being delayed by the BGP Vortex.

6 Mitigations

We discuss mitigations that limit BGP Vortex effects and propose a framework for safe BGP community usage to address the underlying instability. In Section 8, we discuss alternative interdomain architectures that avoid convergence-based attacks entirely by eliminating iterative convergence processes.

6.1 Partial Mitigations

We propose several partial mitigations that can reduce the impact of a BGP Vortex, without addressing the underlying cause of the vortex. We advocate to use these partial mitigations only as temporary solutions until a full mitigation as discussed in the next section is deployed.

First, networks should deploy BGP routers with ample resources (particularly CPU and memory) to mitigate the damage of the BGP Vortex and reduce the risk of a router crash. Second, MRAI timers configured with minimum 1-second timeouts substantially reduce oscillation frequency, albeit at the cost of delayed network convergence. Finally, Route Flap Damping can limit prefix oscillations but risks inadvertently dampening legitimate routes. We refer readers to current best practices for RFD configuration [15].

6.2 Safe BGP Communities

BGP communities allow for the altering of an AS' import, export, or route selection policy which we exploit to cause instability in the BGP ecosystem. However, all of the flaws we exploit trace back to *communities that violate the Gao-Rexford routing model [23]*. Specifically, the Gao-Rexford paper proves stability under the following assumptions:

- **Exporting to a provider:** An AS can export its routes and the routes of its customers, but can not export routes learned from other providers or peers.
- **Exporting to a customer:** An AS can export its routes, as well as routes learned from its customers, providers, and peers.

- **Exporting to a peer:** An AS can export its routes and the routes of its customers, but can not export the routes learned from other providers or peers.

- **Guideline A:** The local preference of any customer-learned route must be greater than the local preference of any peer or provider learned route.[11]

Thus, any community that results in a routing policy which honors all of these properties can be considered stable. Particularly, these communities do not impact any of the stability proofs in the Gao-Rexford paper [23]. We consider these safe communities. Any community that does violate one of these assumptions can potentially be unsafe, as the stability proofs from the Gao-Rexford paper are not guaranteed to hold.

We analyze the BGP communities supported by several large networks as established by prior work [14] and identify safe communities. We assume these communities are only acted on if sent from a BGP customer as is the standard practice for networks supporting BGP communities [14, 47].

- Increase Local Pref Above Customer: Some networks offer a BGP community to increase the local preference of a route above that of a standard customer route [17]. This does not have an impact on BGP stability. The Gao-Rexford model puts no constraints on the preference of routes among customers.

- Lower Local Pref Below Customer but Above Peer: Many networks have a community that allows a customer to lower the local preference of a route below other customer routes but still prefer these routes over peer/provider learned routes. Once again these communities are safe as the Gao-Rexford model puts no constraints on the preference of routes among customers

- Lower Local Pref Below Peer: This community, also supported by many networks [14], allows a customer to **lower the local pref of its route below peer/provider routes**. This is a clear violation of the Gao-Rexford Guideline A (meaning the Gao-Rexford stability proof no longer holds) and is critical to the construction of our BGP Vortices.

- No Export Select and No Export All: These communities are used in our BGP Vortices as a means of pushing the Vortex in a cyclic motion to perpetuate the instability. However, these communities do no violate the Gao-Rexford assumptions. Specifically, the Gao-Rexford model produces sufficient assumptions for stability and generates proofs under the assumption that an AS *can* export a route to its appropriate peers [23]. ASes do not need to export the route to all available peers to achieve stability under the Gao-Rexford proofs.

Overall, most communities supported by major networks are safe, with **the exception of Lower Local Pref Below Peer**. In light of this, we advise networks cease to support this community in the interest of global routing stability.

7 Related Work

Related work in BGP security and stability has demonstrated both theoretical vulnerabilities in the protocol's design and practical attacks against its implementations. We organize related work into two categories: convergence analysis and security vulnerabilities.

BGP Convergence and Stability. Seminal work by Griffin et al. [27] and Varadhan et al. [49] showed that conflicting BGP routing policies can cause persistent route oscillations. Gao and Rexford [23] subsequently developed guidelines for routing policies based on real-world AS relationships that guarantee convergence. Gao [22] then introduced the valley-free property to enforce business relationship constraints.

