Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann,
Cristel Pelsser, Georgios Smaragdakis, Randy Bush

BGP Communities: Even more Worms in the Routing Can

BGP Communities: Even more Worms in the Routing Can

Florian Streibelt
Max Planck Institute for Informatics

Franziska Lichtblau
Max Planck Institute for Informatics

Robert Beverly
Naval Postgraduate School

Anja Feldmann
Max Planck Institute for Informatics

Cristel Pelsser
University of Strasbourg

Georgios Smaragdakis
TU Berlin

Randy Bush
Internet Initiative Japan

ABSTRACT

BGP communities are a mechanism widely used by operators to manage policy, mitigate attacks, and engineer traffic; e.g., to drop unwanted traffic, filter announcements, adjust local preference, and prepend paths to influence peer selection.

Unfortunately, we show that BGP communities can be exploited by remote parties to influence routing in unintended ways. The BGP community-based vulnerabilities we expose are enabled by a combination of complex policies, error-prone configurations, a lack of cryptographic integrity and authenticity over communities, and the wide extent of community propagation. Due in part to their ill-defined semantics, BGP communities are often propagated far further than a single routing hop, even though their intended scope is typically limited to nearby ASes. Indeed, we find 14% of transit ASes forward received BGP communities onward. Given the rich inter-connectivity of transit ASes, this means that communities effectively propagate globally. As a consequence, remote adversaries can use BGP communities to trigger *remote blackholing*, *steer traffic*, and *manipulate routes* even without prefix hijacking. We highlight examples of these attacks via scenarios that we tested and measured both in the lab as well as in the wild. While we suggest what can be done to mitigate such ill effects, it is up to the Internet operations community whether to take up the suggestions.

KEYWORDS

BGP, Communities, Exploits.

1 INTRODUCTION

The Border Gateway Protocol (BGP) communicates reachability information between neighbors in the Internet. As the network evolved, the complexity of connections, policies, and economics drove the need for similarly complex and fine-grained routing policies [30, 44, 45]. As a result BGP, the de facto inter-domain routing protocol, has been extended to help support such policies, and provide value-added services. This work focuses on one such extension, *BGP communities* [25], and the implications of its real-world implementation and deployment.

BGP communities are an optional transitive BGP attribute used to "tag" advertisements. Operators frequently configure their infrastructure to take different actions depending on community tags. So, communities provide not only a common label for groups of prefixes, but also the ability to *signal semantics* between ASes and between routers within an AS.

BGP communities are increasingly popular and are used to encode an ever-wider variety of information [28, 29, 34, 36]. Within the last year the number of observable communities increased by roughly 20%, see Section 4. As we describe in Section 2, communities are used to realize routing policies, bias path or peer selection, steer traffic, etc. ASes also use communities to offer value-added services for customers of ISPs and members of IXPs including tagging of route ingress points and origins [4, 34, 37], selective advertisement [26, 35, 38, 52], traffic engineering [29, 48, 55], and Remotely Triggered Blackholing (RTBH), i.e., dropping of traffic to a target destination to mitigate Denial-of-Service-Attacks (DoS) [27, 28, 36, 47]. Some providers even use communities to encode latency information [6, 7].

While BGP communities are a seemingly innocuous feature, we show that they can be used to influence routing in unintended ways. Although the community-based attacks we consider require certain conditions for success, we show that these conditions hold sufficiently widely to warrant operational attention. Importantly, since our extensive measurements show that communities are widely propagated, see Section 4, an attacker exploiting the BGP communities of a particular AS need not be a directly connected peer. Further, we demonstrate the feasibility of attacks both with and without address space hijacking, suggesting that existing hijack detection methods are insufficient to detect community-based attacks.

The attacks are the result of weaknesses in the current use and implementation of BGP communities and community-based services. Services enabled by communities are typically relevant only between directly connected ASes – for instance, an AS tagging a backup route with a community to indicate that the remote AS should use a lower local preference. Intuitively then, one might expect communities to not propagate through multiple ASes, or beyond their intended destination AS. However, via large-scale analysis of passive BGP datasets, we find that more than 50% of the BGP communities traverse more than four ASes and we see 10% with a hop count of more than six, see Section 4.

To better assess the potential vulnerabilities enabled by BGP communities, we design multiple scenarios that highlight intentional, unintentional, and malicious community use in Section 5. These include the ability to remotely signal blackholing of a prefix for which the attacking AS is not responsible, traffic steering, and route manipulation of another AS's prefixes.

To demonstrate the vulnerabilities of community handling in practice, we conducted two classes of experiments while ensuring prior coordination and permission of all involved ASes and networks, see Section 7.1. First, we tested in the lab the conditions

that must hold to realize the attacks; finding default and recommended configurations which enable the attacks we considered. Using insights from the lab, we conducted experiments in the wild to demonstrate the feasibility of these attacks in the real Internet. Our evaluation in the wild shows that, unfortunately, some of the BGP community-based attacks are easy to achieve. In some attack settings, e.g., when the attacker is on the AS path, even without hijacking and even if BGP route validation is used.

In summary, the community attacks we demonstrate are the result of weaknesses in the current use and implementation of BGP communities and community-based services including: community propagation behavior, complex policies, error-prone configurations, and a lack of cryptographic integrity and authenticity for communities. The main contributions of this paper are:

(1) We analyze BGP community propagation (Section 4) showing that 2.2K networks forward received BGP communities onward. We show that the majority of communities are propagated through the entire Internet.
(2) We highlight this routing system can of worms and identify sufficient conditions for community-based attacks on the routing system (in Section 3).
(3) We sketch three scenarios of how BGP communities can be misused (Section 5).
(4) We show that these attacks are possible in *lab experiments* and in the *wild* (Sections 6 and 7). We highlight traffic dropping due to remotely triggered blackholing, as well as remote steering of traffic and route manipulation, possibly through an interceptor, i.e., a rogue traffic monitor.
(5) We provide recommendations on the use of communities in Section 8.

2 BGP COMMUNITIES: A PRIMER

Communities are an optional BGP attribute used as a signaling mechanism within and between ASes [25]. While the 32-bit community field[1] can take any value, by convention the first 16 bits represent the AS Number (ASN) of the entity defining the community, while the last 16 bits indicate an action or label. The human-readable community presentation format separates numeric representations of the ASN and label with a colon, e.g., `3130:411`.

There is only a small set of standardized well-known community labels, e.g., NO_EXPORT (`65535:65281`) indicates a route should not leave a BGP confederation, NO_PEER (`65535:65284`) [46] indicates a route should not be propagated via a bilateral peering link, and `65535:666` the standardized blackhole community [47]. These well-known communities cover a very small subset of all communities in use (Section 4) and the complex routing policies network operators realize via BGP communities. Indeed, an AS is free to define (or leave undefined) the semantics of the 2^{16} possible values for its communities. For example, in the previous example, AS 3130 "owns" communities `3130:XXXX` and may define them arbitrarily. It is important to note, that there are no explicit mechanisms to enforce this segmentation of the community space, and any AS is free to add, delete, or modify the communities of

[1]With the advent of 32-bit ASNs, RFC8092 [42] introduces "large" 96-bit communities. In this paper, we focus on traditional 32-bit communities as they already offer many intriguing scenarios. We leave an extended investigation of large, extended, and private communities to future work.

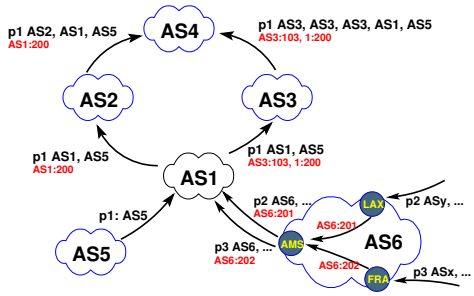


Figure 1: Policies with BGP communities: AS1 requests path prepending by tagging `AS3:103` towards AS3, and informs its peers that prefix p1 is a customer prefix, by attaching community `AS1:200`. At the same time AS6 uses communities `AS6:201` and `AS6:202` to signal where a route is learned.

BGP advertisements that transit its control plane with impunity. Indeed, even cryptographic proposals to protect the authenticity and integrity of routing announcements do not cover BGP communities [24, 40, 41, 43, 50].

Communities can be added, deleted, or set by an AS on prefix origination, ingress, or egress. Bonaventure et al. were the first to propose a taxonomy of community values [23, 29] and identified two main modes of operation. First, there are AS-internal communities that are set when receiving a route. Second, communities labeled on egress are commonly used to signal or pass information down the path. Such outbound communities carry a broad spectrum of meanings, but most fall into the following categories according to [29]: (a) route selection: adjustment of local_pref and AS path prepending, (b) selective announcement: routes are labeled according to which class of ASes (peer, transit) or even specific ASes they should be announced to, (c) route suppression: same as (b), but states explicitly to whom not to announce a route, (d) blackholing: traffic towards this prefix, mostly /32s (in IPv4) should be dropped, and (e) location: to signal where a route has been learned.

