SICO: Surgical Interception Attacks by Manipulating BGP Communities

Henry Birge-Lee
birgelee@princeton.edu
Princeton University

Liang Wang
lw19@princeton.edu
Princeton University

Jennifer Rexford
jrex@cs.princeton.edu
Princeton University

Prateek Mittal
pmittal@princeton.edu
Princeton University

ABSTRACT

The Border Gateway Protocol (BGP) is the primary routing protocol for the Internet backbone, yet it lacks adequate security mechanisms. While simple BGP hijack attacks only involve an adversary hijacking Internet traffic destined to a victim, more complex and challenging interception attacks require that adversary intercept a victim's traffic and forward it on to the victim. If an interception attack is launched incorrectly, the adversary's attack will disrupt its route to the victim making it impossible to forward packets. To overcome these challenges, we introduce SICO attacks (Surgical Interception using COmmunities): a novel method of launching interception attacks that leverages BGP communities to scope an adversary's attack and ensure a route to the victim. We then show how SICO attacks can be targeted to specific source IP addresses for reducing attack costs. Furthermore, we ethically perform SICO attacks on the real Internet backbone to evaluate their feasibility and effectiveness. Results suggest that SICO attacks can achieve interception even when previously proposed attacks would not be feasible and outperforms them by attracting traffic from an additional 16% of Internet hosts (worst case) and 58% of Internet hosts (best case). Finally, we analyze the Internet topology to find that at least 83% of multi-homed ASes are capable of launching these attacks.

CCS CONCEPTS

• **Security and privacy** → **Network security**; • **Networks** → **Network security**; **Public Internet**; *Routing protocols*; *Network experimentation*; Network simulations.

KEYWORDS

networking; security; BGP; hijacking attacks; interception attacks; BGP communities

ACM Reference Format:
Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. 2019. SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3319535.3363197

1 INTRODUCTION

The Border Gateway Protocol (BGP) allows ISPs throughout the world to exchange routing information and is the primary routing protocol for the backbone of the Internet. However, because BGP was first drafted in 1989 [71], BGP contains *no* means of cryptographically verifying the authenticity of routes which allows an Autonomous System (AS) to lie about what routes it has. This fundamental flaw in BGP allows for BGP attacks where an adversary announces a route in BGP that it does not actually have. BGP attacks are routinely seen in the wild and have compromised sensitive communications from cryptocurrencies [49] to financial services data [60].

In a simple BGP attack (known as a BGP hijack) the adversary attracts traffic for the victim's prefix, and either answers or drops that traffic. However, more advanced attacks (like traffic analysis against Tor [83]) require an adversary to *intercept* network traffic and forward it on to the intended recipient (i.e., the victim of the attack). These BGP interception attacks are more difficult to perform because the adversary must successfully forward packets to the victim. This is a key challenge, since the adversary's BGP announcement can disrupt its own valid route to the victim, making the adversary unable to deliver packets to the victim and perform interception.

Contributions: In this paper, we present SICO (**S**urgical **I**nterception using **CO**mmunities) attacks, a novel method of performing BGP interception attacks that increases both the viability and effectiveness of these attacks by exploiting *BGP communities*. BGP communities can be used by an AS to influence the propagation of its route at remote ASes, which is commonly used for network traffic engineering. However, as in our attacks, this feature also helps an adversary to control the propagation of a malicious route. By using communities, SICO restricts the propagation of the adversary's BGP announcement to *only* where necessary to achieve interception, while the adversary's route to the victim is still preserved. This fine-grained propagation control enables SICO to achieve traffic interception in cases when using previously proposed attacks [46, 59] would be difficult or even impossible.

Furthermore, we extend SICO to allow for *targeted* interception. In this variant, an interception attack is engineered to affect only select source IP addresses and affect as less Internet as possible. We achieve targeted interception by using BGP communities to suppress unwanted route propagation while still attracting traffic from the target source IP addresses.

CCS '19, November 11–15, 2019, London, United Kingdom
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6747-9/19/11.
https://doi.org/10.1145/3319535.3363197

Targeted interception decreases the detectability of an attack because fewer ASes would be seeing the malicious route. In addition, targeted interception allows the adversary to more realistically handle the bandwidth required during an interception attack, reduces the cost of performing interception, and minimizes the effect on round-trip time introduced by interception.

We evaluate SICO attacks by launching them on the real Internet backbone in an ethical manner (i.e., attacking our own IP prefixes), and study the Internet topology to better understand how many ASes can launch SICO attacks. Some highlights of our results include:

- We verified the feasibility of SICO attacks under various AS topologies.
- SICO outperforms previous techniques by allowing an adversary to attract traffic from an additional 16% of Internet hosts (worst case) and 58% of Internet hosts (best case).
- When targeting just the IP of the highest bandwidth Tor node, targeted interception attacks can effectively reduce the number of intercepted hosts compared to previous attacks, while still being able to intercept the traffic from the target IP to the victim.
- Our evaluation of the viability of SICO suggests that *at least* 48% of total ASes (or 83% of multi-homed ASes) are capable of launching SICO.

We hope that our work serves to inspire the real-world deployment of secure countermeasures, including RPKI [50] and BGPSec [69], which have not been widely deployed yet.

2 OVERVIEW OF BGP

2.1 BGP routing policies

The Border Gateway Protocol (BGP) allows independently operated networks (known as Autonomous Systems or ASes) to exchange routing information with each other. In BGP, an AS makes a BGP announcement to its neighbors to advertise its routing information (IP prefixes), and includes its Autonomous System Number (ASN) in the *AS-path* field of the announcement. The neighbors then decide if this BGP announcement represents their preferred way to route packets for a given IP prefix. If so, these neighbors can further forward this announcement to their neighbors. We introduce three aspects of BGP in this section.

Filtering routes with loops. When announcements are forwarded, ASes add their own ASNs to the AS-path field so that AS-path contains a list of all the ASes the packets will traverse to reach their destination. The AS-path field prevents loops because an AS will not import a route if its own ASN is already in the AS-path field of the route announcement [75].

Selecting a route. When an AS hears two BGP announcements for the same IP prefix, it uses a series of tiebreakers to determine which route it will use. The first tiebreaker is *local preference*. Local preference is AS-specific and is often based on which type of neighbor a route is learned from. Routes learned from customers are preferred over routes learned from peers, which are preferred over routes learned from providers. The next tiebreaker is AS-path length: ASes prefer routes with shorter AS paths. Finally, in the case of a tie on both local preference and AS-path lengths, routes are compared based on the Interior Gateway Protocol (IGP) [1] metric of the next-hop router for each route. We note that other tiebreakers also exist, but do not impact our attacks, as discussed in Appendix §A.

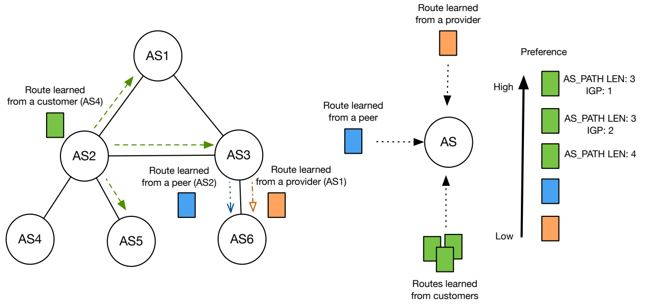

Figure 1: Examples of BGP export rules (left) and route preference (right). Green, blue, and orange rectangles represent routes learned from customer, peer, and provider, respectively.

The above tiebreakers are only used in the case of BGP announcements for the exact same prefix. If a router hears a BGP announcement for a more specific prefix and an announcement for a shorter, more general prefix, the route for the more specific prefix is always used.

Exporting routes based on business relationships. Based on the Gao-Rexford model [55], the main business relationships between ASes are *customer-provider* and *peer-to-peer*. An AS A is a customer of a neighbor B (i.e., the provider) if A pays B for accessing Internet, and is a peer of a neighbor C if A and C can exchange traffic between each other and between their customers free-of-charge. The type of neighbor the routes are learned from determines the neighbors the routes will be announced to. More specifically, routes learned from customers are announced to all neighbors, but routes learned from peers and providers are only announced to customers. See Figure 1 for an example.

2.2 BGP interception attacks

BGP attacks involve an AS making BGP announcements to maliciously attract traffic destined to another AS's prefix, and have been traditionally divided into two categories based on how the attacks impact the data plane [46]. The first category is BGP hijack attacks where an adversary uses a malicious BGP announcement to attract traffic destined to a victim AS, but the adversary does not actually deliver this traffic to the victim. The second category of BGP attacks is interception attacks where an adversary attracts traffic

[1] IGP is the routing protocol used to route traffic *within* an AS (e.g., OSPF).

destined to a victim and then routes this traffic through to the victim.

Motivation for interception. The capability of forwarding intercepted traffic back to the victim AS enables interception attacks to bootstrap more sophisticated attacks, such as traffic correlation attacks against anonymous networks [83] and man-in-the-middle attacks against certificate authorities [47]. Though hijack attacks can be effective for many adversarial objectives (e.g., setting up phishing websites and spoofing DNS responses [49]), they disrupt connectivity for hosts in the victim's network. In contrast, interception attacks preserve connectivity in the data plane, making them much harder to detect than hijack attacks (as seen in [77, 86], data-plane connectivity is a common method for detecting hijack attacks).

Methods of maintaining connectivity. In interception attacks, the adversary builds valid route(s) from an adversary AS to the victim AS via either announcement shaping (strategically crafting bogus BGP announcements so that the adversary's AS itself still has a valid route to the victim) or tunneling (encapsulating the traffic and sending it to a remote destination with a valid route to the victim where it is unencapsulated). We focus on announcement shaping because tunneling requires either a colluding AS, which is beyond the scope of our threat model, or a remote host that is capable of spoofing source IP addresses to make the tunneled traffic have different source IP addresses).[2] Based on the CAIDA spoofer project [42] only 8% of IP blocks allow end hosts to spoof source IP addresses, meaning that it may be difficult for an adversary to find an acceptable end host to use for tunneling. In addition, tunneling incurs significant additional communication resources (since the adversary must now route the victim's traffic through the Internet at each tunneling end-point instead of only at its own AS) and needlessly increases TCP latency when compared to announcement shaping. Finally, we later demonstrate that announcement shaping can be extended to launch targeted attacks, which cannot be achieved with tunneling alone.

Achieving announcement shaping. To achieve announcement shaping, the adversary usually adopts two techniques: *AS-path poisoning* [75] and *selective neighbor announcement* [46, 82].

In AS-path poisoning, the adversary adds a valid route to the victim in the AS path of the bogus announcement and announces that the adversary AS can reach the victim via that route. The ASes on the valid route between the adversary and the victim will ignore this announcement because of BGP loop prevention and deliver the traffic from the adversary normally to the victim, while the other ASes that are not on that route may prefer the bogus announcement and deliver their traffic to the adversary.

$R2$ src $R1$ src	**Customer**	**Peer**	**Provider**
Customer	—	$R1$	$R1$
Peer	$R2$	—	$R1$
Provider	$R2$	$R2$	—

Table 1: The route an AS prefers when learning both $R1$ and $R2$ from different types of neighbors. "—" indicates the AS needs to consider other factors to make a decision.

Selective neighbor announcement exploits routers' local preference to prevent routing loops. The adversary announces to selected neighbors that the adversary AS originates the victim's IP prefix, based on the business relation between the adversary, the victim, and their neighbors. To help better understand this process, we show the route an AS prefers when learning two routes from different types of neighbors in Table 1. For instance, if the adversary delivers traffic to the victim using a valid route learned from a customer or peer, the adversary can announce the bogus route to all its neighbors. The announcements for the valid route only traverse customer-provider edges. Because of the business relationship preferences discussed above, all of the ASes along the valid route will ignore the bogus route, since they will learn the bogus route from a provider or peer (unlike the valid route that is heard from a customer).