Griffin et al. [26] formalized inter-domain routing stability as the stable paths problem, providing a theoretical framework for analyzing BGP convergence properties. Rexford et al. [43] and Labovitz et al. [31] showed that while routes to popular destinations tend to be stable over time, network changes can trigger convergence delays lasting tens of minutes. Our work shows that routing oscillations are not merely theoretical—the network conditions enabling them are widespread in today's Internet and can be deliberately exploited.

BGP Vulnerabilities. An extensive body of prior work has shown practical threats against routing stability and availability by exploiting vulnerabilities in the design and in implementations of BGP. Zhang et al. [50] showed how periodic bursts of data plane traffic can cause routers to drop BGP updates. This pulsing vulnerability lead Schuchard et al. [44] to develop the Coordinated Cross Plane Session Termination (CXPST) attack. Prehn et al. [40] demonstrated how rapidly announcing and withdrawing many prefixes can overwhelm routers with BGP updates. Further Router resource exhaustion has been a practical concern beyond attacks. The growth of Internet routing tables has stressed router memory limits, as documented in analyses of events such as 512K-day [10] and 768K-day [11].

Recent work has highlighted security risks stemming from BGP communities. Streibelt et al. [47] demonstrated that communities can be weaponized to trigger remote blackholing or

[11] We work under the more stringent Guideline A as opposed to the more lax Guideline B as we are aware of networks like AS 8283 which violate Assumption P. As of 10/2/24 AS 8283 has a peer session with AS 2914 (confirmed via https://sysadmin.coloclue.net/peering_overview.php) and has a transit session with AS 57866 (confirmed via https://sysadmin.coloclue.net/). AS 57866 purchases transit from AS 2914 (see https://radar.qrator.net/as/57866/connectivity/neighbors/providers) causing a cycle.

modify routing policies in distant ASes. Birge-Lee et al. [14] further showed how BGP communities increase the effectiveness of BGP route hijacking attacks, as they provide an adversary with fine-grained control over the propagation of fraudulent route advertisements. Our work shows how BGP communities can be exploited to create persistent route oscillations that cause temporary control plane inconsistencies by increasing control plane load across the Internet.

8 Discussion

Throughout the paper, we mainly discussed the impact of BGP Vortex on the current Internet, and BGP in particular. We now discuss our findings in the context of different inter-domain routing protocols, which have been developed and deployed to solve BGP's vulnerabilities and convergence issues.

BGPSec. BGPSec [32] is an extension to BGP which cryptographically authenticates route advertisements to prevent impersonation and route modification attacks such as BGP route hijacking. However, BGPSec adds significant complexity and computational overhead to the router's control plane, which must perform expensive public key cryptography to verify incoming route advertisements and sign outgoing ones. Since a BGP Vortex can be triggered without any impersonation using only standard-compliant BGP route advertisements, BGPSec provides no protection against BGP Vortex. In fact, the additional processing load incurred by BGPSec's cryptographic operations likely further amplifies the adverse effects of a BGP Vortex on route propagation delays and data plane availability.

Alternative Internet Architectures. A potential solution to the BGP Vortex attack are alternative inter-domain routing architectures that do not suffer from convergence attacks. Two prominent examples in this vein are consensus routing by John et al. [29], and the SCION next-generation Internet architecture [16]. In the case of consensus routing, all ASes participate in a global consensus operation to determine the new state of the global forwarding tables. Therefore, when consensus is reached, the global forwarding state is consistent. Under the assumption that consensus is difficult to disrupt, the convergence is assured. In the case of SCION, the route exploration system continuously creates new path segments, which can be registered by ASes in a path server infrastructure. Thus, connectivity is assured as long as the links corresponding to the path segments are available.

9 Conclusion

The Internet's routing infrastructure, with BGP at its core, represents a complex ecosystem that has evolved over decades through incremental additions and extensions. While this approach has maintained backward compatibility and enabled new functionality to meet growing demands, it has also created a layered system whose interactions are increasingly difficult to analyze and predict.