Figure 1 illustrates some ways communities are commonly used in practice. Here, AS6 tags incoming routes with the geographic location where the prefix was received, in this case from Los Angeles (LAX) and Frankfurt (FRA). The first part of the community denotes AS6, while the values `201` and `202` are chosen by AS6 to indicate the location. Further, AS3 defines the received community `AS3:103` to prepend its AS three times to path. AS1 can then perform route selection by attaching the community `AS3:103` to the announcement of $p1$ to AS3. Once AS4 receives both announcements for $p1$, it will prefer the shorter path via AS2.

The level of community support as well as documentation varies considerably among providers. Some networks, especially large ISPs [10, 11] and IXPs [1, 3, 4, 19] implement fine-grained semantics using as many as hundreds of communities. Unfortunately, there is no central database of record for providers' communities and associated actions, but rather scattered and incomplete documentation. In reality, this boils down to networks documenting the communities relevant to their peers and customers on their website and/or in Regional Internet Registries (RIR)/Internet Routing

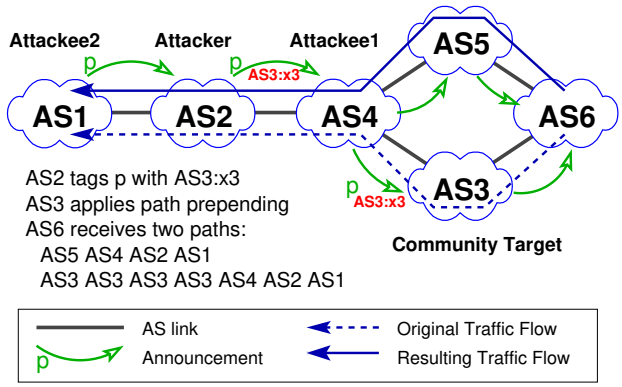


Figure 2: BGP communities scenario: AS path prepending.

Registries (IRR) [8]. We, therefore, lack a definitive understanding of the global definitions and use of communities.

Further complicating the use of communities is that there is no strict policy as to how a network should handle incoming routes tagged with communities. Therefore, there is no consistent behavior in forwarding BGP communities amongst different networks; e.g., some will remove all communities not understood by them, while others will forward everything, and yet others have more complex community propagation policies. We discuss implications of this design choice in Section 3, and measure the extent of community propagation in Section 4.

3 BGP COMMUNITIES: CAN OF WORMS

By allowing ASes to extend the semantics of routing updates, BGP communities can significantly simplify policy implementation. As such they are, as we underline in Section 4, heavily used in today's routing system. However, as we now show, they also present a can of worms in the sense of "a situation that causes a lot of problems for you when you start to deal with it" [2]. We, then, discuss why this is too often the case in today's Internet.

3.1 Motivating Example Scenario

We use a common community service, AS path prepending, to show the intended use of communities as well as noting the potential for abuse, see Figure 2. AS1 announces the prefix p to AS2 and on to AS4, which announces it to AS3 and AS5 and then on to AS6 (see green hollow arrows). Consider traffic from AS6 to p. As the AS paths via AS3 and AS5 have the same length, AS6 may choose to route via AS3 (dotted blue line). AS3 offers AS path prepending via the community `AS3:×n` to prepend n times; where n is typically between 1 and 3. For example, NTT uses `2914:421` for prepending once, `2914:422` for prepending twice, etc. The *intended use* of this service is to enable AS3's peers, e.g., AS4, to do traffic steering.

However, if some AS on the path, e.g., in this case, AS4, does not filter communities, this service can also be (ab)used by other ASes on the announcement path. Potential abuses include: AS2 or AS1 setting the community `AS3:×3` on the announcement of prefix p; causing AS3 to path prepend three times for the announcement of p to AS6. This changes the traffic flow from AS6 toward AS1 to choose the AS5 (shown via the solid blue line) as opposed to AS3. The motivation for AS2 might be:

Malicious interceptor: If AS5 is a malicious interceptor [22, 57], AS2 is able to steer traffic through it.

Impose additional cost: The link from AS5 to AS4 might be more expensive than the link between AS3 and AS4. AS2 forces AS4's ingress traffic to the "expensive link", that yields high cost for AS4.

Performance improvement: If the service offered by AS6 is popular and the performance via AS5 in terms of bandwidth and/or delay is significantly better, AS2 may improve its service to AS1 by tagging the announcement p with the path prepending community of the provider of its provider, i.e., to steer traffic via AS5 rather than AS3.

Performance impairment: If the performance via AS5 is significantly worse than the performance via AS3, AS2 may slow down an application originating in AS6 that is clogging its network.

Because BGP communities are transitive attributes, the above is fully compliant with the specification. But the actual behavior/use depends on the policies of the involved ASes, in particular, AS3 and AS4.

The above is a teaser example to highlight some potentially unintended consequences of *transitive* BGP community use. In Section 5 we show multiple scenarios for traffic steering as well as remotely triggered blackholing (dropping of traffic). When combined with prefix hijacking [39] this raises significant security concerns. Thus, we argue that transitive BGP communities are "a can of worms" for the routing system.

3.2 BGP Communities Shortcomings

We believe that BGP communities may be an insufficiently constrained feature for the Internet routing system for the following reasons.

Missing Semantics: Communities are "just tags." This has multiple consequences: (a) Communities do not have a generally agreed upon semantic. Only a few communities and the "expected" community format are standardized via RFCs (Section 2). This is analogous to having a program's semantics in the comment statements. (b) Communities are AS specific. Each AS can define their own communities and determine how to publish them, e.g., publicly or only to their peers/customers. (c) The order in which communities are processed by a router is not well-specified and differs by operator configuration as well as by equipment vendor.

No authentication of tagger/community: Any AS on the path can add or modify any of the communities of a routing update. The recipient of a community cannot determine which AS on the path added or modified any of the communities.

Yet, communities are critical for operation since complex routing policies are a reality and unlikely to change. Currently, BGP communities are the most convenient way for signaling information between ASes – an essential component for realizing routing policies. Moreover, an AS may not only mistakenly or maliciously

[2] Definition of "can of worms" according to Cambridge Advanced Learner's Dictionary & Thesaurus.

Source	BGP messages (in Billions)	IPv4 prefixes	IPv6 prefixes	Collectors	IP peers	AS peers	Communities	ASes	Origin	Transit	Stub
RIS	4.80	823,619	76,783	13	275	268	53,208	62,210	61,806	15,016	47,194
RV	9.12	874,054	65,812	15	357	206	57,344	62,424	62,020	9,418	50,991
IS	23.48	830,527	63,584	4	154	97	50,128	62,153	61,754	11,067	51,086
PCH	1.57	802,637	64,136	162	4,640	1,924	40,719	62,033	61,620	10,914	51,119
Total	38.98	967,499	84,953	194	5,158	2,133	63,797	62,681	62,253	15,578	47,103

Table 1: Overview of BGP dataset (April 2018). IPv4 prefixes contributed 92% to the total number of prefixes while IPv6 contributes 8%. Therefore, we focus on IPv4 for all other statistics.

tag a route with a community, it may even free-ride, i.e., hijack a prefix or subprefix[3] by announcing them tagged with a community of their choice.

Given the above, one has to ask what this implies for the Internet routing system. First, each AS should define its policy in regard to remote community use and/or install appropriate filters and community parsers. Second, policy implementation should account for ill-specified and misused communities. Misuse of communities can either happen due to malicious intent or by mistake, e.g., due to fat or thin fingers. Indeed, when considering the above shortcomings, together with the scenarios highlighted in Section 5, we urge the community to rethink whether communities are the right mechanism and, if so, how to ameliorate the above shortcomings.

3.3 Terminology

In the rest of the paper we use the following terminology:

Attackee: The attackee in our context is the AS whose prefix/traffic is affected by manipulating the community attribute of an update.

Attacker: The attacker is the AS which is manipulating the community attribute of an update or announcing a hijacked (sub-) prefix.

Community target: The AS whose community service is used to change the route or traffic flow. We sometimes also refer to this AS as the community provider.

Thus, in Figure 2 AS3 is the community target, AS2 is the attacker and depending on AS2's motivation the attackee is AS4 or AS1.

4 BGP COMMUNITIES PROPAGATION

According to RFC1997 [25] BGP communities are an optional transitive attribute. Yet, their expected use is often between two AS neighbors. In this section, we tackle this apparent contradiction. First, we measure how common BGP communities use is. Then, we show how often communities are propagated beyond a single hop, i.e., are transitive, or if they even on the AS path. Finally, we check for indications that ASes actively strip communities.

4.1 Datasets

We rely on a multitude of vantage points within the Internet routing system.

[3]Hijacking a route corresponds to announcing a prefix for which the AS is not responsible for.