2.3 BGP community

BGP communities are optional attributes that can be added to a BGP announcement for controlling the routing policy in upstream ASes. There are a small set of standardized communities defined by RFCs (e.g., RFC 1997, RFC 3765, RFC 7999 [51, 61, 64]). However, the vast majority of community use is non-standardized and varies from AS to AS.

Previous works have proposed more extensive standardization as well as security improvements for BGP communities [76], but these proposals have not seen widespread adoption. Although communities are often not standardized, there are common themes in how communities are used which have been explored by previous work [53, 81]. One type of communities is *information communities* that are added to a route by an AS to signal properties about that route (e.g., what Internet exchange the route was learned at or whether it was learned from a peer or customer). Another type is *action communities*, which are added to a route to cause an AS further down the path to perform a specific action related to this route. A common example of a community performing an action is the use of communities to remotely trigger black hole filtering (as documented in [67]). Despite common themes, there are no limits on the potential uses of communities because any community can be matched against

[2] Alternately, an adversary could use network address translation with overloading (a.k.a. port address translation) to rewrite the source IP address of all traffic that is passing through and avoid the need to spoof, but this will break any connections that are initiated by the victim's end hosts and cause a noticeable anomaly since a large number of connections are from the same source IP address.

in a router's configuration and can be used to trigger any action the network administrators would like.

Action communities are meaningfully transited **UP** (from customers to providers) the Internet hierarchy, but not over (across peering links) or down (across provider-customer links). Many ASes do not accept communities from peers[3] [13, 33, 81]. Accepting communities from providers also works against the interest of an AS, since action communities can limit the propagation of routes (which the AS is paying its providers for).

Recent work has begun to explore communities in an adversarial setting. Streibelt et al. explored how an adversary can exploit remotely triggered blackholing, traffic steering, and route manipulation for adversarial purposes [80, 81]. We build on this line of work and are the first to consider communities in the context of interception attacks and in strategically limiting announcement propagation for adversarial purposes.

In addition, several efforts have begun to gain popularity which standardize security-grounded community values. For example RFC 7999 [64] standardizes the use of the BLACKHOLE community which triggers blackholing (and can be used to act upon remotely triggered black hole lists [62]) and clearly outlines a secure implementation that avoids potential exploitation (e.g., only accepting the BLACKHOLE community for routes a customer is authorized to announce). Even beyond the general BLACKHOLE community, recent work has proposed a method to communicate port and protocol specific blackholing via BGP communities [52].

3 ATTACK OVERVIEW

Previous state-of-the-art attacks have severe limitations that make them *infeasible* or *ineffective* in many real-world scenarios. However, through the use of fine-grained announcement propagation control offered by BGP communities, SICO attacks can overcome these challenges.

3.1 Threat Model

We assume the adversary can control at least one AS (denoted by AS_{adv}), either via gaining full control or by compromising an AS's border routers, and is able to make arbitrary BGP announcements to neighboring ASes. The goal of the adversary is to get some traffic destined to a victim's IP prefix to route through AS_{adv}, and be able to route the traffic to the AS, denoted by AS_{vic}, that the victim's IP prefix belongs to (i.e., achieving traffic interception as opposed to simply hijacking).

The interception attacks performed by the adversary can be either targeted or untargeted. In an untargeted attack, the adversary wants to intercept as much traffic as possible. In a targeted attack, the adversary is more interested in intercepting the traffic destined to the victim from given target IP addresses, and may want to *reduce* the traffic from the rest of the Internet (i.e., untargeted IP addresses). See §3.2 for more discussions.

Symbol	Description
AS_{adv}	AS controlled by the adversary
AS_{vic}	AS for the victim's IP prefix
AS_{tar}	AS for the target IPs in targeted interception attacks
A, B	Providers of AS_{adv} B is used by the adversary for forwarding the intercepted traffic to the victim
R	Route from B to AS_{vic} (learned by B)
R^*	Route from B to AS_{adv} (learned by B)
X, Y, Z	Arbitrary AS
$\mathrm{R}(X)$	Route from AS X to AS_{vic} (learned by X)
$\mathrm{R}^*(X)$	Route from AS X to AS_{adv} (learned by X)
$\mathbb{R}^*(X)$	Set of all routes from AS X to AS_{adv} (learned by X)

Table 2: Notation used throughout this paper.

The adversary claims that AS_{adv} can reach the victim's IP prefix via some routes in the bogus BGP announcements. A legitimate route from AS_{adv} to the victim's IP prefix is leveraged by the adversary to route the intercepted traffic to the victim. Note that the actual routing path used by the adversary for reaching the victim's IP prefix may not necessarily be the same as the path claimed in the bogus announcements. We use $|r|$ to denote the length (i.e., number of ASes) of a route r.

3.2 Limitations of previous attacks

AS topological diversity could make AS-path poisoning and selective neighbor announcement severely limited in their effectiveness, or even infeasible. In many cases, ASes have no neighbors other than providers to whom they can make BGP announcements. (73% of ASes have neither peers nor customers according to the CAIDA March 2019 AS-relationship dataset [12]). For AS-path poisoning, large ASes (like tier-1 providers) often deploy *defensive AS-path filtering*, which blocks BGP announcements from customers that contain the ASN of another tier-1 provider anywhere in the AS path [79]. Thus, if an adversary needs to poison the ASN of a tier-1 provider for AS-path poisoning to be successful, the adversary's announcement may not be propagated by other tier-1 providers which significantly hinders announcement propagation. Besides (even if a tier-1 provider's ASN is not included in the AS path), an increased AS-path length will globally lower the attractiveness of the adversary's route which also limits announcement propagation. Selective neighbor announcement in many cases may not work at all because the rich interconnectivity of the Internet topology (especially among ASes with geographic proximity to each other, as is the case with the providers of a geographically small adversary) often causes all the providers of AS_{adv} to route

[3] A notable exception to this rule is the behavior of route servers at Internet exchanges that often use communities to signal which peers to announce to [19].

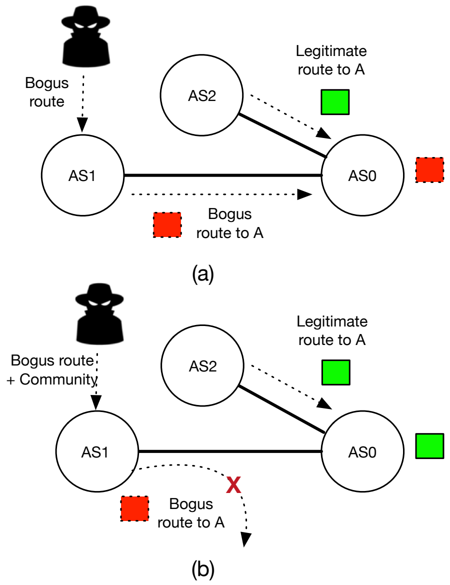

Figure 2: With no communities, $AS0$ imports the bogus route hindering interception. SICO attacks use communities to overcome these scenarios by strategically limiting the announcement propagation of bogus routes.

traffic destined to AS_{vic} back to AS_{adv}. We experimentally demonstrate these limitations in §5.

Moreover, previous attacks provide limited or no support for targeted interception attacks. In some cases, targeted interception attacks are advantageous to the adversary for the following reasons: (1) If the victim has high-volume traffic, the adversary may not have enough resources in its routers to handle the intercepted traffic, resulting in a significant and noticeable reduction in performance (e.g., higher TCP latency) for the victim. Such performance degradation could be used to detect interception attempts. (2) Besides, even if the adversary has the required resources to handle the traffic towards the victim, the more traffic the adversary gets, the more expensive the attack becomes — ultimately, the adversary must pay its upstreams for the additional bandwidth used. Unfortunately, previous attacks are insufficient for targeted interception because they offer little control over which source ASes redirect traffic towards the victim via the adversary.

3.3 Our attacks

To overcome the aforementioned limitations, we develop a novel interception attack that provides fine-grained control over announcement propagation that we call SICO (**S**urgical **I**nterception using **CO**mmunities) attack. SICO uses BGP communities to manipulate the local preferences and announcement exporting behaviors of the routers in neighboring or remote ASes in order to control the propagation of bogus announcements. As a result, selected ASes will never hear or will not prefer the bogus announcements, and thus always use a valid route to forward the traffic from the adversary to the victim. As illustrated in Figure 2 (a), $AS0$ hears the bogus route from a peer *AS1* and the valid route from a provider *AS2*. Because of local preference, $AS0$ may import the peer-learned route (i.e., the bogus route), which could be problematic for interception attacks if $AS0$ is used by the adversary for routing traffic. Using SICO, the adversary can prevent such failure by sending a community along with the bogus route to notify *AS1* to stop exporting this route to $AS0$, as in Figure 2 (b), while the propagation of the other (valid) routes will not be affected. Besides, SICO does not need to modify AS-path in an announcement, which bypasses some AS-path-based filtering mechanisms or detection techniques.

Further, we leverage SICO to develop targeted interception attacks, which allows the adversary to intercept the traffic to a victim IP prefix *from given target source IP addresses*, while leaving as much of the Internet "untouched" (i.e., not delivering traffic to the adversary) as possible. In targeted interception attacks, the data-plane effect of the interception becomes less noticeable and the cost of performing the attack is reduced.

4 BGP COMMUNITY BASED INTERCEPTION ATTACKS

Our key insight is that, to achieve interception, an adversary sometimes needs to control the behaviors of routers in other ASes beyond what is achievable by simply selecting which neighbors to announce to or using AS-path poisoning. We achieve this via BGP communities.

4.1 Attack setup

Assume without loss of generality that AS_{adv} has two providers A and B, and announces a bogus BGP route to A and routes the intercepted traffic to AS_{vic} through B. [4] We assume in the bogus announcement the adversary claims that AS_{adv} originates the victim's IP prefix(es). B learns R, a legitimate route from B to AS_{vic}, and R^*, the route from B to AS_{adv}, from some neighbors. From B's perspective, both R and R^* appear to be the legitimate routes to AS_{vic}.

The key challenge in interception is to maintain a valid route to the victim. Thus, we want the ASes along the path from the adversary to the victim (such as B) to **NOT** prefer the adversary's bogus route so that the adversary can deliver traffic to the victim normally.

We focus on the scenario that AS_{adv} does not have any peers or customers, since this is the most challenging scenario for launching interception attacks (as discussed in §3.2). Even if an adversary has peers or customers, it can propagate its attack further by making a malicious announcement to providers as well.

[4] In cases where an adversary has more than two providers, we can apply the same reasoning to a two-provider subset of the adversary's providers.

4.2 Attack toolkit: Communities that can enable interception attacks

With substantial resources invested in their IP networks, large ASes usually have support for a wide range of action communities. Another hotspot for community support is Internet exchanges points (IXPs) which often have route servers that support communities. Despite the specific community support varying from network to network, we find several common use cases throughout many large ASes and route servers. Specifically most of the top 30 ASes and top 10 Internet exchanges we studied support the following community actions (see Appendix §C for details):

- **Lower local-preference below peer** (LowerPref): This community action allows a customer to lower the local preference of its routes below default local preference of peer routes. For instance, a tier-1 provider, who learns R from peers and learns R^* from a customer (i.e., the adversary), will prefer R over R^* if the adversary has applied this action to R^*.
- **No export to select peer** (NoExportSelect): This community action causes a tier-1 provider to not export a route to specific peers. The tier-1 provider exports a route to all peers with the exception of the peers specified (by their ASNs) in the community string.
- **No export to all peers** (NoExportAll): Here, a tier-1 provider will only use a route among its customers and not share the route with any peers[5]. This has a very adverse impact on route propagation, but is sometimes needed to maintain a valid route to the victim. This is one of the most common action communities and is even standardized through an RFC [61].

Overall, 8 of the top 10 Internet exchanges we studied supported these three communities at their route servers and 21 of the top 30 ISPs supported all of these communities as well (see Appendix §C). We found one Internet exchange that did not support communities and one that we could not get information on. Of the 9 ISPs that did not offer full community support, 5 of them offered partial community support that could still be used to facilitate attacks. For 2 ISPs we were unable to find information regarding community support, and for only 2 ISPs we found evidence that relevant action communities were not supported.