Our discovery of the BGP Vortex—arising from the unexpected interaction between *Selective NOPEER* and *Local Preference Lowering* BGP communities—proves that this unchecked complexity may create new threats for the Internet. What appears as a simple combination of two legitimate BGP extensions can trigger a cascade of routing updates and create persistent instability, which eventually lead to router overload and traffic black holes. Moreover, the BGP Vortex can also be seen as a form of amplification attack, generating floods of updates from minimal input, with serious security and availability implications.

We anticipate that this paper will help to create urgency to harden BGP and BGPSEC against such attacks, as well as to work on safer inter-domain routing systems.

10 Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00440780, Development of Automated SBOM and VEX Verification Technologies for Securing Software Supply Chains).

Ethics

The research conducted in this paper has followed strict ethical guidelines both in our experiments and disclosure.

We follow responsible disclosure practices and are coordinating with a National Cyber Security Centre (NCSC) to inform affected stakeholders and aid in the deployment of appropriate mitigation strategies for BGP Vortex. The NCSC is proficient in globally orchestrating ethical disclosures in the area of computer networking. A manuscript describing the BGP Vortex attack and mitigations will be shared globally by the NCSC with partner institutions, which further disseminate the information within their jurisdiction. Furthermore, we are actively engaging with networking communities such as RIPE and NANOG to ensure that operators at leading ISPs are made aware of the attack and how to mitigate it.

The majority of our experiments were conducted either through simulation or in isolated laboratory environments, ensuring no impact on external systems or the Internet. The only exception involves the BGP community probes detailed in Section 4.2: Ethically Probing Vulnerability of Tier-1ASes, which required interaction with the live routing infrastructure. To maintain the stability and integrity of the global routing system, we tested BGP communities individually—rather than concurrently—to verify that indeed they could be used in

the wild. We tested the NOPEER communities individually, confirming their behavior one-by-one. The local preference lowering communities were tested in combination, which was necessary to demonstrate that the final routing configuration is dependent on BGP update ordering. Still, the combination of the three local preference lowering communities is not sufficient to trigger persistent instability. This controlled order-dependent equilibrium, if combined with the separately verified NOPEER communities, proves that a BGP Vortex could exist in the wild.

In summary, our careful experimental procedures ensured that while we could validate our theoretical findings, we did not risk disrupting the broader Internet infrastructure. Our ethical disclosure through the NCSC ensures that the most affected stakeholders receive timely support in implementing appropriate defenses.

Open Science

We are committed to the principles of open science and reproducible research. To support these principles and enable verification and extension of our work, we have made all research artifacts publicly available via a stable Zenodo link. Our artifacts package includes:

- Simulation scripts and configurations used in our simulation experiments;
- documentation of our laboratory testbed setup and configuration;
- router configurations;
- analysis scripts and lab notebooks.

References

[1] Quagga documentation. https://www.nongnu.org/quagga/docs/quagga.html. Accessed 04. August 2024.

[2] Cisco ios ip routing: Bgp command reference - chapter: Bgp commands: neighbor timers through show bgp nsap summary. https://www.cisco.com/c/en/us/td/docs/ios/iproute_bgp/command/reference/irg_book/irg_bgp4.html, August 2010. Accessed 04. August 2024.

[3] Cisco ios ip routing: Bgp command reference. https://www.cisco.com/c/en/us/td/docs/ios/iproute_bgp/command/reference/irg_book/irg_bgp3.html, July 2011. Accessed 02. Aug. 2024.

[4] CVE-2014-9222. https://nvd.nist.gov/vuln/detail/CVE-2014-9222, December 2014. Accessed 06. January 2025.

[5] CVE-2018-0037. https://nvd.nist.gov/vuln/detail/CVE-2018-0037, July 2018. Accessed 06. January 2025.

[6] Cisco asr 9000 series route switch processor 440 data sheet. https://www.cisco.com/c/en/us/products/collateral/routers/asr-9000-series-aggregation-services-routers/datasheet-c78-674143.html, November 2019. Accessed 19. January 2025.

[7] CVE-2022-37035. https://nvd.nist.gov/vuln/detail/CVE-2022-37035, February 2022. Accessed 06. January 2025.

[8] Unicast routing protocols guide release 22.2.r1. https://infocenter.nokia.com/public/7750SR222R1A/index.jsp?topic=%2Fcom.nokia.Unicast_Guide%2Fmin_route_adver-d497e9445.html, February 2022. Accessed 02. Aug. 2024.