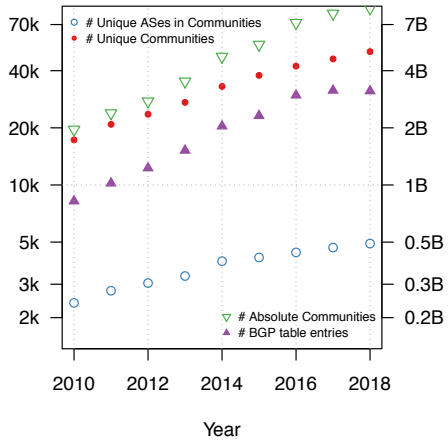
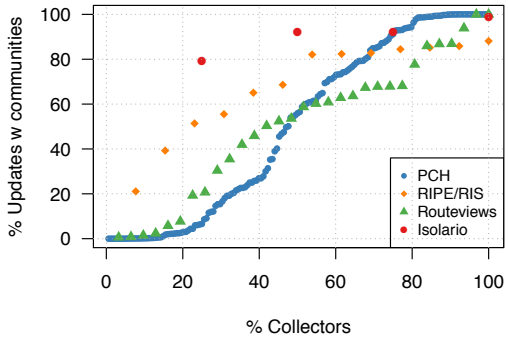


Figure 3: BGP communities use over time.

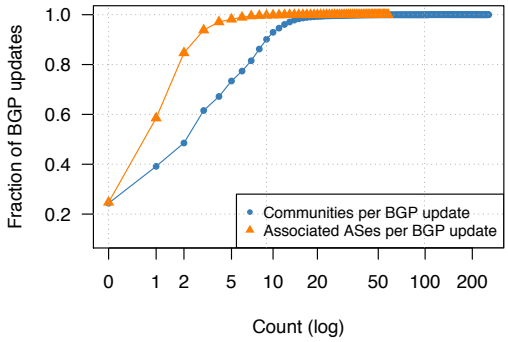
BGP routing tables and updates: We rely on the widely-used public datasets of the route collectors from (i) RIPE NCC Routing Information Service (RIS) [17], (ii) University of Oregon Route Views (RV) [18], (iii) Isolario project (IS) [9], and (iv) Packet Clearing House (PCH) [13]. Each of these platforms consists of multiple routers which collect BGP updates from many BGP peers. Some BGP peers send full routing tables, others partial views, and even others only their customer routes. We use the data for the month of April 2018. We remove AS path prepending to not bias the AS path. For an overview see Table 1. One specialty of the PCH platform is that it maintains route collectors that peer with the route servers at about 180 different IXPs around the Globe (ca. April 2018) [12]. Route servers are typically a value-added service of the IXP that collect routing information in a centralized manner and redistribute it to connected member routers. Thus, PCH offers BGP routing information for most of the IXP members [52].

Looking Glasses: We use looking glasses of certain ASes, when available, to confirm (i) community availability and propagation, (ii) route changes, as well as (iii) reachability of prefixes.

Active Measurements: We use the RIPE Atlas platform [16] to ping and traceroute to multiple targets during and after routing experiments. RIPE Atlas is an open distributed Internet measurement platform with roughly 10K active measurement nodes. When studying traffic shifting and/or dropping attacks, we use



(a) ECDF: Updates with BGP communities by collector per platform.



(b) ECDF: Communities per BGP update and per AS.

Figure 4: BGP communities use as observed in the collectors of our study.

traceroutes along the expected and the altered path to ensure the effect of the routing attack on the data plane.

4.2 BGP Communities Use: A first look

As a first step, we measure how wide-spread community use is. Overall, our results validate previous observations [28, 29, 34, 36] that it has increased significantly over the last five years, see Figure 3. Indeed, today more than 5K ASes offer community-based services[4] and we observe more than 63K different communities in our dataset from April 2018. This is an increase of 18% over 2017.

Overall, we find that more than 75% of all BGP announcements at the more than 190 BGP collectors have at least one community set. This means that we can indeed use these collectors to study community use and propagation. Interestingly, some collectors observe more communities than others. Figure 4(a) shows for each BGP collector the fraction of their updates which have at least one community set (in increasing order for each of the four platforms). A large number of our observation points allow us to study community propagation.

We also measure the number of distinct ASes for which we see communities at each BGP collector, see Table 2. We see more than 60K unique communities from more than 5.6K ASes which are not

[4]This statistic is computed under the assumption that communities follow the format convention, namely, `AS:value`.

Source	Total # of ASes	w/o collector peer	on-path	off-path	off-path w/o private
RIS	4,931	4,925	3,647	1,826	1,480
RV	5,383	5,375	3,510	1,668	1,279
IS	4,728	4,723	3,513	1,757	1,420
PCH	4,170	4,118	3,002	1,585	1,259
Total	5,659	5,630	3,958	2,154	1,721

Table 2: Summary of ASes with observed BGP communities.

directly peering with the respective BGP collector. This suggests that communities are propagated beyond direct BGP neighbors; or one would only see communities associated with direct BGP-peers of the collector.

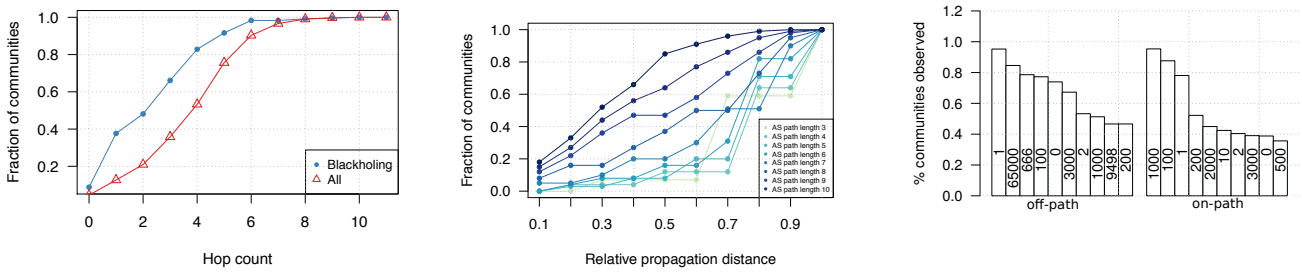
Next, we measure the number of communities per BGP announcement, see Figure 4(b). Recall, 75% carry at least one BGP community. Moreover, 51% have more than two communities set and 0.06% have more than 50 communities set (blue dots). These communities are often (41%) associated with more than a single AS (orange triangle). This is yet another signal that communities are indeed transitive.

4.3 BGP Communities Propagation Properties

Next, we measure how far communities propagate. We rely on the format convention, i.e.: `AS:value`. Consider a BGP update for prefix p originated at AS1 and observed at AS5 with AS path `AS5 AS4 AS3 AS2 AS1`. Assume that the update is tagged with `AS1:X` and `AS3:Y`. We assume that AS1 tagged the route with community `AS1:X` since it is the origin AS. The second community is tagged with AS3 and can be either a community received by AS3 from AS2 on ingress or set by AS3 on egress towards AS4. To estimate how far communities propagate we conservatively assume that the route is tagged with the community `AS3:Y` by AS3 rather than by AS2.

However, there is a significant number (21K) of communities of the form `ASX:Y` where ASX is not on the AS path. We call these communities "off-path" and the others "on-path". The former can occur, e.g., at an IXP where the IXP's AS provides the service signaled by the community but, by convention, IXPs are not on the AS path. Other reasons involve widespread tagging (community bundling) to simplify configuration, see, e.g., as reported by Giotsas et al. [36]. Overall, see Table 2, we find that 4K ASes are encoded in the on-path communities and 2K in off-path communities. Among the off-path communities there are roughly 400 private ASes [49]. Private ASes are per se off-path as they are not routed. They are often used by networks with large AS numbers which do not fit into the 32-bit community format. Thus, we focus on the communities with public AS numbers.

For on-path communities Figure 5(a) shows an ECDF of the number of AS hops that each community is relayed along the AS path. The red triangles represent the all BGP communities we observed. We find that a significant number of communities are propagated multiple hops. Almost 50% of the communities travel more than four hops (the mean hop length of all announcements [51]). The



(a) Propagation distance ECDF: path communities.

(b) Propagation distance ECDF: by AS path length.

(c) Top-10 values for off- and on-path communities.

Figure 5: BGP communities propagation properties.

maximum hop distance we observed is 11 which, given the highly connected AS graph, is rather large.

To check if specific classes of communities are more likely to be propagated we consider blackholing communities as a case study. Hereby, we identify blackholing communities either by the value 666 as defined in RFC7999 [47] or based on the list of verified and inferred blackholing communities from previous work [36]. The resulting ECDF is shown by the purple squares in Figure 5(a). The difference between the two ECDFs clearly shows that blackholing communities do not travel (on average) as far as other BGP communities. Around 50% of the blackholing communities travel only up to two AS hops, about 80% travel up to four. This is a clear indication that blackholing communities are treated differently by network operators. On the other hand, we still observe some blackholing communities with large hop counts – up to 11.

To check to which extent the above observations are biased by the AS-path length, Figure 5(b) shows the ECDF of the number AS edges that each community is relayed on for different AS path lengths. Hereby, we do not consider communities of the monitor AS but do include the edge to the monitor. The color gradient corresponds to the respective AS path length—light green for path length of three up to dark blue with a path length of 10 ASes. This plot highlights that a significant number of the communities travel more than 50% of the AS-path distance. However, as the path length increases the fraction of the communities that travel longer distances decreases somewhat. The reason for is that each AS on the path can add communities. Therefore, the expected number of communities that can only travel some portion of the AS path is higher. Thus, the plot highlights that communities are propagated significant distances in the Internet independent of the AS path length.