We use the three communities as the "gadgets" to construct our attacks. Note that NoExportAll can actually be replaced with NoExportSelect; therefore, we always prefer to use NoExportSelect because it enables more fine-grained route propagation control, and only use NoExportAll as a fallback when NoExportSelect is not supported. [6] Many providers support community usage *beyond* this model that can be used to improve the effectiveness of attacks (we use some of these communities in §5), but the above model is commonly supported and is sufficient for enabling interception attacks. We will show how to use these gadgets to achieve interception in various AS-level topologies next.

4.3 Case study: propagation control via community

Recall that an interception attack will fail if B prefers R^* over R (i.e., B believes it is better to reach the victim via AS_{adv}) or if B does not hear R from any neighbors. In both cases, when the adversary tries to use the legitimate route from B to AS_{vic} to forward the intercepted traffic, B will follow R^* and forward the traffic back to AS_{adv}, and thus fails to deliver the intercepted traffic to the victim.

As discussed in §2.1, a router needs to examine a set of metrics (local preference, AS-path length, and IGP metric) to determine the preferred route between multiple options. Next, we demonstrate how each metric could cause failures of interceptions using three representative cases, and discuss how to use communities to achieve interception in the three cases. For a more general algorithm for achieving interception in all cases, see §4.4.

Local preference. As in Figure 3 (a), if A and B are (1) not tier-1 and (2) peer with each other, B will learn R^* as a two-hop peer route. Since neither A or B are tier-1s, B will most likely learn R from a provider. Based on Table 1, R^* is preferred over R.

Solution: To solve this issue, the adversary can use the NoExportAll or NoExportSelect community to prevent A from exporting R^* to B. In addition, many peerings are facilitated by route servers at Internet exchanges. If A and B peer through a route server, even if A does not support any action communities, the route server may support communities that can suppress the announcement of R^* to B.

AS-path length. In Figure 3 (b), A and B are (1) not tier-1 and (2) share a common tier-1 provider. Because of sharing the tier-1 provider with A, B learns R^* as 3-hop-long route (i.e., $|R^*| = 3$) from the tier-1 provider. If B does not also share a tier-1 provider with the victim, B will likely learn R as a 4-hop-long path ($|R| = 4$) from a different tier-1 provider. In this case, B will once again prefer R^*.

Solution: To overcome this, the adversary can simply use the LowerPref community to reduce the preference of R^* at the shared provider. This causes the shared tier-1 to prefer R and announce R to B, eliminating the problem caused by the shared provider.

IGP metric. In Figure 3 (c), both A and B are tier-1s. Assume AS_{adv} and AS_{vic} are at the same level of the Internet hierarchy. Therefore, B will learn R and R^* as equal-length paths ($|R| = |R^*|$).

Which path (R or R^*) B will prefer depends on the IGP metric, so there is a chance that B will prefer R, allowing the adversary to launch an attack with no communities. However, there is also a significant chance B will prefer R^* hindering the adversary's interception attack.

[5] NoExportAll is not to be confused with the well-known RFC 1997 community NO_EXPORT which prevents export to peers and customers [51]. NoExportAll is a provider-specific community that only restricts export to peers, not customers (i.e., a provider-specific version of NO_PEER [61]).

[6] The exception to this is in targeted attacks where increased spread is non-optimal and NoExportAll may be preferable to NoExportSelect.

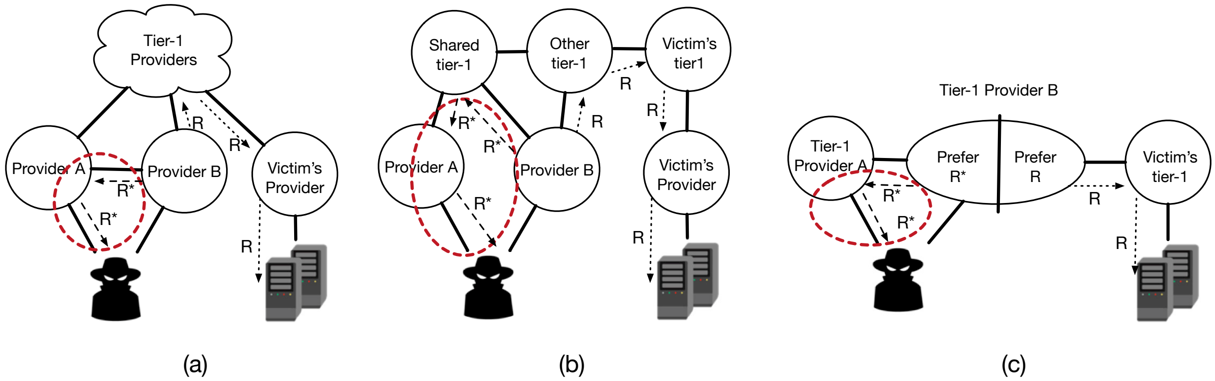

Figure 3: R^* (dashed lines) and R (dotted lines) learned by B in three different cases: (a) The adversary has two non-tier-1 providers that peer with each other. (b) The adversary has two non-tier-1 providers that share a common tier-1 provider. (c) The adversary has two tier-1 providers. The red cycles highlight the route B prefers.

Solution: Similar to Figure 3 (a), this situation can be remedied by placing peer export controls on A that stop it from exporting R^* to provider B. Ideally the adversary would use the NoExportSelect community to allow its bogus route to be propagated as far as possible without being announced to B. Alternatively, the adversary can simply use the NoExportAll community to suppress exporting and rely on the route propagating to A's clients.

4.4 Launching attacks in the general case

While the above three cases serve as a demonstration of our attacks, the algorithm below allows an adversary to launch these attacks with arbitrary topological relationships between AS_{adv} and AS_{vic}. This algorithm also allows an adversary to be assured an attack will be successful before launching the attack (as to not needlessly raise suspicion by attempting to launch faulty attacks).

Let us use the notation $\mathrm{R}(X)$ to be the route to the victim used by AS X and $\mathrm{R}^*(X)$ to be the route to the adversary used by AS X. Also let $\mathbb{R}^*(X)$ be the *set of all routes* to the adversary heard by AS X. Because an AS cannot use a route that was not announced to it, $\mathrm{R}^*(X) \in \mathbb{R}^*(X)$.

Our algorithm consists of four steps: `MakeSampleAnnouncement`, `CollectInfo`, `AddCommunities`, and `LaunchAttack`.

(1) `MakeSampleAnnouncement`: The best way for the adversary to understand topological relationships is by observing real route propagation. To do this, the adversary should announce its *own* prefix to A and allow this announcement to fully propagate and let Internet routes to converge to a stable state.

(2) `CollectInfo`: Next, for each AS X in $\mathrm{R}(B)$, the adversary should examine to see if *any member* of $\mathbb{R}^*(X)$ will be preferred by X over $\mathrm{R}(X)$ based on the information in Table 3. For each member of $\mathbb{R}^*(X)$ preferred over $\mathrm{R}(X)$, the adversary should suppress this route with communities.

(3) `AddCommunities`: With knowledge of which routes must be suppressed, the adversary can add communities to strategically limit its announcement propagation. To suppress a given route r, the adversary should:

- If r contains a peering link from, say, AS Y to AS Z, the adversary should apply NoExportSelect at AS Y towards AS Z.
- If the peering link in r is facilitated by a route server, the adversary should additionally apply NoExportSelect at AS Y towards the route server or ideally apply NoExportSelect at the route server towards AS Z (so that other peers at the route server will still hear the announcement from AS Y).
- If r does not contain a peering link, apply LowerLocalPref at the highest provider in the route.

The adversary iterates `CollectInfo` and `AddCommunities` until there is no member of $\mathbb{R}^*(X)$ that will be preferred over $\mathrm{R}(X)$ for each X in the AS path of $\mathrm{R}(B)$.

(4) `LaunchAttack`: Finally, the adversary can simply announce the victim's prefix (instead of its own) along with the communities from step 3, and it can be assured it will have a route to the victim.

Note that the above algorithm solely employs NoExportSelect, NoExportAll, and LowerPref, but some ASes have significantly more extensive community support [16, 33]. There are cases where an adversary many want to employ a more nuanced community supported by one of its providers to achieve the same effect as a more basic community recommended by the previous algorithm. For example, some providers allow for local preference adjusting by region (as opposed to AS-wide) and export prepending (as opposed to outright suppression). These more nuanced communities may have a smaller impact on benign announcement propagation, allowing an adversary to attract more traffic with interception attacks.

We will discuss the limitations of SICO attacks in Appendix §F.

Src / Length	$R^*(X)$: Provider $R(X)$: Provider		$R^*(X)$: Peer	$R^*(X)$: Peer	$R^*(X)$: Provider
	No peering link	With peering link	$R(X)$: Peer	$R(X)$: Provider	$R(X)$: Peer
$\lvert R^*(X)\rvert = \lvert R(X)\rvert$	LowerPref	NoExport	NoExport	NoExport	—
$\lvert R^*(X)\rvert < \lvert R(X)\rvert$	LowerPref	NoExport	NoExport	NoExport	—
$\lvert R^*(X)\rvert > \lvert R(X)\rvert$	—	—	—	NoExport	—

Table 3: Comparing the preference of two routes $\mathbf{R}(X)$ and $\mathbf{R}^*(X)$ (excluding the IGP metric) at AS X and selecting which community should be used to suppress $\mathbf{R}^*(X)$. – indicates no action is needed. SICO typically aims to suppress a route by restricting exporting (so other neighbors of the AS implementing the community can still use the route). However, export restrictions often do not apply to customers. Thus, if the route does not contain a peering link, LowerPref should be used at the highest provider in the route to stop this provider from preferring the route.

4.5 Targeted interception attacks

Finally, we discuss how to use communities to achieve *targeted* interception attacks.

Let us assume the adversary wishes to attract traffic *from* a target IP within AS_{tar} that is destined to a victim's IP prefix. For targeted interception to be possible, the adversary must be capable of attracting the relevant IP traffic from AS_{tar} (AS_{tar} is in the portion of the Internet that would be affected if the adversary were to launch a hijack attack against the victim).

For each AS link X -¿ Y in $R^*(AS_{tar})$ (starting from the origin as X), the adversary should apply communities at X that prevent X from exporting $R^*(X)$ as much as possible while still allowing X to export $R^*(X)$ to Y. If X -¿ Y is a customer -¿ provider link, the adversary should use NoExportAll at X to prevent X from exporting $R^*(X)$ to peers and should use LowerLocalPref at each of X's *providers* (other than Y) to cause them to prefer the victim's route.[7] If X -¿ Y is a peering link, the adversary should use NoExportSelect at each of X's peers other than Y. In fact, X may have more peers than can realistically be enumerated without the adversary attaching too many communities (some ASes filter BGP communities if an announcement contains too many). If this is the case, the adversary should only suppress X's largest peering sessions that will carry the route the farthest. Once the adversary finds a provider -¿ customer link, it should stop adding communities and launch its attack (by announcing the victim's prefix) because this is the farthest along the route that communities will be honored.

5 EVALUATION

We performed both experimental and simulation-based evaluations of SICO attacks. Our results suggest that SICO has a minimal impact on the propagation of the adversary's announcement and is viable to a significant number of ASes throughout the Internet. We evaluate both targeted and untargeted SICO. In addition, we made our evaluation tools publicly available on GitHub [41].