[9] damping (protocols bgp). https://www.juniper.net/documentation/us/en/software/junos/cli-reference/topics/ref/statement/damping-edit-protocols-bgp.html, June 2024. Accessed 04. August 2024.

[10] Emile Aben. 512k-mageddon. https://labs.ripe.net/author/emileaben/512k-mageddon/, August 2014. Accessed 17. January 2025.

[11] Emile Aben. 768k day. will it happen? did it happen? https://labs.ripe.net/author/emileaben/768k-day-will-it-happen-did-it-happen/, April 2019. Accessed 17. January 2025.

[12] Arelion. Bgp communities | arelion. https://www.arelion.com/our-network/bgp-routing/bgp-communities.

[13] Tony J. Bates and Enke Chen. An Application of the BGP Community Attribute in Multi-home Routing. RFC 1998, August 1996.

[14] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. Sico: Surgical interception attacks by manipulating bgp communities. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[15] Randy Bush. Ripe routing working group recommendations on route flap damping. https://www.ripe.net/publications/docs/ripe-580/, January 2013. Accessed 04. August 2024.

[16] Laurent Chuat, Markus Legner, David Basin, David Hausheer, Samuel Hitz, Peter Müller, and Adrian Perrig. *The Complete Guide to SCION. From Design Principles*

to Formal Verification. Springer International Publishing AG, 2022.

[17] Coloclue. AS8283 COLOCLUE-AS Whois. https://apps.db.ripe.net/db-web-ui/query?bflag=false&dflag=false&rflag=true&searchtext=AS8283&source=RIPE.

[18] Jerome Durand, Ivan Pepelnjak, and Gert Döring. BGP Operations and Security. RFC 7454, February 2015.

[19] Nicolas Fevrier. Introducing the acx7024x. https://community.juniper.net/blogs/nicolas-fevrier/2024/04/24/introducing-the-acx7024x, April 2024. Accessed on 19. January 2024.

[20] Ondrej Filip, Martin Mares, and Maria Matejka. Bird 2.15.1 user's guide. https://bird.network.cz/?get_doc&v=20&f=bird.html. Accessed 04. August 2024.

[21] FRR. Frrouting user guide. https://docs.frrouting.org/en/latest/, July 2024. Accessed 29. July 2024.

[22] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9(6):733–745, dec 2001.

[23] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. In *Proceedings of ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2000.

[24] Phillipa Gill, Michael Schapira, and Sharon Goldberg. A survey of interdomain routing policies. *Comput. Commun. Rev.*, 44:28–34, 2013.

[25] Vasileios Giotsas and Shi Zhou. Inferring AS relationships from BGP attributes. *CoRR*, abs/1106.2417, 2011.

[26] T.G. Griffin, F.B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, 2002.

[27] Timothy G. Griffin and Gordon Wilfong. An analysis of bgp convergence properties. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '99, page 277–288, New York, NY, USA, 1999. Association for Computing Machinery.

[28] Geoff Houston. Bgp updates in 2024. https://blog.apnic.net/2025/01/07/bgp-updates-in-2024/. Accesssed 10. January 2025.

[29] John P. John, Ethan Katz-Bassett, Arvind Krishnamurthy, Thomas Anderson, and Arun Venkataramani. Consensus routing: The Internet as a distributed system. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2008.

[30] Juniper. out-delay. https://www.juniper.net/documentation/us/en/software/junos/cli-reference/topics/ref/statement/out-delay-edit-protocols-bgp.html, June 2024. Accessed 29. July 2024.

[31] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM00. ACM, August 2000.

[32] Matt Lepinski and Kotikalapudi Sriram. BGPsec Protocol Specification. RFC 8205, September 2017.

[33] Tony Li, Ravi Chandra, and Paul S. Traina. BGP Communities Attribute. RFC 1997, August 1996.

[34] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route flap damping exacerbates internet routing convergence. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, page 221–233, New York, NY, USA, 2002. Association for Computing Machinery.

[35] Networkworld. Global Microsoft cloud-service outage traced to rapid BGP router updates. https://www.networkworld.com/article/971873/global-microsoft-cloud-service-outage-traced-to-rapid-bgp-router-updates.html, 2023. Accessed 20. January 2024.