Using the same data we measure how many ASes propagate communities, i.e., are transitive for at least one BGP community of another AS. We do not include the ASes that directly peer with the collector[5]. Thus, for AS2 to be considered transitive we require at least one BGP update for a prefix p tagged with a community `AS1:X` on a path `AS3 AS2 AS1`. We find that there are 2.2K transit ASes[6] that relay communities relative to a total of 15.5K transit ASes in our dataset.

Next, we explore popular values involved in the observed communities and how these differ for off- vs. on-path communities. Overall, we find that the tails are extremely long—a consequence of the non-standardization of communities. Figure 5(c) shows a histogram of the top-10 most popular values for both off- and on-path communities. Each bar is annotated with the corresponding community values. Note, that their individual contribution is rather small and that they differ significantly. Among the most popular off-path communities is 666 which is used for blackholing. For on-path 666 is not among the top-10 community values. Rather, it is far down in the tail. One explanation is that it is often not observable for on-path since the respective AS should have acted upon receiving the blackhole community. For off-path we see more announcements with blackholing as they are often applied on all peering sessions rather than only selectively [36]. The other values look like convenient values, e.g., for local pref with 100, 200, and 1000.

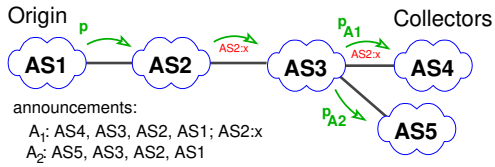
4.4 BGP Communities Filters

So far we focused on how common communities are and if they are forwarded. We have yet to measure if ASes only selectively forward communities or if they actively filter them. As there is no best practice on how to handle communities, networks may filter out all, none, or just specific ones. Measuring this is not straightforward as the only indication of filtering (resp. selective forwarding) is the lack of community propagation as seen in the BGP data. Further compounding the measurement difficulty are that (a) any AS on the path may remove a community, and (b) an AS may receive a "better route" (in the sense of BGP best path selection) not tagged with the community.

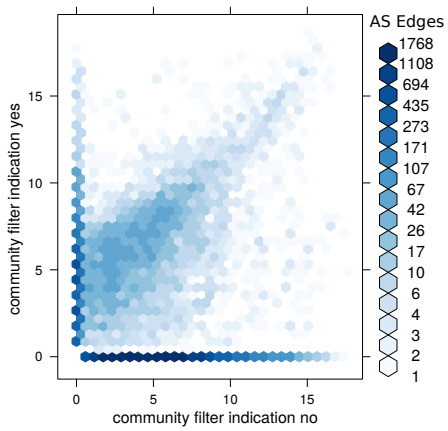
We nevertheless try to identify BGP neighbors where communities are not propagated by collecting indication counts for each directed AS pair. We iterate through all prefixes and, for each prefix p, we consider all updates at the same time and look for ASes where a community that has already been forwarded is propagated to one peer but not to another. The latter is an indication of filtering or

[5]The configuration for these peerings is often collector specific and may differ from the "regular" policy of the AS.

[6]We consider an AS a transit AS if there is at least one AS path in which it is neither the origin nor the collector.



(a) AS edges community change indication counts. After annotating all AS-edges with the observed communities for prefix p from all announcements, we find edges with and without communities, e.g., at segments of the AS-paths. This is repeated for all prefixes.



(b) AS neighbor indication counts: Forwarding vs. filtering.

Figure 6: Community forwarding behavior.

selective forwarding. The former is an indication of forwarding, i.e., *no* filtering.

To make this more concrete consider the example shown in Figure 6(a). We find two announcements A_1, A_2 for prefix p originating in AS1 in the bgpdumps of BGP collectors in AS4 and AS5. Announcement A_1 contains AS-Path `AS1, AS2, AS3, AS4` and carries a community `AS2:X`, while A_2 has AS-Path `AS1, AS2, AS3, AS5` and carries no communities at all. For this analysis we assume the community was not added earlier than AS2. Thus we increase the *community-added* indication on the edge (AS2, AS3).

Here, A_1 serves as an indication that AS3 transitively forwards the community from AS2 onwards. Therefore, we increase the *community-forwarded* indication count for the AS pair (AS3, AS4). A_2 allows us to increase the *community-filtered* indication count of AS pair (AS3, AS5). We know from A_1 that for this prefix the community `AS2:X` is forwarded to AS3 and that AS3 forwards it to some other peers; but we do not see it on the edge (AS3, AS5).

We find signs of transitive forwarding of communities for 4% of the almost 400,000 AS edges and for filtering for roughly 10%. These numbers increase to 6% resp. 15% if we consider AS edges with at least 100 AS paths. We acknowledge that the results of the above heuristic are biased by the BGP collectors which give us different degrees of visibility of the AS edges, as well as by the number of paths observed within the observation period. However, since we consider a full month of BGP updates from four different collector platforms we have reasonable coverage of the AS graph.

Figure 6(b) shows a scatter plot on log-log axis (to the base of 10)[7] of the filtering vs. non-filtering indicators per AS edge. We only include AS edges with at least 100 BGP paths and where we can find either an indication for or against filtering. The count values per AS edge range from 0 to 98 million (thus, the values on the x- and y-axis) which comes from the number of different communities and paths that are used in the filter indication computation. The color of the hex-bins correspond to the number of AS edges (darker color indicates more AS edges).

For some AS edges, we find indications that they strip all communities. Those are the ones on the bottom. For others, on the left hand side, we see no indication of filtering, i.e., they forward all communities without touching them. Naturally, we have also many AS edges in the middle of the plot, where we have mixed indications: some communities are forwarded and some are filtered.

The explanation for this mixed picture lies in the absence of best practices regarding BGP communities. After inquiring within the operator community, we found that nearly everyone has a different view on this—some remove all communities, some do not tamper with them at all, while others act upon and remove communities directed at them and leave the rest in place. On the other hand, there are operational reasons to only forward some communities to some BGP neighbors, e.g., different handling of customers and peers.

One natural question in this context is if the relationship type of an AS edge has any influence on filtering. To check this we use the CAIDA AS relationship dataset [2] to distinguish between customer-provider, provider-customer, and peering edges. However, we find that this classification is too coarse grained to allow for a conclusive picture regarding handling of communities. Thus, we plan in future work to correlate filtering/non-filtering of communities with the role of an AS in the Internet topology.

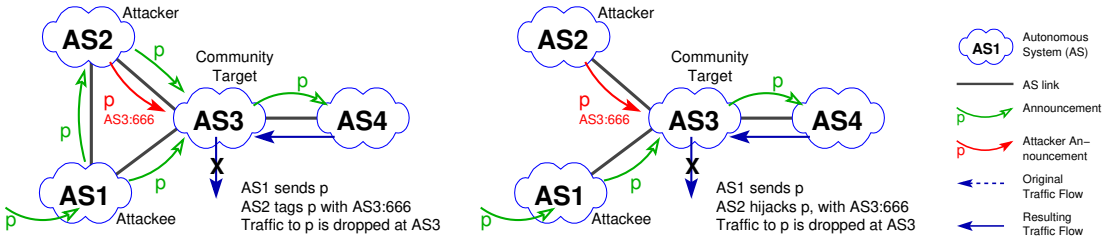
5 UNHAPPY SCENARIOS

In this section, we highlight different scenarios where transitive community propagation can enable unintended results, including remotely triggered blackholing, traffic steering, and route manipulation.

5.1 Remotely Triggered Blackholing

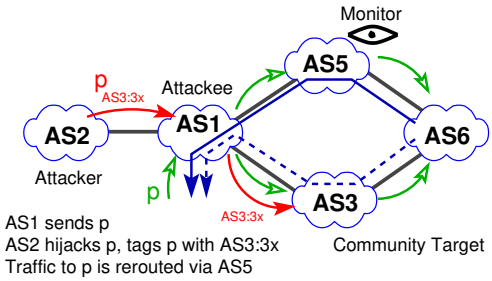
High-volume DDoS network attacks can heavily degrade network performance even to the point of making services unavailable [21, 32]. Especially edge networks often suffer as they cannot handle such high traffic levels. One mitigation option is *blackholing*, i.e., *dropping* all traffic going to a destination under attack, ideally, as close to the source as possible. As result, the victim IP address or the entire prefix becomes intentionally *unreachable*. Many networks provide their customers with the ability to automatically blackhole traffic using BGP communities as a signaling mechanism, so-called "remotely triggered blackholing" (RTBH) [47]. Networks issue blackholing requests by sending BGP announcements to their direct BGP neighbors for specific destination prefixes with the blackholing community of the respective network. The neighbor, upon receiving such an announcement discards, at its ingress, traffic whose

[7] The plot uses a logarithmic x- and y-axis. To include zero values we plot the logarithms of the (values + 1).