5.1 Methodology

We evaluated three aspects of SICO attacks:

- *Feasibility evaluation.* We first evaluate the feasibility of SICO attacks by performing live attacks on the real-world Internet backbone. We used the PEERING testbed [78] to ethically launch attacks in the wild. The PEERING testbed operates multiple geographically distributed points of presence and allows researchers to make real-world BGP announcements to study inter-domain routing. Our experimental setup was comprised of an adversary server and a victim server. Each server was then connected to the PEERING testbed via a secure VPN so that it could make BGP announcements and forward packets through the peering points of presence (known as muxes). The victim server was connected to the PEERING mux in Northeastern University while the adversary server was connected to the PEERING muxes in Amsterdam and Seattle.
- *Measuring effect on announcement propagation.* To understand how different interception techniques affect announcement propagation, we measured the fraction of internet hosts affected by our interception attacks. Specifically, we sent probes to random samples of Internet hosts and recorded the fraction of hosts that had their responses routed to the adversary. This allows us to compare SICO to state-of-the-art techniques and quantitatively measure the propagation difference.
- *Viable AS estimation.* Finally, we used the CAIDA March 2019 AS-relationship dataset [12] to estimate the number of *viable* ASes, i.e., ASes that could be used for launching SICO attacks.

Ethical considerations. To perform these attacks in an ethical manner, we only hijacked/intercepted IP prefixes that we controlled so that no Internet traffic that was not destined to our own IP prefix was affected. We also adhered to the PEERING testbed acceptable use policy as to not overwhelm or crash routers.

5.2 Feasibility evaluation

We tested the feasibility of SICO from two different nodes (Amsterdam and Seattle) on the PEERING testbed.

[7] In addition, some ASes allow NoExportSelect to apply to providers.

Name (ASN)	By provider	By IX	By ASN
Coloclue (8283)	Yes	No	Yes
BIT (12859)	Yes	Yes	No

Table 4: Community-based export controls supported by Coloclue and BIT. While BIT did allow for export controls, Coloclue offered the ability to restrict exporting to individual peer ASNs.

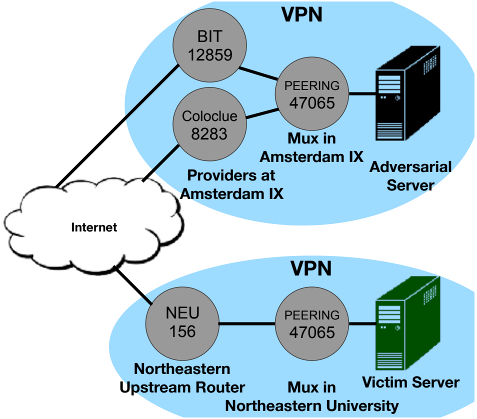

Figure 4: Experimental setup to launch BGP attacks at Amsterdam.

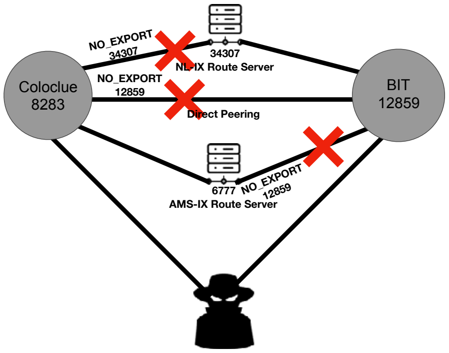

Figure 5: The configuration of the peering links between Coloclue and BIT.

Case 1: Feasibility at Amsterdam. The PEERING testbed has a mux in AMS-IX Amsterdam with two providers (Netwerkvereniging Coloclue and BIT BV), making it a logical choice to serve as an adversary in an interception attack (see Figure 4). Recall that the victim is the mux at Northeastern University. Both Coloclue and BIT support BGP communities, but Coloclue offered more fine-grained control by allowing export suppression to individual peers by ASN as opposed to grouping peers together by Internet Exchange (IX) route servers (see Table 4). In our attack, we announced the bogus route to Coloclue and used BIT for forwarding intercepted traffic to the victim.

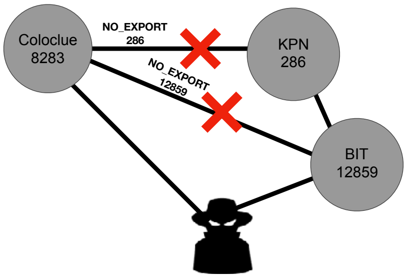

Figure 6: The providers of BIT that Coloclue had peering sessions with.

By making BGP announcements to Coloclue, we implemented the algorithm from §4.4. We performed 4 iterations of `CollectInfo` and `AddCommunities` (① — ④). We will use "[AS_1, ..., AS_n]" to denote a route and use (x:y) to denote a community, where x is an AS who should enforce the action y.

`MakeSampleAnnouncement`: We started by making a sample announcement for the adversary's own prefix to Coloclue.

① `CollectInfo`: We observed that BIT exported the path [BIT, Coloclue, Adversary] for our prefix. This implied that BIT was learning the route from Coloclue over a peering link, and we also confirmed this by looking at publicly available topology data. Further inspecting the looking glass data at Coloclue, we found Coloclue and BIT were additionally peering through the route servers at AMS-IX and NL-IX, as shown in Figure 5. The AMS-IX route server supported community controls while the NL-IX route server did not support communities.

① `AddCommunities`: We added the (Coloclue:No export to BIT) community to prevent Coloclue from exporting the malicious route to BIT. In addition, we used (AMS-IX-RS:No export to BIT) and (Coloclue:No export to NL-IX-RS) to prevent the malicious route from being exported to BIT via the route servers.

② `CollectInfo`: After applying the aforementioned communities, we observed that BIT exported the route [BIT, KPN, Coloclue, Adversary] for the adversary's prefix (see Figure 6). This was problematic because the route from BIT to the victim's prefix was [BIT, KPN, Cogent, Northeastern, Victim], and KPN would prefer the adversary's route through Coloclue over the victim's route through Cogent because of the shorter AS path (both Coloclue and Cogent are peers of KPN with equal local preference) and BIT would no longer hear its route to the victim.

② `AddCommunities`: We added (Coloclue:No export to KPN), which successfully stopped BIT from exporting the route [BIT, KPN, Coloclue, Adversary].

③ `CollectInfo`: BIT now exported the route [BIT, NTT, Atom86, Coloclue, Adversary] for the adversary's prefix. Notice that for BIT, this route is provider learned with an equal AS path length as its route to the victim [BIT, KPN, Cogent, Northeastern, Victim]. Thus, announcing the

Community	Target AS	Action	Reason
0:12859	AMS-IX Route Server	No export to BIT	Prevent peering routes between Coloclue and BIT via the AMS-IX route server
2914:4211	NTT	Prepend 1x to all customers in Europe	Lengthen the adversary's route through NTT so that it is longer than the victim's route through KPN
8283:4:12859	Coloclue	No export to BIT	prevent the direct peering between Coloclue and BIT
8283:4:34307	Coloclue	No export NL-IX route server	Prevent peering routes between Coloclue and BIT via the NL-IX route server
8283:4:286	Coloclue	No export KPN	Allow KPN to prefer its route to the victim and export it to BIT

Table 5: Communities used to achieve interception at Amsterdam. Note that the community string for a given action varies across providers, and we show the exact community strings to facilitate reproducing experiments.

victim's route at this point would be a gamble since BIT could either export the adversary's route or the victim's route based on the IGP metric (or a further tie-break condition).

③ `AddCommunities`: The route [BIT, NTT, Atom86, Coloclue, Adversary] has no peering links in it (Atom86 is a provider to Coloclue and NTT is a provider to Atom86 and BIT), so the conservative way to stop BIT from exporting this route would be to lower the local preference of the adversary's route at NTT. One approach was to regionally lower the local preference of the adversary's route in the locations where NTT had BGP sessions with BIT. This had potential, but BIT's sessions with NTT were located in Amsterdam which is a hub for Internet connectivity. Thus, lowering local preference in Amsterdam would cause NTT not to export the adversary's route across a large number of other BGP sessions that it had in Amsterdam, which may have a major impact on announcement propagation. The second approach was to use prepending to simply make the route through NTT longer, so that BIT would prefer the route through KPN. NTT actually provides a community for performing prepending on the routes that will be announced to its customer ASes in Europe. We ended up using this community to have a minimal impact on propagation.

④ `CollectInfo`: The communities we added caused BIT to export the route [BIT, NTT, NTT, Atom86, Coloclue, Adversary] for the adversary's prefix and [BIT, KPN, Cogent, Northeastern, Victim] for the victim's prefix. Here, BIT learns both routes through providers but the victim's route is one hop shorter. Thus, we were confident that BIT would choose the victim's route.

`LaunchAttack`: Using the adversary mux in Amsterdam, we announced the victim's prefix with the appropriate communities (the exact values of the communities we used are in Table 5). BIT still exported the route [BIT, KPN, Cogent, Northeastern, Victim] and allowed us to forward traffic to the victim.

Thus, through the algorithm in §4.4, we were able to launch an interception attack at Amsterdam by strategically limiting announcement propagation with communities.

Case 2: Feasibility at Seattle. We studied the applicability of these attacks from the PEERING mux in Seattle (with the victim at the PEERING node at Northeastern University). The PEERING mux in Seattle only has one provider (RGNet) so there is no way to forward traffic directly out of Seattle. Therefore, we used a VPN tunnel to the PEERING mux in Amsterdam so that the adversary could attract and forward traffic (see Figure 7). While we used the mux in Amsterdam previously to make announcements and forward traffic, here we only used it as a means of forwarding traffic because the mux in Seattle did not have a second provider. When the adversary made its announcement from Seattle, the providers of the mux in Amsterdam all preferred the adversary's announcement and did not have a valid route to the victim. To overcome this we used SICO. Also, we noticed that the provider to the PEERING mux in Seattle (RGNet) did not support any community actions but it did transit communities up to higher-up providers that did. The details how we achieved interception in Seattle are presented in Appendix §D.

Overall, we successfully employed the algorithm from §4.4 demonstrating the viability of SICO in a setting where the direct provider did not support communities and instead only forwarded communities.

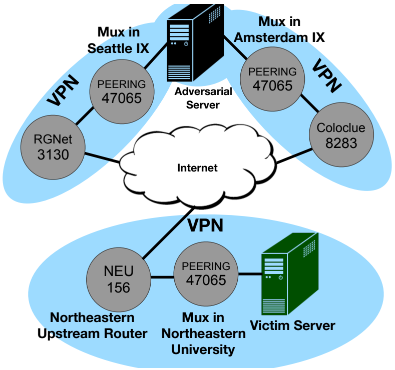

Figure 7: Experimental setup to launch BGP attacks at Seattle and forward traffic through Coloclue.

5.3 Quantifying announcement propagation through spread

To quantify how much of the Internet is affected by malicious announcements with different interception techniques, we measure their *spread* which represents the percentage of Internet hosts that use a given route. Spread loss measures the amount of spread that is given up (relative to the theoretical maximum spread for an Interception attack) when a particular interception technique is implemented.

Spread measurement methodology. We operated two servers, one to pose as the victim and one to pose as the adversary. We first made an unmodified BGP announcement from the victim's server, and then launched a given BGP interception attack (AS-path poisoning, selective neighbor announcement, or SICO) from the adversary's server. Given a set of active hosts (IP addresses), we sent a probe to each host from the victim's server, and measured the *attack spread* (S_{atk}), which is defined as the fraction of hosts that had their responses routed to the adversary. See Figure 8 for an example. As a baseline, we also measured the attack spread when the adversary simply announces the victim's prefix to one provider (without communities or AS-path poisoning), and call such spread *baseline spread* (S_{base}). One may not be able to use this type of announcement to achieve interception, but we can use it to measure the maximum spread achievable by an interception attack. Then, we use *spread loss*, S_{loss}, to measure the efficiency of an attack, where

$$S_{loss} = (S_{base} - S_{atk})/S_{base}$$

An attack with a lower spread loss can intercept more traffic and is considered to be more effective.

We are particularly interested in the active hosts that are running ICMP and HTTPS. ICMP is a very widely used protocol and is often enabled on end hosts and home routers, which can be used as a rough estimation of the distribution of end hosts. Studying the effect of interception on HTTPS hosts is important as interception attacks against HTTPS sites could cause devastating results [47]. For each protocol, we constructed a random sample of 1,000 hosts that supported that protocol. These hosts served as the target hosts for our measurements. For ICMP ping hosts we used an ICMP ping request as the probe and for HTTPS hosts we used a TCP SYN to port 443.