[36] CBC News. Human error caused 2022 rogers outage, system 'deficiencies' made it worse: report. https://www.cbc.ca/news/politics/rogers-outage-human-error-system-deficiencies-1.7255641, 2024. Accessed 20. January 2024.

[37] NTT. Routing policies. https://www.gin.ntt.net/support-center/policies-procedures/routing/.

[38] Cristel Pelsser, Randy Bush, Keyur Patel, Prodosh Mohapatra, and Olaf Maennel. Making Route Flap Damping Usable. RFC 7196, May 2014.

[39] Cristel Pelsser, Olaf Manuel Maennel, Pradosh Kumar Mohapatra, Randy Bush, and Keyur Patel. Route flap damping made usable. In *Passive and Active Network Measurement Conference*, 2011.

[40] Lars Prehn, Paweł Foremski, and Oliver Gasser. Kirin: Hitting the internet with distributed bgp announcements. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '24, pages 19–34. ACM, July 2024.

[41] Matthew Prince. August 30th 2020: Analysis of centurylink/level(3) outage. https://blog.cloudflare.com/analysis-of-todays-centurylink-level-3-outage/, August 2020. Accessed 19. January 2024.

[42] Yakov Rekhter, Susan Hares, and Tony Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, January 2006.

[43] Jennifer Rexford, Jia Wang, Zhen Xiao, and Yin Zhang. Bgp routing stability of popular destinations. In *International Memory Workshop*, 2002.

[44] Max Schuchard, Abedelaziz Mohaisen, Denis Foo Kune, Nicholas Hopper, Yongdae Kim, and Eugene Y. Vasserman. Losing control of the internet: using the data plane to attack the control plane. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 726–728. ACM, October 2010.

[45] Philip Smith and Rob Evans. Ripe routing working group recommendations on route-flap damping. https://www.ripe.net/publications/docs/ripe-378/, 05 2006. Accessed 04. August 2024.

[46] One Step. as3356 one step. https://onestep.net/communities/as3356/.

[47] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. BGP communities: Even more worms in the routing can. In *Proceedings of the Internet Measurement Conference 2018, IMC 2018, Boston, MA, USA, October 31 - November 02, 2018*, pages 279–292. ACM, 2018.

[48] Tobias Fiebig Urban Suhadolnik. Asn assignment criteria revisited. https://www.ripe.net/community/policies/proposals/2025-01/, May 2025. Accessed 24. May 2025.

[49] Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32(1):1–16, 2000.

[50] Ying Zhang, Z. Morley Mao, and Jia Wang. Low-rate tcp-targeted dos attack disrupts internet routing. In *Network and Distributed System Security Symposium*, 2007.

A Triggering BGP Vortices Using Transitive Communities

In the appendix, we discuss how one can utilize the transitivity of BGP communities to induce a BGP Vortex in upstream ASes without requiring a direct connection. The transitivity of BGP communities ensures that an AS preserves any unknown community values when redistributing route advertisements. Prior work has shown that the transitivity of BGP communities enables various practical attacks [14, 47].

Consider adversary AS *A* connected to provider *P*, where *P* connects to three mutually peering providers, each supporting the selective NOPEER and local preference lowering communities as described in Section 3.4. Further assume that each of *P*'s providers uses a distinct BGP community numbering scheme to ensure each provider uses distinct community values. This is confirmed to be typical by prior work [14].

AS *A* constructs a BGP advertisement containing all communities necessary to trigger vortex behavior at *P*'s three providers. When *A* announces this prefix to *P*, the provider redistributes the advertisement to all upstream links while preserving the complete community set. Each upstream provider processes only communities matching its own policies, ignoring others.

This approach produces routing behavior equivalent to a direct vortex attack launched by *P* itself, effectively extending the adversary's reach beyond immediate customer-provider relationships.

The main consideration for this technique is that providers need to use distinct community numbering schemes that to not overlap, as this could potentially cause unintended community interpretation by the providers. However, standard BGP community configurations enforce a segmented namespace where the first 16 bits of standard communities (or 32 bits for large/extended communities) specify the defining ASN [47]. This ensures communities are processed only by their originating AS and prevents cross-provider interference and enables reliable transitive community exploitation.