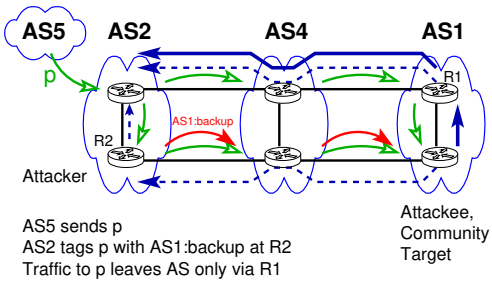


(a) Without prefix hijacking. (b) With prefix hijacking. (c) Legend.

Figure 7: Remotely triggered blackholing.



(a) AS path prepending/with hijacking



(b) Local pref/Without hijacking

Figure 8: Traffic steering.

destination address is in the blackholed prefix. In principle, this service should only be used in case of attack and by networks which actually have authority for the blackholed prefix or IP address.

However, consider the example shown in Figure 7(a). Here, *AS*1 announces prefix *p* to both *AS*2 and *AS*3. *AS*3 offers blackholing service and is the community target in this scenario. If *AS*2, the attacker, adds the blackhole tagged for *AS*3 to its announcement for *p* to *AS*3, traffic to *p* may be blackholed at *AS*3 even though the AS path of the tagged route is longer. The reason is often preferred treatment of the blackhole community before best path selection, see, e.g., the suggested configuration in [27]. Alternatively, *AS*2, the attacker, may announce a more specific of *p* which again has higher priority than the direct announcement from *AS*1, the attackee. Note, if *AS*4 also offers blackholing services via communities the same attack can be launched with *AS*4 as community target as long as *AS*3 propagates communities.

The above example requires the attacker, *AS*2, to be on a path from *AS*1 to *AS*3. However, even if this is not the case *AS*2 may be able to hijack prefix *p*, especially if *AS*2 and *AS*3 are peering, since strict prefix validation is often not in place, see Figure 7(b). Indeed, [53] reports 5,295 routing attacks (route leaks and hijacks) alone in 2017 which arguably should not be so frequent if proper filtering would be in place.

Even when prefix validation is in place, it may be possible to hijack prefixes, by tagging them with a blackhole community, depending on the order in which announcements are processed by a router's filters. For example, there are configurations, e.g., [56], where instead of discarding the announcement (due to hijacking) the router might process the hijacked announcement if tagged with the blackhole community as the community raises the routes precedence.

If AS2 has the ability to hijack prefix p of the attackee (AS1), it can announce p with a short AS path tagged with the blackhole community of AS3. This causes AS3, the community target, to drop all traffic to p. Again, a similar scenario is possible with AS4 as community target if AS3 propagates AS4s blackholing communities. Note, such an attack may be more or less interesting than simply hijacking. First, it may be effective only because of the community tag (validation done after blackholing). Second, whereas hijacking may only partially disrupt traffic (to the poisoned ASes), the hijacking plus blackholing attack disrupts all traffic to the victim.

5.2 Traffic Steering

Traffic engineering is one of the essential tasks of a network operator. The generally preferred choice for an AS is selective announcement of prefixes. Sometimes, this is not desired or not sufficient. A common alternative is for remote ASes to provide AS path prepending, Local Preference tuning, Multiple Exit Discriminator (MED) tuning, or partial route announcements, e.g., in specific regions such as Europe only, US only, Asia only. Many ASes accept signals for these tunings via BGP communities; and many ASes are offering these traffic steering services to their customers.

Recall the example from Figure 2 in Section 3. It highlights that it is possible to intentionally or unintentionally steer traffic over a link that should not be used according to the AS's policy. Indeed, if the involved ASes are susceptible to prefix hijacking this can be further misused as shown in Figure 8. An attacker AS may hijack prefix *p* (which AS1 receives from a peer) and tag it with the

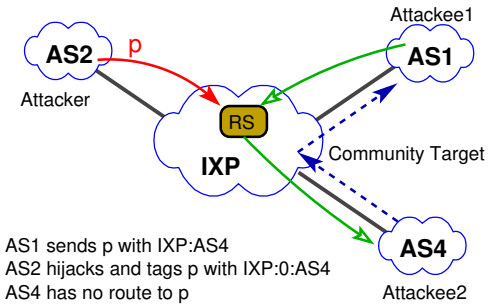


Figure 9: Route manipulation at an IXP with hijacking.

prepending community, thus, rerouting traffic via AS5[8]. This can cause trouble for AS1, either due to the unintentional heavy use of the link between AS1 and AS5, e.g., a paid peering link, or if AS2 and AS5 collaborate and AS5 has a malicious traffic tap to inspect all traffic to p.

The next example, see Figure 8(b), shows how AS2, the attacker, can use the local pref communities of AS1 to force AS1 to route all its traffic to AS2 over a single link via AS4 to AS1. While this may look undesirable at first, this can be highly beneficial for AS2, e.g., if R2 is in Hong Kong and the origin of p is in the US. AS2 in effect forces AS1 to pay for expensive intercontinental transport. In this case, the local pref community can be used to declare the undesired path (from the view of AS2) a backup path. We leave the decision on whether this is an attack or a smart way of reducing cost to the informed reader.

5.3 Route Manipulation

Using communities it is not only possible to cause blackholing or change traffic paths; but it is also possible to manipulate routes. In particular, Figure 9 shows how this can be done at an IXP. IXPs often offer community services via their route servers as value added service to customers. One popular service is the ability to tag routes with communities to signal to which peer a route should be advertised, e.g., the community `IXP:AS4` is used to selectively advertise a route to IXP member AS4. Thus, if AS1 sends the route with this prefix it can expect to receive traffic for p via the IXP for AS4. Now, if an attacker, AS2, uses another community service of the IXP, namely a community to signal that a prefix should not be advertised to a peer, e.g., using the community `0:AS4`, there is a conflict. This conflict is resolved at the route server by applying the rules for community-based services in a specific order. For some IXPs this order as well as their route server configuration is publicly available. We checked that at least for one IXP communities that are used to "not advertise a prefix to a peer AS" are handled before those that are used to "advertise to peer AS". This causes the attacker of Figure 9 to succeed in not advertising a route for prefix p to AS4.

[8]Even though AS2 announces a route for p it may not receive much if any traffic if the best route on most routers remains the one to the origin AS for p. If AS2 receives any traffic for p it can loop it back to AS1.

5.4 Summary

Thus, we conclude that communities add even more worms to the routing can. To check their realizability we review the above scenarios and identify the following necessary and sufficient conditions:

Necessary condition: The above weaknesses of remotely triggered community actions can, in principle, be used if *communities are propagated beyond a single AS* and if the *community service is known.*

Sufficient condition: For the above weaknesses to be triggered a sufficient condition is that the attacker is able to advertise BGP prefixes with the appropriate communities, respectively hijack community tagged prefixes. Note, the propagation has to hold for all ASes on the path from the attacker to the community target.

We find, see Section 4, that the necessary conditions exist in the wild since communities are commonly propagated beyond their direct neighbors. Thus, we next identify sufficient conditions for each scenario by first setting up a *controlled experiment in the lab* and then showcasing at least one instance of each scenario in a *controlled experiment in the real Internet.* Hereby, we explicitly address ethical consideration, see Section 7.1.

6 TESTING THE FEASIBILITY

To better understand the feasibility of BGP community manipulation, and to expose nuances in their practical application and implementation, we performed a series of experiments within a controlled testbed. We experimented using Cisco 7200 routers running IOS 15.2(4)S7 (released in 2015) and Juniper routers running JunOS 12.1R1.9 (released in 2012). While this router hardware and software may not reflect the hardware currently deployed in the Internet, routers from these two vendors are heavily deployed. Thus, experiments with these two vendors can help shed light on what we might expect in the wild.

In the lab, we configured each of the scenarios from Section 5, relying on available vendor documentation, e.g., [27] and public documentation on community best practices, e.g., [56]. For reproducible, we make our configurations publicly available at https://www.cmand.org/caas/.

In addition to verifying necessary conditions for multi-hop community propagation, the laboratory experiments allow us to identify sufficient conditions for each scenario. We summarize our findings along three lines of insights, namely, i) propagation; ii) exploitation; and iii) misconfiguration.

6.1 Community Propagation

A cornerstone of our investigation is the necessary condition that a network path propagates BGP communities. As shown in Section 4, this condition frequently holds in practice; our controlled experiments help explain why some paths propagate communities while others do not.

Default behavior: While both Cisco and Juniper accept BGP updates with communities attributes by default, only Juniper propagates them by default. Cisco requires explicit per-peer or group configuration; a behavior that persists in both legacy and modern Cisco IOS implementations, including IOS XE. However,

since communities are often used to implement basic services one can expect that, even on Cisco routers, community propagation is typically enabled.

Adding communities: Both Cisco and JunOS provide configuration to add, subtract, or set communities to in-bound and out-bound prefixes. More complex behaviors are possible with regular expressions. In lieu of complex logic or error-prone expressions, we conjecture that some of the instances of community propagation we observe are due to simple and expedient configurations that use additive behavior for unknown communities.