For HTTPS, we queried a list of 15,000 random hosts from the Censys Internet-wide scans [54] (using ORDER BY RANDOM() in Google Big Query SQL) that had port 443 open and were serving browser-trusted certificates. We then filtered this sample by recording only the hosts that actively responded to our own TCP SYN packets sent to port 443. Finally we limited the sample size to 1,000 as to not overwhelm the PEERING testbed. To validate that our filtering did not tamper with randomness of the sample from the Censys database, we performed a chai-squared analysis presented in Appendix §E.

For ICMP ping we started with a list of 15,000 collected with no selection criteria (since the Censys data definition did not include ping connectivity) using ORDER BY RANDOM() in Google Big Query SQL and then filtered this sample by recording which hosts actively responded to our pings. Finally, we limited the sample to 1,000 hosts. We did not perform a chai-squared analysis for reasons discussed in Appendix §E.

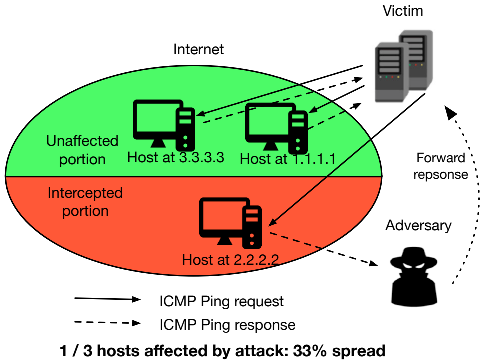

Figure 8: Hosts that are in the intercepted portion of the Internet will send responses to the adversary while hosts in the portion of the Internet unaffected by the adversary's attack will send responses directly to the victim.

Measuring the spread loss of SICO. We measured the spread loss of SICO and found that SICO only reduces average spread (i.e., SYN and Ping spread averaged) by 0.1% ($S_{atk} = 68.8\%$, $S_{base} = 68.9\%$) when implemented at the Amsterdam mux and 11.4% ($S_{atk} = 38.9\%$, $S_{base} = 43.9\%$) when implemented in Seattle.

5.4 Comparison with state-of-the-art

We compared SICO to selective neighbor announcement and AS-path poisoning. We found that selective neighbor announcement was incapable of launching interception attacks and AS-path poisoning causing a greatly increased spread loss when compared to SICO.

Comparison with selective neighbor announcement. When launching interception attacks against Northeastern University from both Amsterdam and Seattle, *selective neighbor announcement was NOT viable* because, whichever upstream the adversary announced to, the adversary's announcement prevented the other provider from having a route to the victim.

Comparison with AS-path poisoning. We experimented with AS-path poisoning by prepending the ASNs of the adversary's (one to three) upstream ASes that were used for forwarding intercepted traffic in the adversary's announcement. We reused the setups from the feasibility measurements (Figure 4 and Figure 7). In both setups, AS-path poisoning prevented the adversary's upstream from importing the adversary's announcement and gave the adversary a route to the victim. However, due to the longer AS path, AS-path poisoning also caused a significant reduction in the spread

Setting	SICO	Poisoning 1 AS	Poisoning 2 ASes	Poisoning 3 ASes
Amsterdam SYN	0%	22%	82%	85%
Amsterdam Ping	0%	25%	79%	83%
Amsterdam Avg.	0%	24%	81%	84%
Seattle SYN	9%	70%	99%	100%
Seattle Ping	14%	73%	98%	100%
Seattle Avg.	11%	72%	99%	100%

Table 6: Spread losses under different settings (rounded to the nearest percent). Evaluated with Coloclue at Amsterdam and RGnet at Seattle as upstreams ("Avg" is SYN and Ping spread losses averaged).

of the adversary's announcement, reducing the amount of Internet traffic the adversary could collect.

In Table 6 we show the spread losses of SICO and AS-path poisoning under different settings. SICO outperformed AS-path poisoning by a factor of over **100x** at Amsterdam and over **6x** at Seattle. Specifically, SICO only reduces average spread by 0.1% when implemented at the Amsterdam mux and 11.4% when implemented in Seattle. In the *optimum* case for AS path poisoning (i.e., prepending only a single ASN), the spreads were reduced by 23.7% ($S_{atk} = 52.6\%$, $S_{base} = 68.9\%$) and 71.8% ($S_{atk} = 12.4\%$, $S_{base} = 43.9\%$) on average at Amsterdam and Seattle, respectively. Even in this optimum case, SICO has a 16.2 greater absolute spread than AS-path poisoning at Amsterdam and a 26.5 greater absolute spread at Seattle. Considering the poisoning of additional ASNs (which may be necessary in certain topologies), the spread losses became much higher (even near 100%), and SICO outperforms AS-path poisoning by an absolute spread of 57.8% at Amsterdam and 38.8% at Seattle.

The dramatic decrease in propagation caused by poisoning more than one AS is likely due to prefix filtering practices at major providers, which filter announcements coming from a customer containing the ASN of a peer [79]. [8] In both cases we studied, poisoning two or more ASes required poisoning the ASNs of major transit providers that triggered prefix filtering at other transit providers (which would otherwise carry our route). The dilemma of AS-path poisoning triggering prefix filters is inherent to the technique and is a major drawback of AS-path poisoning. While in our setup this problem was only encountered when poisoning two or more ASes, if an adversary has a large provider (e.g., a tier-1 provider) that it wants to use to forward traffic, even poisoning the single ASN of its immediate provider could trigger prefix filtering and make AS-path poisoning not viable. This highlights one of the fundamental benefits of community-based interception over previous techniques: *SICO leaves the AS-path unchanged, which bypasses AS-path-based filtering and other AS-path related detection techniques.*

5.5 Viable AS estimation

To launch a SICO attack, an AS must meet two conditions:

(1) It is multi-homed so it has a provider to forward traffic and a provider to receive traffic (a requirement of all announcement-shaping based interception attacks).

(2) It can use communities to influence the behavior of one of its direct providers or indirect providers (a provider's provider).

SICO viability without considering community forwarding. Based on the CAIDA March 2019 topology [12], 59% of all ASes are multi-homed and thus satisfy condition 1, making them potentially capable of interception. However, the second condition pertaining to community support is more difficult to measure directly since there is no centralized database of AS community support. To overcome this we do not attempt to model community support across the entire AS graph. Instead, we only model community support by ASes that we know support the relevant communities via manual inspection of their routing policy (i.e., ASes that have a Yes in all three columns of Table 10). As a conservative metric, we counted an AS as being capable of launching SICO attacks only it was multi-homed and had a direct provider (listed in the CAIDA topology) supporting the required communities. 24% of ASes (or 41% of multi-homed ASes) satisfied this condition giving us a lower bound of 24% on attack viability.

SICO viability considering community forwarding. However, directly having a provider that supports communities is not the only way to achieve community controls. Many providers forward communities on and may have providers above them that support communities. To model this, we collected three months of Route Views project (from May 2019 – July 2019) data [44] and referred to this dataset as *RV dataset*. We extracted 176 million BGP updates that contain communities from the RV dataset, and recorded ASes that were seen forwarding communities in a manner similar to Streibelt *et al.* [81]. Specifically, if we observed the AS path:

$$AS1, AS2, AS3, AS4$$

with the community AS 4:101, where AS 4 is the prefix's origin, we can assume this community was attached by AS 4.[9] Then we can record AS 2 and AS 3 as forwarding communities. We do not consider this evidence that AS 1 forwards communities because AS 1 is the Route Views peer. Route Views peers may use a different configuration for their peering session with the Route Views collector than for their other BGP sessions. Overall we recorded 3.5 K ASes as forwarding communities.

[8] To confirm this, we repeated the longer AS-path poisoning experiments, but instead of using the first two and three ASNs in the path, we simply poisoned the adversary's immediate upstream two and three times. With these announcements we noticed significantly larger spread which indicates that the specific ASNs that were poisoned caused our announcements to be filtered and were responsible for the reduction in spread (as opposed to only the path length).

[9] There is no reason for another AS in this path to attach this community because this update will never pass through AS 4 again.

With this information, we counted how many ASes had either a provider that supported communities or a chain of providers (all of which forwarded communities) that eventually lead to a provider that supported communities. While this implies that an adversary can propagate communities to an AS that will support them, it is worth noting that sometimes, to launch a SICO attack, community controls are needed at a lower-tier provider to suppress route propagation over peering links lower in the Internet hierarchy. Thus, sometimes routing decisions must be influenced at providers that only forward communities. The most common case of this is when an AS has a provider that forwards communities but also peers with all of its other providers. Here, even if the adversary uses community controls to influence routing at higher-up ASes, the adversary cannot suppress route propagation over the relevant peering links between its providers. To exclude this case, we did not count an AS as begin capable of launching SICO attacks if its providers that forwarded communities peered with all of its other providers.

Using this analysis technique, *we estimate that SICO is viable to 48% of ASes* (or 83% of multi-homed ASes).

5.6 Targeted interception attacks

To measure the effectiveness of targeted interception attacks, we used the PEERING mux in Amsterdam as an adversary and the PEERING mux at Northeastern University as a victim. We then generated a list of sample targets to study. We chose the top 9 Tor nodes by bandwidth in the February 15, 2019 Tor consensus (the official document containing all Tor nodes bandwidths) as sample targets. Traffic from these nodes to top websites would hypothetically be the target of a BGP attack to deanonymize Tor users, as shown by Sun et al. [83].

For each node we engineered a targeted BGP attack (against a victim prefix we controlled) that affected as little of the Internet as possible while still including the IP address of the target node. We then confirmed that the node was affected by our attack by sending a TCP SYN packet to a known open port listed in the Tor consensus, and listening for the responding SYN+ACK packet. Once we confirmed our attack affected the node we were targeting, we took a spread measurement to observe the fraction of other Internet hosts that were affected by the attack.

Of the 9 nodes we studied, 1 node routed traffic to the victim even when the adversary launched a BGP attack with the maximum possible spread. Given the Internet topology, the maximum spread of the adversary was 73%, so it was not unexpected that of the 9 nodes some of them would be beyond the adversary's reach. On the remaining 8 nodes, the average attack spread was only 2.7% meaning that, on average, 97.3% of the Internet hosts were oblivious to our attacks. See Table 7 for more details.

We found that, on average, launching a targeted attack reduces the traffic load the adversary must handle by a factor of **25x** since the adversary must only route traffic to the victim's prefix from 2.7% of the Internet as opposed to 68.8% (the spread of an untargeted BGP interception attack against this prefix from the PEERING node at Amsterdam). In addition, targeting a BGP interception attack reduces the overall (Internet-wide) effect on latency to the victim's prefix since a larger portion of Internet traffic still uses a direct route to the victim and does not have to be additionally routed through the adversary.

Tor node IP	**Tor node ASN**	**Spread SYN**	**Spread Ping**	**Spread Avg.**
46.165.245.154	28753	5.3	5.6	5.5
94.23.150.81	16276	1.6	2.1	1.8
31.220.0.225	206264	0.1	0.2	0.2
62.210.177.181	12876	0.3	0.6	0.5
199.249.230.72	62744	2.0	4.6	3.3
178.32.181.96	16276	1.6	2.1	1.9
195.206.105.217	9009	8.8	7.1	8.0
176.9.44.232	24940	0.7	0.8	0.7
Average	NA	2.6	2.9	2.7

Table 7: Results of BGP attacks targeting Tor nodes.

Importantly, while AS-path poisoning causes an *indiscriminate* reduction in the propagation of an attack, BGP communities can be used to *strategically* limit unwanted propagation beyond the target (or targets) an attack is designed to affect. When AS-path poisoning is used, which parts of the Internet no longer prefer the adversary's announcement because of the longer AS path is beyond the control of the adversary. This is distinctly different from the targeted attacks, where an adversary can choose which sections of the Internet no longer prefer its route while allowing its target source IP to still prefer its announcement.

5.7 Limitations in Evaluation

We were limited by only being able to launch attacks from the nodes of the PEERING testbed as opposed to randomly selected ASes, and we acknowledge that these nodes are not necessarily representative of the Internet as a whole. However, we performed analysis of the Internet topology to find that there are a significant number (roughly 48%) of ASes that are in situations similar to the cases we evaluated with the PEERING testbed, i.e., have providers with a comparable level of community support or forward communities to providers that do.

6 DISCUSSION: COUNTERMEASURES

6.1 Existing solutions

We analyze SICO and alternative interception techniques under several deployed solutions to routing security.

Prefix filtering. Prefix filtering can eliminate BGP hijack and interception attacks by preventing adversaries from announcing prefixes that are not allocated to them [79] (best-practices for prefix filtering are well outlined by the Internet society's MANRS project [20]). In the case of a stub network, *prefix filtering is theoretically effective against all methods of achieving interception.* However, non-stub networks may still be able to launch BGP hijacks and interceptions if they are targeting an AS that is a direct or indirect customer of theirs. This is viable because customer prefixes are allowed through prefix filters. In addition, many ASes still do not properly implement prefix filtering as evident by the continual stream of new BGP attacks [49, 60, 73] and the widespread propagation of a recent route leak that could have been stopped by prefix filtering through a major US carrier [70].

Route origin validation. Route origin validation involves filtering BGP announcements by origin AS and IP prefix to only allow announcements for an IP prefix by the legitimate prefix owner. This is most commonly done by using RPKI to generate a list of origin ASes allowed to announce specific prefixes and then filtering based on this list (a.k.a. RPKI ROV) [50]. *Route origin validation does not prevent SICO or other interception techniques* because the adversary can simply prepend the legitimate prefix owner's ASN to its announcement which fools the route origin check. However, it does have the beneficial aspect of making the adversary's announcement one hop longer thus lowering its route ranking and reducing its route's spread.

AS path filtering. AS path filtering (like peer locking [79]) filters *all* members of an AS path to prevent an AS from accepting routes that contain suspicious ASNs or ASN combinations in the AS path. A simple conservative application of this involves filtering routes coming from customers that contain the ASN of a tier-1 transit provider, but more extensive configurations also exist [79]. *AS path filtering is highly effective against AS-path poisoning but completely ineffective against SICO* because AS-path poisoning uses AS-path manipulation to control update propagation and may require an adversary to put ASNs in its AS path that will ultimately lead to its announcement being filtered. However, SICO leaves the AS path unmodified (unless an adversary intentionally chooses to modify it to evade route origin validation) since propagation control is instead achieved with communities. This prevents SICO from triggering AS path filtering.

Ultimately, a cryptographic solution like BGPsec offers the most comprehensive resolution to the problem of BGP interception and hijack attacks. We hope that our work serves to motivate and accelerate the adoption of comprehensive security mechanisms such as BGPsec.

6.2 Potential Countermeasures

We investigate four potential countermeasures: (1) restricting community propagation, (2) restricting the number of communities in a BGP update, (3) only allowing certain community actions, and (4) using historical BGP updates to detect abnormal communities (i.e., communities that are not normally seen). We find that although they can mitigate SICO attacks, they all affect legitimate BGP community usage to some extent. Ultimately network operators must consider a tradeoff between allowing full community use while enabling SICO attacks, or restricting community use (including some legitimate usage) to mitigate the effects of SICO attacks.

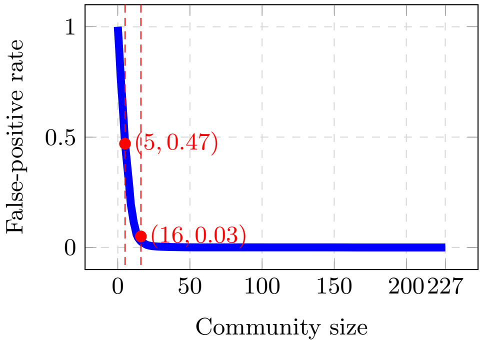

Figure 9: False-positive rate vs. community size.

Restricting community propagation. Streibelt *et al.* recommend in an extreme case that ASes only propagate communities to the immediate peer the communities are targeting, i.e., community propagation would be limited to only 2 AS hops [81]. This can weaken an adversary's ability to launch attacks since it cannot influence routing at an AS that is more than two hops away. However, this proposal, similar to the countermeasure that simply disable or reduce support for BGP communities, may limit some of the legitimate uses of BGP communities for traffic engineering purposes.

We used the BGP updates collected from RouteView (i.e., the RV dataset) (see §5.5) to inspect the number of hops a community can propagate and investigate the impact of limiting community propagation. If we observe an update, whose AS path is $AS1, AS2, ...ASk$ and is associated with communities from ASk, we assume the communities from ASk can propagate $k-1$ hops. We found communities can propagate up to 14 hops in our dataset, and restricting the community propagation to 1 hop can cause 32.0% (i.e., false-positive rate) of updates to be dropped , which can affect the updates from 4,217 ASes. If one restricts the community propagation to 2, 3, 4, and 5 hops, 9.7%, 2.5%, 0.6%, and 0.1% of updates will be affected, corresponding to 4,003, 3,657, 2,943, and 1,902 ASes, respectively. AS we can see, even a small false-positive rate (e.g., 0.1%) can affect a considerable number of ASes.

Restricting community size. One potential countermeasure is restricting the number of communities in an announcement since in SICO attacks an attacker may need to attach a lot of communities to the announcements. We define the number of unique communities being attached to an update as *community size*, and examine the false-positive rates (i.e., the fractions of updates that are being incorrectly blocked under various community size restrictions) of this countermeasure. As shown in Figure 9, the community sizes of legitimate BGP updates can be quite large (up to 227). In fact, in our

	Action-only	Number-only	Action + Number
1/2	1,064 (19.6)	200 (3.7)	186 (3.4)
1/3	1,295 (23.9)	277 (5.1)	269 (5.0)
1+2/3	887 (16.4)	162 (3.0)	152 (2.8)

Table 8: False positives of historical-update-based anomaly detection. $a(+b)/c$ indicates using the updates from the a^{th} (or a^{th} and b^{th}) month to build a model and examine the updates from the c^{th} month. “Action-only”, “Number-only”, and “Action + Number” show the numbers of ASes fail the community action check, the community number check, or both checks, respectively. False-positive rates are in the parenthesis.

experiments we only need at most 5 communities for non-targeted attacks and 16 communities for targeted attacks. The community sizes of more than 47% (6,192 ASes) and 3% (4,399 ASes) of the updates are longer than 5 and 16, respectively.

Restricting community action. Another potential countermeasure is to further limit the actions communities can perform. However, the actions that enable interception attacks are very similar to the legitimate actions a network operator would want to have access to for traffic-engineering purposes. Fundamentally, traffic-engineering involves shaping BGP announcements to optimize cost or quality of service. These same communities that allow for this type of announcement shaping (e.g., local preference adjusting, announcement suppression) let an adversary shape announcements to enable interception.

Anomaly detection based on historical updates. We further examine the efficiency of using historical BGP updates to detect “abnormal” updates. For a given AS, we use its historical updates to model its updates, i.e., observing the set of common community actions and the maximum number of communities sent by the AS in the updates, and then examine if the community actions and community numbers in its future updates are consistent with the built model. We used the 5,416 ASes that appear in all the three months of updates in the RV dataset as our target ASes, and consider three settings: using the first month (May 2019) of updates as the historical updates to examine the remaining two months (June 2019 and July 2019) of updates, and using the first two months of updates as the historical updates to examine the third month of updates. An AS is a false positive if it fails the community action check (i.e., some of its updates contain unseen community actions) or the community number check (i.e., the number of communities sent in some updates exceeds the maximum number seen from the historical updates.). The results are shown in Table 8. Even if we require that a false positive should fail BOTH checks, this countermeasure still affects a considerable number of ASes (152 or 2.8% of the target ASes) in the best-case scenario.

As the above examples indicate, even a basic level of community support amplifies the effectiveness of BGP attacks by enabling interception, and this undesired effect is hard to remove without stripping BGP communities of one of their primary uses. A possible method to overcoming this challenge would be to couple support for BGP action communities with AS reputation mechanisms [66]. This way, more reputable ASes could leverage the advantages of communities while potential attackers would not be able to use them to facilitate attacks.

7 RELATED WORK

BGP interception attacks. Ballani *et al.* [46] performed an in-depth study of BGP interception but only considered announcing to select neighbors as a way of enabling interception. Goldberg *et al.* [59] consider a clever combination of AS-path poisoning and selective neighbor announcement in the context of various BGP security proposals (like soBGP and S-BGP [63, 84]) but still cannot overcome the fundamental challenges of these techniques (i.e., difficulty maintaining a route to the victim and limited announcement propagation). Thus, while interception achieved with the method presented by Goldberg *et al.* is “difficult for stubs” [59], SICO attacks are highly effective even in the case of stub networks. Pilosov and Kapela [75] looked into interception via AS-path poisoning on a sub-prefix announcement. While this attack elegantly performs internet-wide interception, it has several disadvantages compared to the attacks outlined in this paper. It is more difficult to target (the attack is inherently global since it is a sub-prefix attack), it is more noticeable to BGP monitoring, and it is not viable against /24 prefixes (since /25s are often filtered).

Studying BGP communities. Streibelt *et al.* performed innovative work studying attacks enabled by BGP communities and the BGP community ecosystem [81]. They highlighted how the ability of communities to influence route propagation at remote ASes can be exploited by adversaries to manipulate Internet routing. However, they did not study interception attacks or the targeting of attacks to different portions of the internet. Donnet *et al.* present early work showing a taxonomy of BGP communities [53]. For our work we augment this taxonomy by taking a more fine-grained look at where communities are accepted and propagated, as well as going more in depth into the communities used for peer export suppression.

In addition, there is a large body of recent work that highlights the lack of coherent design and standardization of BGP communities. Giotsas et al. examined communities that geographically tag route origins and found that there were no standardized values across providers [57]. In addition, even though RFC 7999 standardizes the black hole community [64], Giotsas et al. found that several nonstandard variants still exist and some ASes do not adhere to the proper implementation of the standard (particularly regarding the propagation of blackholed prefixes) [58]. The severe lack of standardization and centralized documentation for BGP communities has caused researchers to resort to applying natural language processing on routing policies as a means of measuring large scale community usage [57, 58]. We considered this approach but instead opted to manually parse routing policies from a smaller number of ASes to eliminate potential inaccuracies and extract more nuanced levels of community support.

Defenses against BGP attacks. The defenses outlined in §6.2 represent only a small portion of the potential countermeasures to BGP attacks. Lad *et al.* introduced the early monitoring system that detected route origin changes [68]. RPKI takes a proactive approach to validation origins by having ASes participate in ROV to restrict the propagation of BGP attacks [50]. However, origin authentication is only effective to an extent given that an adversary can prepend the required ASNs to evade defenses that only consider route origins [56]. BGPsec offers a more comprehensive cryptographic solution to BGP attacks [69] but currently has seen little deployment and offers only marginal benefit at low adoption percentages [72]. Clean-slate approaches like the SCION architecture [85] offer alternatives to BGP for inter-domain routing, but once again deployment rates are currently relatively low.

8 CONCLUSION

We present novel community-based BGP interception attacks that can strategically target small portions of the Internet. We then evaluate the feasibility of these attacks in the wild and measure their effectiveness to find that our attacks are *significantly* more effective then the state-of-the-art. We also successfully launched targeted interception attacks that were isolated to only 2.7% of the Internet on average. Through Internet topology analysis we found that, at a minimum, 83% of multi-homed ASes are capable of launching interception attacks via BGP communities. Overall, our work is the first work to use BGP communities to enable interception attacks and the first work to propose *targeted* interception attacks that are aimed at specific source IP addresses.