Number of communities: Adding communities may come with the danger of exceeding the maximum number of communities per prefix. However, this is unlikely for the following reason: the BGP communities attribute is 4 bytes while the attribute length field is 2 bytes. Thus, a BGP update can carry up to $2^{16}/2^4 = 16K$ communities. Yet, Cisco only permits adding 32 distinct communities to a prefix, in addition to the communities the prefix arrived with. So, there is little risk in using the additive community propagation strategy, and no specific need to limit the number of communities carried in an announcement because most advertisements cross fewer than ten ASes [51].

6.2 Requirements for Exploitation

To exploit BGP communities we require the ability to trigger the community-based services. This is limited by which communities propagate along which route, as well as how routers resolve conflicting paths.

Community propagation: If a BGP router receives multiple routes to the same prefix with different communities, the ones of the best path are the only ones propagated. Thus, any attack needing to propagate a specific community from A to B must not only meet the necessary condition of A not stripping communities, but must also be chosen by A as the best BGP path.

Best route selection: A target that implements policies based on communities may receive announcements for the same prefix from two different peers. If only one of the announcements carries a community, BGP follows its standard route preference algorithm (e.g., shortest AS_PATH). However, implementations of RTBH may alter this preference (by setting a higher local preference).

6.3 Potential to Exploit

Third, it is well-known that production router configurations are complex and hard to validate [31]. We experienced, and thus expect, communities to further increase configuration complexity and, therefore, contribute to the potential success of BGP community misuse.

For example: Both Cisco and Juniper normalize communities within their configurations, when displaying BGP prefixes, and when sending BGP messages—by numerically sorting them. However, the order in which communities are evaluated depends on the configuration. Rules are evaluated in a specified order that is independent of the community value (and, indeed, non-community based rules may be preceded or followed by community rules). The difficulty of ensuring the correctness of such configurations, especially as a network grows and becomes more complex, is non-trivial. For example, we note that even simple configurations can exhibit unintended behaviors such as the snippet of Cisco router configuration that appeared in a NANOG tutorial on RTBH [56]. Here, the intent is to prevent hijacking by validating BGP announcements against a list of accepted customer prefixes. However, the route-map checks whether the prefix carries the blackhole community *before* performing the validation, thereby enabling hijacking-based attacks.

7 EXPERIMENTS IN THE WILD

To assess the real-world feasibility of the aforementioned scenarios (Section 5), we perform a number of experiments on the live Internet. Overall, we realize most of the scenarios in practice and gained a deep understanding of the requirements for, and difficulty of, successful attacks. For validation, we used a combination of (i) public BGP looking glasses [33]; and (ii) RIPE Atlas active probes [16]. Table 3 summarizes our major findings.

7.1 Ethical Considerations

Due to the inherently disruptive nature of the scenarios, we ensured prior coordination with, and permission of, the ASes and networks involved. To avoid the potential for collateral damage, the addresses and prefixes we use belong to networks that explicitly gave us permission to use as part of our experiments, even for hijacking-based attacks[9].

Our goal is to demonstrate that the weaknesses we identify are not merely theoretical, but present in the wild. For this, individual examples derived from our network partners suffice – we explicitly do not perform active Internet-wide experiments to assess the overall vulnerability, as doing so would pose undue operational and ethical risk.

We coordinate with the operators of three networks to target them as attackees and, respectively, target prefixes that were given to us as targets. We use two networks as attackers, i.e., prefix injection points, that were under our control: (i) the PEERING experimental platform [15, 54], and (ii) an experimental research network. Each of these points has its own ASN and can set arbitrary communities on announcements. We strictly follow the Acceptable Use Policy (AUP) [14] of PEERING and the research network; in particular, we only announce prefixes we control and with the correct respective origin ASN (i.e., no hijacking from PEERING).

7.2 Propagation Checking

To direct our in vivo experiments – again in consideration of both risk minimization and feasibility – we first infer community propagation behavior along the path from the attacker to the target using a benign community. We advertise a prefix tagged with the benign community from both of our injection platforms: PEERING and the research network. This benign community sets the high-order bits to the ASN of our injection point, and uses low-order bits that we have not observed in the wild. Our intent is to observe whether ASes propagate unknown communities, rather than to trigger any particular action.

[9]Therefore, in some sense, these are not true hijacks as we had permission to send the announcements from this origin.

We announce this prefix via a single physical location of the research network with two upstream providers. We find that only one of the upstream providers propagates communities. As observed at the route collectors, we see that seven transit providers further propagate the prefix with the community intact.

In contrast, the PEERING platform peers with hundreds of networks (via route servers at ten different points of presence), many of which propagate communities. Thus, it provides better visibility into the community propagation across a large number of paths, as observed at the route collectors. Across all of our available BGP views, we see more than 50 transit providers forwarding the prefix with the community within ~30 minutes of the initial announcement. Within a day, more than 112 transit providers (out of the 434 transit and origin ASes in the paths, as observed by the collectors in this study) were seen to propagate the prefix's community.

7.3 Remotely Triggered Blackholing

Informed by the benign community propagation inference, we find a provider that is two AS hops away from our injection point. While the benign community propagates to many ASes, we select a provider that both supports RTBH and offers a public looking glass. Because the target AS is not a direct peer of the attacker, this attack exploits the necessary conditions discussed previously.

Experiment: Using the target's blackhole community, we announce a /24 sub-prefix of our allocation (non-hijack). Next, we announce a /24 from a block of address space we had permission to hijack. Because of protections in place by the research network and its provider, the hijack based attack required updating the IRR [8]. While IRR validation adds a layer of defense for the hijack version of this attack, we note that many other injection points do not validate and, even when they do, it is often easy to circumvent [20, 58].

Validation: We examine the two prefixes (hijacked and non-hijacked) using the target's looking glass, before and after these announcements, as well as by sending active data plane probes using Atlas before and after (we ensure that the prefix contains an address that is responsive to ICMP echo requests). Further, the immediate upstream of our injection point provides a public looking glass enabling us to check community propagation. For all RTBH experiments, we saw that the prefix and community was accepted (target's looking glass). Further, we observed that the next-hop address for the prefix changed to a null interface address as result to the blackhole community. At this point, the target prefix was no longer reachable via the data plane tested using Atlas probes.

Additional constraints: To manage routing table growth and fragmentation, many providers enforce a limit on the maximum prefix mask length of announcements they will accept. In contrast, blackhole announcements typically must be for a /24 or more specific prefix. Some networks only accept blackhole announcements for a single host (a /32 prefix). Thus, an intermediate AS along the blackhole attack path must accept and propagate small prefixes if it is not aware of the target's blackhole community.

Summary: RTBH is the easiest scenario to realize in the wild, independent of hijacking. Unlike other attacks, we find that prefixes with blackhole communities are accepted independent of AS relationships, and are generally preferred even when the attacking AS path is longer.

At first blush, blackholing in conjunction with hijacking may seem redundant as they both impact reachability to the attackee's address space. However, an important distinction is that hijacking only poisons those ASes near the attacker, whereas a hijack-based blackholing attack drops traffic at the destination AS, thereby denying service universally.

7.4 Traffic Steering

Again, we leverage community propagation paths to identify potential targets for traffic steering attacks. An initially unexpected impediment is the role AS relationships play in traffic steering, as discussed below. We, therefore, relied on PEERING. We found a community propagating path from PEERING, through an intermediate provider, to a target AS that implements community-based steering. Since the intermediate provider is a customer of the target AS, the target AS accepts and acts upon the communities.

Experiment: Using the PEERING testbed, we advertise a prefix allocated to our experiment tagged first with the target's community to prepend the target's AS twice, and then with the target's community to lower the local preference to a value defined to be "customer fallback."

Validation: We primarily relied on looking glasses along the attack path, as well as public route collectors, to verify the steering attacks. Using the looking glass, we verified that the path prepending community was present at the target. We examined the prefix within public route collector views both before and during the attack, and verified that not only did the AS path change for many of the best paths received from peers, but also that the best path for many peers contained AS prepending for the target. Because of difficulties in finding an active Atlas monitor that uses the target AS as the best path toward our prefix, we relied on the looking glass within the target to verify the effect of the local pref community. Prior to the attack, we observed the prefix in the looking glass with the provider's default local preference, whereas during the attack we see it with the requested lower preference.

Additional constraints: Because of the AUP limitations on PEERING, we only implement the non-hijacking based multi-hop steering attack. Mounting the hijacking-based steering attack from the research network only successfully influenced the direct upstream. This limitation is largely due to our experimental environment, but does illustrate complications with steering attacks. While we verified the ability to prepend the AS path and local preference of our prefix within the immediate upstream provider of the research network injection point, we did not trigger similar behavior within an AS that was two hops away. The reason is that business relationships, either customer, provider, or peer, impact whether these communities are accepted and acted upon in practice (even when they propagate). Providers typically have different policies depending on the relationship type, and often *only* act on traffic steering communities that *arrive from a BGP customer* (operators maintain customer groups in their configuration files). Because the research network is a customer of a top-tier network that is not a customer of any other AS, we did not perform the multi-AS hop traffic steering attack from this injection point.