ACKNOWLEDGMENTS

We would like to thank the CCS reviewers for their helpful feedback. In addition we are grateful for support from the National Science Foundation under grant CNS-1553437 and the Open Technology Fund through their Securing Domain Validation project. The paper is also based upon work supported by DARPA and AFRL under Contract FA8750-19-C-0079.

REFERENCES

[1] 2019. Administrative How To's. https://www.he.net/adm/
[2] 2019. AS209. https://bgp.he.net/irr/as-set/AS-AS209
[3] 2019. AS2828: One Step. https://onestep.net/communities/as2828/
[4] 2019. AS286 Communities. https://as286.net/AS286-communities.html
[5] 2019. AS286 Routing Policy. https://as286.net/AS286-routing-policy.html
[6] 2019. AS3320: One Step. https://onestep.net/communities/as3320/
[7] 2019. AS701: One Step. https://onestep.net/communities/as701/
[8] 2019. AS7018: One Step. https://onestep.net/communities/as7018/
[9] 2019. AS7922: One Step. https://onestep.net/communities/as7922/
[10] 2019. [AusNOG] Telstra BGP Communities. http://lists.ausnog.net/pipermail/ausnog/2014-January/022314.html
[11] 2019. BGP Community support for AS6762 Customers. https://etabeta.noc.seabone.net/communities.html
[12] 2019. The CAIDA AS Relationships Dataset March 2019. http://www.caida.org/data/as-relationships/
[13] 2019. Cogent Customer User Guide. http://cogentco.com/files/docs/customer_service/guide/global_cogent_customer_user_guide.pdf
[14] 2019. Customer tagable Community Attribute Values. http://www.cw.net/incommunities.shtml
[15] 2019. GBLX Customer BGP Communities. https://onestep.net/communities/as3549/
[16] 2019. GTT BGP Communities. https://www.gtt.net/us-en/services/internet/ip-transit/bgp-communities/
[17] 2019. Internet Exchange Directory. https://www.pch.net/ixp/dir#!mt-sort=avg,desc!mt-pivot=avg
[18] 2019. IX Route-Servers. https://www.franceix.net/en/technical/france-ix-route-servers/
[19] 2019. IX Route Servers — AMS-IX Amsterdam. https://www.ams-ix.net/ams/documentation/ams-ix-route-servers
[20] 2019. MANRS Project Homepage. https://www.manrs.org/
[21] 2019. NTT looking glass. http://www.us.ntt.net/support/looking-glass/
[22] 2019. Object name: AS12389, ROSTELECOM-AS. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS12389
[23] 2019. Object name: AS1299, Telia. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS1299
[24] 2019. Object name: AS20485, TRANSTELECOM. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS20485
[25] 2019. Object name: AS31133, MF-MGSM-AS. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS31133
[26] 2019. Object name: AS3216, SOVAM-AS. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS3216
[27] 2019. Object name: AS3356, Level3. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS3356
[28] 2019. Object name: AS3491, PCCW Global. $whois-hwhois.ripe.netAS3491
[29] 2019. Object name: AS5511, Opentransit. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS5511
[30] 2019. Object name: AS8359, MTS. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS8359
[31] 2019. Object name: AS9002, RETN-AS. https://apps.db.ripe.net/db-web-ui/#/query?searchtext=AS9002
[32] 2019. Operational BGP Communities. https://www.de-cix.net/en/resources/route-server-guides/operational-bgp-communities
[33] 2019. Policies & Procedures - Routing Policies - NTT America. http://www.us.ntt.net/support/policy/routing.cfm
[34] 2019. RASCOM GiGANET community list. https://noc.rascom.ru/communities_en.html
[35] 2019. Report on RPKI Invalid Prefixes (Archived version: https://perma.cc/286U-4WA6). https://as286.net/data/ana-invalids.txt
[36] 2019. Route Policy. https://www.hkix.net/hkix/route-policy.htm
[37] 2019. Route Server and Communities. https://ix.br/route-server-and-commmunities
[38] 2019. Route Servers. https://www.seattleix.net/route-servers
[39] 2019. Route servers. https://www.netnod.se/ix/route-servers
[40] 2019. Route Servers: LINX Portal. https://portal.linx.net/tech-info-help/route-servers
[41] 2019. SICO-tools. https://github.com/inspire-group/SICO-tools
[42] 2019. State of IP Spoofing. https://spoofer.caida.org/summary.php
[43] 2019. TATA AS6453 BGP Communities. https://www.scribd.com/document/399871041/TATA-AS6453-BGP-Communities
[44] 2019. University of Oregon Route Views Project. http://www.routeviews.org/routeviews/
[45] 2019. Zayo BGP communities. https://onestep.net/communities/as6461/
[46] Hitesh Ballani, Paul Francis, and Xinyang Zhang. 2007. A Study of Prefix Hijacking and Interception in the Internet. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '07)*. ACM, New York, NY, USA, 265–276. https:

//doi.org/10.1145/1282380.1282411

[47] Henry Birge-Lee, Yixin Sun, Anne Edmundson, Jennifer Rexford, and Prateek Mittal. 2018. Bamboozling Certificate Authorities with BGP. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, Baltimore, MD, 833–849. https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee

[48] Jay Borkenhagen. [n.d.]. NANOG Mailing List: AT&T/as7018 now drops invalid prefixes from peers. https://mailman.nanog.org/pipermail/nanog/2019-February/099501.html

[49] Russell Brandom. 2018. Hackers emptied Ethereum wallets by breaking the basic infrastructure of the internet. https://www.theverge.com/2018/4/24/17275982/myetherwallet-hack-bgp-dns-hijacking-stolen-ethereum

[50] R. Bush and R. Austein. 2013. *The Resource Public Key Infrastructure (RPKI) to Router Protocol*. RFC 6810. RFC Editor.

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

[52] Christoph Dietzel, Matthias Wichtlhuber, Georgios Smaragdakis, and Anja Feldmann. 2018. Stellar: Network Attack Mitigation Using Advanced Blackholing. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '18)*. ACM, New York, NY, USA, 152–164. https://doi.org/10.1145/3281411.3281413

[53] Benoit Donnet and Olivier Bonaventure. 2008. On BGP Communities. *SIGCOMM Comput. Commun. Rev.* 38, 2 (March 2008), 55–59. https://doi.org/10.1145/1355734.1355743

[54] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *ACM Conference on Computer and Communications Security*.

[55] L. Gao and J. Rexford. 2001. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking* 9, 6 (Dec 2001), 681–692. https://doi.org/10.1109/90.974523

[56] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. 2017. Are We There Yet? On RPKI's Deployment and Security. In *Network and Distributed Systems Security Symposium (NDSS)*.

[57] V. Giotsas, C. Dietzel, G. Smaragdakis, A. Feldmann, A. Berger, and E. Aben. 2017. Detecting Peering Infrastructure Outages in the Wild. In *ACM SIGCOMM*.

[58] V. Giotsas, G. Smaragdakis, C. Dietzel, P. Richter, A. Feldmann, and A. Berger. 2017. Inferring BGP Blackholing Activity in the Internet. In *Internet Measurement Conference (IMC)*.

[59] Sharon Goldberg, Michael Schapira, Peter Hummon, and Jennifer Rexford. 2010. How Secure Are Secure Interdomain Routing Protocols. In *Proceedings of the ACM SIGCOMM 2010 Conference (SIGCOMM '10)*. ACM, New York, NY, USA, 87–98. https://doi.org/10.1145/1851182.1851195

[60] Dan Goodin. 2017. Russian-controlled telecom hijacks financial services' Internet traffic. https://arstechnica.com/information-technology/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/

[61] G. Huston. 2004. *NOPEER Community for Border Gateway Protocol (BGP) Route Scope Control*. RFC 3765. RFC Editor.

[62] Team Cymru Inc. [n.d.]. The Bogon Reference. https://www.team-cymru.com/bogon-reference.html

[63] S. Kent, C. Lynn, and K. Seo. 2000. Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications* 18, 4 (April 2000), 582–592. https://doi.org/10.1109/49.839934

[64] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins. 2016. *BLACKHOLE Community*. RFC 7999. RFC Editor.

[65] Jac Kloots. 2014. RPKI Routing Policy Decision-Making - a SURFnet Perspective. https://labs.ripe.net/Members/jac_kloots/rpki-routing-policy-decision-making-a-surfnet-perspective

[66] Maria Konte, Roberto Perdisci, and Nick Feamster. 2015. ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. ACM, New York, NY, USA, 625–638. https://doi.org/10.1145/2785956.2787494

[67] W. Kumari and D. McPherson. 2009. *Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)*. RFC 5635. RFC Editor.

[68] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. 2006. PHAS: A Prefix Hijack Alert System.. In *USENIX Security Symposium*, Vol. 1. 3.

[69] M. Lepinski and K. Sriram. 2017. *BGPsec Protocol Specification*. RFC 8205. RFC Editor.

[70] Martin J Levy. 2019. The deep-dive into how Verizon and a BGP Optimizer Knocked Large Parts of the Internet Offline Monday. https://blog.cloudflare.com/the-deep-dive-into-how-verizon-and-a-bgp-optimizer-knocked-large-parts-of-the-internet-offline-monday/

[71] Kirk Lougheed and Jacob Rekhter. 1989. *Border Gateway Protocol (BGP)*. RFC 1105. RFC Editor. http://www.rfc-editor.org/rfc/rfc1105.txt http://www.rfc-editor.org/rfc/rfc1105.txt.

[72] Robert Lychev, Sharon Goldberg, and Michael Schapira. 2013. BGP Security in Partial Deployment: Is the Juice Worth the Squeeze?. In *ACM SIGCOMM*. New York, NY, USA, 171–182. https://doi.org/10.1145/2486001.2486010

[73] Apostolaki Maria, Zohar Aviv, and Vanbever Laurent. 2017. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE.

[74] NL-ix. 2019. https://www.nl-ix.net/noc/how-get-most-out-your-nl-ix-connection/

[75] Alex Pilosov and Tony Kapela. 2008. Stealing the Internet: An Internet-scale man in the middle attack. *NANOG-44, Los Angeles, October* (2008), 12–15.

[76] Bruno Quoitin, Steve Uhlig, and Olivier Bonaventure. 2002. Using Redistribution Communities for Interdomain Traffic Engineering. In *Proceedings of the 3rd International Conference on Quality of Future Internet Services and Internet Charging and QoS Technologies 2Nd International Conference on From QoS Provisioning to QoS Charging (QofIS'02/ICQT'02)*. Springer-Verlag, Berlin, Heidelberg, 125–134. http://dl.acm.org/citation.cfm?id=1754656.1754672

[77] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack. 2016. HEAP: Reliable Assessment of BGP Hijacking Attacks. *IEEE Journal on Selected Areas in Communications* 34, 6 (June 2016), 1849–1861. https://doi.org/10.1109/JSAC.2016.2558978

[78] Brandon Schlinker, Kyriakos Zarifis, Italo Cunha, Nick Feamster, and Ethan Katz-Bassett. 2014. PEERING: An AS for us. In *ACM Workshop on Hot Topics in Networks*. ACM, 18.

[79] Job Snijders. 2016. Practical everyday BGP filtering with AS_PATH filters:Peer Locking. *NANOG-67, Chicago, June* (2016).

[80] Florian Streibelt. 2019. BGP Communities - A Weapon for the Internet (Part 2). https://labs.ripe.net/Members/florian_streibelt/bgp-communities-a-weapon-for-the-internet-part-2

[81] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP Communities: Even More Worms in the Routing Can. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*. ACM, New York, NY, USA, 279–292. https://doi.org/10.1145/3278532.3278557

[82] Y. Sun, A. Edmundson, N. Feamster, M. Chiang, and P. Mittal. 2017. Counter-RAPTOR: Safeguarding Tor Against Active Routing Attacks. In *2017 IEEE Symposium on Security and Privacy (SP)*. 977–992. https://doi.org/10.1109/SP.2017.34

[83] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. 2015. RAPTOR: Routing Attacks on Privacy in Tor. In *USENIX Security Symposium*. 271–286.