Summary: Access to multiple injection points is highly beneficial to orchestrate traffic steering attacks, and stub networks are preferred. On the other hand, given the flattening of the Internet hierarchy, in many cases, multiple levels of upstreams is no longer common. Thus, these types of attacks may be hard to launch.

7.5 Route Manipulation

To explore route manipulation, we utilize a well-known exchange point's route server that uses communities to control route redistribution. Not only does PEERING connect to this route server, it also provides a public per-peer view of the accepted prefixes and communities. This route server is a particularly attractive target as it publicly documents its community evaluation order and route server configurations – a policy common among neutral IXPs for transparency [1, 5, 19].

Table 3: Summary of our insights from our attacks in the wild.

Scenario	Hijack	Insights gained from running experiments in the wild	Difficulty
Blackholing	no	Allowed prefix length is checked; activation of RTBH service is typically required.	easy
	yes	Allowed prefix length is checked; origin validation was not always checked, thus the attack was easier.	easy
Traffic Steering with local pref	no	The business relationship of the attacker with the attackee or transit networks is checked – the flattening of the Internet makes this attacks hard to launch (providers only act on communities set by their customers).	hard
	yes	The business relationship of the attacker with the attackee or transit networks is checked – the flattening of the Internet makes this attacks hard to launch (providers only act on communities set by their customers); IRR records for origin validation are typically checked, but the check can be circumvented.	hard
Traffic Steering with path prepending	no	The business relationship of the attacker with the attackee or transit networks is typically checked – the flattening of the Internet makes this attacks hard to launch (providers only act on communities set by their customers); AS path prepending has typically low evaluation order, thus the attack may not be successful.	hard
	yes	The business relationship of the attacker with the attackee or transit networks is typically checked – the flattening of the Internet makes this attacks hard to launch (providers only act on communities set by their customers); IRR records for origin validation are typically checked, but the check can be circumvented; AS path prepending has typically low evaluation order, thus the attack may not be successful.	hard
Route Manipulation	no	Requires inference of community evaluation order when this information is not public.	medium
	yes	Requires inference of community evaluation order when this information is not public; IRR records for origin validation are typically checked, but the check can be circumvented.	medium

Experiment: From PEERING, we first sent our prefix with a community that instructs the route server to redistribute the route to a particular attackee AS. We then sent the prefix with this community and a community that suppresses advertisement to the attackee AS. Because these communities conflict, the resulting behavior exposes the route server configuration.

Validation: We utilize the looking glass of the attackee to determine whether or not the prefix was redistributed by the route server. Prior to sending the conflicting communities, we observe the announcement at the route server, whereas during the attack it is not present.

Summary: While we originated and tagged a prefix using an injection point that is a direct peer of the target route server, this is not a fundamental limitation. However, it does demonstrate that an intermediate provider, upon observing a community for the route server, can add a conflicting community to exploit the evaluation order in a similar fashion. Furthermore, because this route server has hundreds of peers, the potential impact can be large.

7.6 Applicability in the Wild

To obtain a more complete understanding of the real-world potential to mount remote community-based blackholing attacks, we conduct an automated experiment that explores the impact of each of the 307 verified blackhole communities identified in [36]. We employ the PEERING testbed to advertise a /24 prefix assigned to us (p), and RIPE Atlas to send active probes to this prefix between August 28 and September 1, 2018.

Specifically, for each community c in the set of blackhole communities, we: 1) advertise p without communities; 2) issue Atlas ICMP probes from 200 vantage points toward p; 3) advertise p with community c; 4) re-issue the same Atlas ICMP probes. Between each step, we wait five minutes to allow routing to converge, and for the Atlas probe requests to finish. The set of 200 Atlas vantage points is randomly chosen, but constant across all measurements. We then fetch the probing results and compare responses on a per-vantage point basis. We find 25 distinct communities (8.1%) that induce at least one vantage point to be fully responsive prior to advertising the community and then unresponsive once c is attached to the advertisement. These 25 communities affect a total of 48 (24%) of the vantage points.

To confirm that the community is the cause of the dropped Atlas probes, rather than some transient network event, we re-ran the experiment two days later. The results from this second round of probing exactly matched the first – suggesting that the root cause of the behavior we observe is indeed due to the blackhole communities being accepted and acted upon by various ASes along the path.

Finally, we issue traceroutes from all of the Atlas vantage points to our prefix p and use a current routeview routing table to naïvely map router interfaces to AS numbers. We then determine a lower-bound on the number of AS hops that the blackhole community traverses by finding the community's target AS in the path. 13 of 74 community-path pairs receive the blackhole community directly, i.e., the PEERING AS directly peers with the community's target AS. Four of the pairs involve two AS hops, while one involves three AS hops. Fully 75% of the community-path pairs that experience blocking due to the blackhole community did not have the community's target AS on the path, either because of non-AS specific communities (e.g., `65535:666`) or due to inaccurate IP-to-AS mapping. These results largely confirm our passive measurements. While the ASes directly connected to PEERING are expected to honor the advertised communities, we find further evidence of communities propagating multiple hops and being acted upon.

In addition to the testing of previously inferred blackhole communities, we note that the automated framework we describe here can also be used to gain confidence that a particular community is indeed used by a provider as a label for RTBH. This is particularly important when using machine learning or other statistical inference techniques to identify communities. For instance, while [36] found 307 verifiable blackhole communities, an additional 115 were labeled as likely communities. In future work we plan to test these additional communities.

Limitations: Similarly, we wish in the future to perform automated active experiments using non-RTBH communities to better understand their behavior. Such experiments require more complex inference as the resulting behavior can be subtle and hard to detect (e.g., a path change) as

compared to RTBH where reachability is a binary test. Moreover, following the PEERING AUP we are not allowed to conduct an automated experiment for traffic steering and route manipulation attacks as they can trigger path changes that can potentially impact the operation of the involved networks.

7.7 Other observations

In the process of running experiments in the wild, we find that it is possible to inject seemingly contradictory communities. As a case study, we inject fake location communities, i.e., communities used to tag a prefix's ingress reception point. We then observe the prefix at remote collectors labeled with communities indicating reception on different continents. Speaking with one large operator, we confirm that this will not disrupt the overall operation, as only a few customers will be affected. Nevertheless, we cannot exclude that other operators may rely on community-based location information in unanticipated ways, e.g., for traffic engineering or other operations.

8 DISCUSSION

BGP communities: Have we gone too far? The effectively global propagation of community tags allows attacks as shown above. On the other hand, BGP communities are used by network operators to implement policies and may add useful additional information, e.g., when debugging a network. They provide a low overhead simple communication channel between ASes. As such, they are widely in use. However, based on our interaction with the Internet operators community we gather, the scope of relevance for most communities is one or two hops. Based on that the Internet operators community needs to decide whether the benefits of easy communication outweigh the risks for potential attacks. An extreme way of preventing the kinds of attacks outlined in this paper would be the following: an AS only propagates communities which are useful to the receiving peer. To exemplify, AS1 should send to AS2 only communities of the form `2:xxx`. Au contraire, if AS2 is a route collector, such as RIPE RIS or Route Views, AS1 might not filter.

Be aware of standardized BGP communities: As we have shown with the remote blackhole attacks above, there are drawbacks for standardizing well-known transitive communities which have possibly disruptive semantics. On the other hand, those which are purely informative are much less of a concern. Having highly useful active communities not globally published might be called security through obscurity. Although, as we have shown, having a highly active community globally known makes life too easy for the attacker. As such, every operator of a network with known BGP communities and active semantics should be aware of the potential ramifications and have appropriate countermeasures in place, e.g., BGP community filters.

Need for BGP communities authentication: Clearly there is a strong need for the authentication of the right to attach a community to an announcement or modify one in transit. Unfortunately, there are no known means to do this. Moreover, adoption of authentication in Internet protocols has been shown to be a slow process, despite the critical role that the Internet plays in today's economy and society.

Monitoring the hygiene of BGP communities use: Abuse of communities might be discouraged by monitoring from the points of view of global BGP collectors such as RIPE RIS and Route Views, analogous to what is being done for BGP hijacks today. This strategy comes with all the problems of BGP monitoring: there is no global BGP view and route collectors only see the announcement they receive. The latter inferences on what happens on the path between the origin and the collector very difficult. In addition, the lack of structural semantics of BGP communities leaves a lot of room for misinterpretations. Of course, well known communities can and should be monitored. Yet, this only covers a small fraction of the available community space. Of course, monitoring BGP community behavior is not an active defense; but attribution of abuse might strongly discourage abuse.

Need for proper documentation: Similar to what is ongoing for bogon and other filtering, the operational community should publish and update well tested best current practices and configuration patterns for community generation, propagation, action semantics, etc.

9 CONCLUSION

In this paper, we measure the increasing use and propagation of BGP communities and demonstrate the resulting increase in exposure to abuse by remote parties; e.g., to blackhole prefixes, steer traffic, and manipulate routes. A key insight is that a significant fraction of transit providers, more than 14%, forward received communities onward. Given the rich interconnectivity of the Internet, this means that communities effectively propagate globally. Attacks are possible due in part to ill-defined BGP community semantics, error-prone configurations, as well as lack of cryptographic integrity and authentication for BGP communities.

By analyzing BGP announcements at many collectors worldwide we observe that, indeed, the propagation of communities is a global Internet phenomenon which enables routing vulnerabilities at scale. We tested the feasibility of BGP community-based attacks in lab experiments, then comment on the possibilities to launch such attacks in the wild. Unfortunately, such BGP attacks are successful even without prefix hijacking and even if BGP authentication is used. We highlight the need to increase awareness among current and future users of BGP communities regarding their possible abuse cases. We conclude that BGP communities are yet another highly used BGP feature which can, yet again, yield many unintended consequences.

As part of our future agenda we want to investigate ways to infer instances of any of the three types of BGP community-based attacks using passive measurements. This requires the development of a new methodology that assigns the role of the tagger of the BGP community to a network with the intent to perform one of the attacks described in the paper. Notice that both the relative position of the network in the path and the BGP community that it tags have to be considered for this inference. Identifying an attacker in BGP is not trivial due to the lack of authentication and integrity. We also want to investigate other types of BGP community-based attacks and assess their feasibility.

ACKNOWLEDGMENTS

We would like to thank the PEERING [15] team, especially Todd Arnold (Columbia University) who implemented the BGP communities attribute functionality in the PEERING infrastructure, and the operators of the networks that contributed network resources and volunteered to be "attackers" or "attackees" during our active experiments in the wild. We would also like to thank our shepherd, Priya Mahadevan, for her constructive comments.

This work and its dissemination efforts were supported in part by the European Research Council (ERC) grant ResolutioNet (ERC-StG-679158), by the Leibniz Prize project funds of DFG–German Research Foundation: Gottfried Wilhelm Leibniz-Preis 2011 (FKZ FE570/4-1), and the German Ministry for Education and Research as Berlin Big Data Center (funding mark 01IS14013A).

REFERENCES

[1] AMS-IX Deployment guides. https://ams-ix.net/technical/specifications-descriptions/ams-ix-route-servers/deployment-guides.
[2] AS Rank: CAIDA's ranking of Autonomous Systems. http://as-rank.caida.org.
[3] DE-CIX Frankfurt Route Server Guide. https://www.de-cix.net/en/locations/germany/frankfurt/routeserver-guide.
[4] DE-CIX Informational BGP Communities: Origin tagging. https://www.de-cix.net/de/resources/informational-bgp-communities.
[5] DE-CIX Operational BGP Communities. https://www.de-cix.net/en/resources/operational-bgp-communities.
[6] ECIX's New Route Server RTT Communities. https://www.ecix.net/about-us/news/ecixs-new-route-server-rtt-communities.

[7] Euro-IX: Large BGP Communities. https://www.euro-ix.net/en/forixps/large-bgp-communities/.
[8] IRR - Internet Routing Registry. http://www.irr.net.
[9] Isolario Project. https://isolario.it/.
[10] KPN BGP communities. https://as286.net/AS286-communities.html.
[11] NTT routing policies. https://www.us.ntt.net/support/policy/routing.cfm.
[12] Packet Clearing House Peering. https://www.pch.net/about/peering.
[13] Packet Clearing House Routing Archive. https://www.pch.net/resources/data.php.
[14] PEERING Acceptable Use Policy. https://peering.usc.edu/aup//.
[15] PEERING: The BGP Testbed. https://peering.usc.edu/.
[16] RIPE Atlas. https://atlas.ripe.net/.
[17] RIPE Routing Information Service. http://www.ripe.net/ris/.
[18] Routeviews Project – University of Oregon. http://www.routeviews.org/.
[19] Seattle Internet Exchange Route Servers. https://www.seattleix.net/route-servers.
[20] NANOG mailing list, October 4, 2017. "BGP hijack: 64.68.207.0/24 from as133955". https://mailman.nanog.org/pipermail/nanog/2017-October/092601.html, 2017.
[21] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai Botnet. In *USENIX Security Symposium*, 2017.
[22] M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. In *IEEE Symposium on Security and Privacy*, 2017.
[23] O. Bonaventure and B. Quoitin. Common utilizations of the BGP community attribute. IETF draft, work in progress, draft-bq-bgp-communities-00.txt, June 2003.
[24] R. Bush and R. Austein. The Resource Public Key Infrastructure (RPKI) to Router Protocol, January 2013. IETF RFC 6810.
[25] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. IETF RFC 1997, 1996.
[26] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. There is More to IXPs than Meets the Eye. *ACM CCR*, 43(5), 2013.
[27] CISCO. Remotely Triggered Black Hole Filtering - Destination Based and Source Based. Cisco White Paper, http://www.cisco.com/c/dam/en_us/about/security/intelligence/blackhole.pdf, 2005.
[28] C. Dietzel, A. Feldmann, and T. King. Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild. In *PAM*, 2016.
[29] B. Donnet and O. Bonaventure. On BGP Communities. *ACM CCR*, 38(2), 2008.
[30] P. Faratin, D. D. Clark, S. Bauer, W. Lehr, P. Gilmore, and A. Berger. The Growing Complexity of Internet Interconnection. *Communications and Strategies*, 72, 2008.
[31] N. Feamster and H. Balakrishnan. Detecting BGP configuration faults with static analysis. In *NSDI*, 2015.
[32] D. Gillman, Y. Lin, B. Maggs, and R. K. Sitaraman. Protecting Websites from Attack with Secure Delivery Networks. *IEEE Computer Magazine*, 48(4), 2015.
[33] V. Giotsas, A. Dhamdhere, and kc claffy. Periscope: Unifying Looking Glass Querying. In *PAM*, 2016.
[34] V. Giotsas, C. Dietzel, G. Smaragdakis, A. Feldmann, A. Berger, and E. Aben. Detecting Peering Infrastructure Outages in the Wild. In *ACM SIGCOMM*, 2017.
[35] V. Giotsas, M. Luckie, B. Huffaker, and kc claffy. Inferring Complex AS Relationships. In *ACM IMC*, 2014.
[36] V. Giotsas, G. Smaragdakis, C. Dietzel, P. Richter, A. Feldmann, and A. Berger. Inferring BGP Blackholing Activity in the Internet. In *ACM IMC*, 2017.
[37] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and kc claffy. Mapping Peering Interconnections at the Facility Level. In *CoNEXT*, 2015.
[38] V. Giotsas, S. Zhou, M. Luckie, and kc claffy. Inferring Multilateral Peering. In *CoNEXT*, 2013.
[39] S. Goldberg. Why is It Taking So Long to Secure Internet Routing? *Comm. of the ACM*, 57(10), 2014.
[40] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *NDSS*, 2003.
[41] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg. From the Consent of the Routed: Improving the Transparency of the RPKI. In *ACM SIGCOMM*, 2014.
[42] J. Heitz, J. Snijders, K. Patel, I. Bagdonas, and N. Hilliard. BGP Large Communities Attribute. IETF RFC 8092, 2017.
[43] Y. C. Hu, A. Perrig, and M. Sirbu. SPV: Secure Path Vector Routing for Securing BGP. In *SIGCOMM*, 2004.
[44] G. Huston. Peering and Settlements: Part I. *The Internet Protocol Journal*, 2(1), 1999.
[45] G. Huston. Peering and Settlements: Part II. *The Internet Protocol Journal*, 2(2), 1999.
[46] G. Huston. NOPEER Community for Border Gateway Protocol (BGP) Route Scope Control, April 2004. IETF RFC 3765.
[47] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins. BLACKHOLE Community. IETF RFC 7999, 2016.
[48] M. J. Levy. Using BGP communities to control your transit providers. APRICOT 2013, 2013.
[49] J. Mitchell. Autonomous System (AS) Reservation for Private Use, July 2013. IETF RFC 6996.
[50] O. Nordstrom and C. Dovrolis. Beware of BGP attacks. *ACM CCR*, 34(2), 2004.
[51] B. Quoitin, S. Uhlig, and O. Bonaventure. Using Redistribution Communities for Interdomain Traffic Engineering. In *QofIS'02/ICQT'02*, 2002.
[52] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger. Peering at Peerings: On the Role of IXP Route Servers. In *ACM IMC*, 2014.
[53] A. Robachevsky. 14,000 Incidents: A 2017 Routing Security Year in Review. Internet Society, https://www.internetsociety.org/blog/2018/01/14000-incidents-2017-routing-security-year-review/, January 2018.
[54] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, and E. Katz-Bassett. PEERING: An AS for Us. In *HotNets*, 2014.
[55] P. Smith. BGP Techniques for Internet Service Providers. NANOG 50, 2010.
[56] J. Soricelli and W. Custavus. NANOG Tutorial: Option for Blackhole and Discard Routing. https://www.nanog.org/meetings/nanog32/presentations/soricelli.pdf, Oct 2004.
[57] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal. RAPTOR: Routing Attacks on Privacy in Tor. In *NSDI*, 2015.
[58] A. Toonk. Using BGP to find Spammers. https://bgpmon.net/using-bgp-data-to-find-spammers, 2014.