[84] Russ White. 2003. *Deployment Considerations for Secure Origin BGP (soBGP)*. Internet-Draft draft-white-sobgp-bgp-deployment-01. IETF Secretariat. https://tools.ietf.org/html/draft-white-sobgp-bgp-deployment-01 https://tools.ietf.org/html/draft-white-sobgp-bgp-deployment-01.

[85] X. Zhang, H. C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen. 2011. SCION: Scalability, Control, and Isolation on Next-Generation Networks. In *IEEE Symposium on Security and Privacy (SP)*. 212–227. https://doi.org/10.1109/SP.2011.45

[86] Zheng Zhang, Ying Zhang, Y. Charlie Hu, Z. Morley Mao, and Randy Bush. 2008. Ispy: Detecting IP Prefix Hijacking on My Own. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication (SIGCOMM '08)*. ACM, New York, NY, USA, 327–338. https://doi.org/10.1145/1402958.1402996

A ROUTER PREFERENCE DECISION FACTORS

Multi-Exit Discriminators (MEDs) are compared before the IGP metric, but MEDs are disabled on many BGP sessions and are not relevant to these attacks. Thus, we study routing decisions in the absence of MEDs. There are other tie breakers before the IGP metric that always tie for externally learned Internet routes, and there are several tie breakers after the IGP metric that behave similarly to the IGP metric because they are also functions of what External BGP (eBGP) session a route was learned on (like the router-id tie breaker). Thus, although route selection is arbitrarily configurable and varies by vendor, the model we present works as an effective abstraction.

B OTHER METHODS OF ACHIEVING INTERCEPTION

BGP communities offer a fine-grained method of controlling announcement propagation, but they are not universally supported. An adversary may have a provider that does not forward communities. However, an adversary can still exploit discrepancies in routing policies that shape announcement propagation to achieve interception.

Consider a scenario where providers A and B do not have peering links and are customers of the sets of tier-1 providers $\mathbb{A}$ and $\mathbb{B}$ respectively. The victim is a customer of the set of tier-1s $\mathbb{V}$. Assuming there is no overlap between $\mathbb{V}$ and $\mathbb{B}$ (if there is indeed overlap between $\mathbb{V}$ and $\mathbb{B}$ the adversary's job is easier), an adversary must 1) find a route filtering policy that is applied to customer routes and is used by providers in $\mathbb{A} \cap \mathbb{B}$ but is not used by at least one provider in $\mathbb{A} - \mathbb{B}$ 2) find B's most preferred tier-1 provider (in the case of routes with equal AS paths) and find a route filtering policy that is applied to peer routes and is used by this provider. Because routing policies vary, it is possible for an adversary to find required policies that match the above criterium. Below are some of such policies.

RPKI Route Origin Validation (ROV). The adoption of RPKI has been growing and so is the adoption of filtering policies that drop routes with invalid RPKI origins. However, these policies are inconsistent across ASes (not only do some tier-1s perform validation while others don't, whether validation is performed on peer-learned routes, client-learned routes or both is also inconsistent [5, 48]) potentially allow an adversary to make an RPKI invalid BGP announcement that will be dropped by select ASes, thus enabling interception. Although the announcement may be more suspicious given that it is RPKI invalid (although some routes in the global route table are RPKI invalid [35, 56, 65]), this may be compensated by the increased stealthiness gained through interception.

Defensive AS-path Filtering. As mentioned in [79], some ASes filter customer routes that contain the ASN of large peer networks anywhere in the AS path. However, which providers implement this policy and the exact ASNs that cause these routes to be filtered vary. Through exploration with its own prefix, an adversary can find which ASNs in the AS path will cause the route not to be imported at select tier-1s. Importantly, each ASN in the AS path has the downside of making the announcement longer and thus attracting less Internet traffic (as outlined in §5.3). However, such a strategy can potentially be carried out with a single ASN that will cause filtering at the required tier-1s while path poisoning may require a greater number of ASNs to be successful.

Route Flap Dampening (RFD) Route Flap Dampening (RFD) is another routing policy that has varying support at tier-1s [5, 23]. An adversary can continually modify its announcements as to trigger RFD at tier-1s that implement it (and maintain a route to the victim) while allowing other tier-1s to still propagate the adversary's route.

C COMMUNITY SUPPORT BY TOP ISPS AND INTERNET EXCHANGES

Through manual inspection of routing policies and usage guides, we verified community support for the top 30 ISPs (in Table 10) as stated in routing policy and top 10 Internet exchanges (in Table 11).

D FEASIBILITY AT SEATTLE

Following is our implementation of the algorithm from §4.4 at the PEERING mux in Seattle. The algorithm performed 4 iterations of `CollectInfo` and `AddCommunities` (① — ④).

`MakeSampleAnnouncement`: We made a sample announcement for the adversary's prefix.

① `CollectInfo`: We recorded the route exported by Coloclue at as [Coloclue, NTT, RGNet, Adversary]. Coloclue's route to the victim was [Coloclue, Fiberring, Cogent, Northeastern, Victim]. We noticed that Coloclue's path the the victim was provider-learned while Coloclue's path to the adversary was peer-learned which would cause Coloclue not to export a valid route to the victim.

① `AddCommunities`: A logical first choice was to add the community (NTT:No export to Coloclue) to prevent Coloclue from learning this route.

② `CollectInfo`: Coloclue now exported the route [Coloclue, Sprint, RGNet, Adversary].

② `AddCommunities`: We applied the (Sprint:No export to Coloclue) community, which successfully stopped Coloclue from exporting the route.

③ `CollectInfo`: By removing all of its peer routes to the adversary's prefix, Coloclue exported a provider-learned route: [Coloclue, Atom86, NTT, RGNet, Adversary]. Even though NTT was no longer exporting its route to Coloclue through its peering session, Coloclue was still learning NTT's route through its provider Atom86, because Coloclue is both a peer and an indirect customer of NTT. The provider-learned route exported by Coloclue to the adversary was the same length as its route to the victim, meaning there was a chance Colcolue would export a valid route to the victim. Therefore, we decided to suppress the route [Coloclue, Atom86, NTT, RGNet, Adversary] as well.

Community	Target AS	Action	Reason
65000:8283	Sprint	No export to Coloclue	Prevent Coloclue from learning shorter route from Sprint
65520:2203	NTT	Lower local preference in Netherlands	Prevent Coloclue (or Coloclue's providers in the Netherlands) from learning shorter route from NTT

Table 9: Communities used to achieve interception at Seattle.

ASN(Name)	LowerPref	NoExportSelect	NoExportAll
3356 (Level3)	Yes	Yes	Yes
1299 (Telia)	Yes	Yes*	Yes
174 (Cogent)	Yes	Yes**	Yes
2914 (NTT)	Yes	Yes	Yes
3257 (GTT)	Yes	Yes	Yes
6762 (Sparkle)	Yes	No	Yes***
6939 (Hurricane)	No	No	No
6453 (TATA)	Yes	Yes	Yes
3491 (PCCW)	Yes	Yes*	Yes
6461 (Zero)	Yes	Yes*	Yes
1273 (Vodafone)	Yes	Yes*	Yes
3549 (Level3)	Yes	Yes*	Yes
9002 (RETN)	Yes	Yes	Yes
12956 (Telefonica)	unknown	unknown	unknown
4637 (Telstra)	No	No	No
209 (CenturyLink)	Yes	Yes*	Yes
7473 (SINGTEL)	unknown	unknown	unknown
12389 (Rostelecom)	Yes	Yes*	Yes
20485 (TransTeleCom)	No	Yes*	Yes
3320 (Deutsche)	Yes	Yes	Yes
701 (MCI)	Yes	No	Yes
7018 (AT&T)	Yes	No	Yes
7922 (Comcast)	Yes	Yes	Yes
5511 (Orange)	Yes	Yes*	Yes
8359 (MTS)	No	Yes*	Yes
3216 (Vimpelcom)	Yes	Yes*	Yes
2828 (MCI)	Yes	Yes*	Yes
31133 (MegaFon)	Yes	Yes*	Yes
286 (KPN)	Yes	Yes	Yes
20764 (RASCOM)	Yes	Yes*	Yes

Table 10: Community support (as stated in routing policy [1–4, 6–11, 13–16, 22–31, 33, 34, 43, 45]) by the top 30 ASes (as per as-rank.caida.org accessed March, 2019). * Does not allow export control to peers via ASN but enumerates major peering sessions and allows for suppression to individual peering sessions via communities. ** Only allows suppression to private peers by region. * Only allows suppression to public peers (not private peers).**

③ `AddCommunities`: The route [Coloclue, Atom86, NTT, RGNet, Adversary] has no peering links in it, so we decided to add the community (NTT:Lower the preference of the route) to lower the local preference of the adversary's route at NTT.

④ `CollectInfo`: Coloclue announced a longer route to the victim than the adversary, so we knew the it's safe to launch the attacks.

Name	NoExportSelect	NoExportAll
DE-CIX	Yes	Yes
AMS-IX	Yes	Yes
IX.br	Yes	Yes
LINX	Yes	Yes
NL-IX	No	No
France-IX	Yes	Yes
HKIX	Yes	Yes
Seattle-IX	Yes	Yes
JPNAP	unknown	unknown
Netnod	Yes	Yes

Table 11: Community support at IXPs' route servers (as stated in routing policy [18, 19, 32, 36–40, 74]) by the top 10 IXPs operated by unique organizations (as per the Packet Clearing House list of Internet exchanges by average traffic accessed August, 2019 [17]). LowerPref is not considered because route servers are always across peering links, so an adversary can use NoExportAll to prevent route exporting and does not need to employ LowerPref.

Continent	Hosts in Sample	Expected Hosts in Sample	Chai-squared Contribution
Asia	159	171	.887
Europe	181	175	.216
Africa	4	3	.166
Oceania	14	16	.373
Americas	641	634	.078
Unknown	1	1	.001
Total	1000	1000	1.72

Table 12: Expected values rounded to nearest host (based on entire Censys database) and observed (from sample) values of number of hosts in each continent as well as chai squared contribution.

`LaunchAttack`: We announced the victim's prefix and successfully achieved interception demonstrating the viability at Seattle (the full list of communities used is shown in Table 9).

E VALIDATING RANDOM SAMPLE WITH CHAI-SQUARED ANALYSIS

For our sample of 1,000 HTTPS hosts, we computed a chai-squared value to confirm our sample was not biased. We counted how many hosts in our sample were in each continent. We also computed expected values for the number of hosts in each continent based on the fraction of hosts in that continent (that served browser-trusted certificates) in the entire Censys database using SELECT COUNT(ip) and GROUP BY autonomous_system.country_code (see Table 12). We computed a chai-squared value of 1.72. With 5 degrees of freedom (for the 6 possible continent values), a chai-squared value of 1.72 is well below the 80th percentile critical value of 8.558 and has a P value of .89 (meaning 89% of random samples have greater variation than our sample and 11% have lower variation).

For our sample of 1,000 ICMP ping hosts, a chai-squared analysis was not relevant because we could not filter the original Censys database for ping support. Thus, unlike the HTTPS case where our filtering simply confirmed the hosts were currently active, by filtering for ping support we knowingly made the sample unrepresentative of the entire Censys database (since it only included the hosts that responded to ping). However, we hold that this sample is representative of ping hosts given that it was constructed using the same overall sampling technique as the HTTPS sample.

F LIMITATIONS OF SICO ATTACKS

The primary limitations of SICO Attacks are their reliance on support for BGP communities and their reliance on predicting the exact route preferences of different ASes. However, measures can be taken to (at least partially) overcome these limitations.

Adversaries can work around gaps in community support. For example, if an adversary has two providers but only one provider offers support for action communities, the adversary can choose to make announcements to the provider that supports action communities. In addition, even if an adversary's providers do not support any action communities, as long as they transit communities to higher up ASes, an attack can still be viable.

Route preference (as well as the routes heard by an AS) can be seen through a BGP looking-glass (a service that shows which routes are heard by an AS and their preference for debugging purposes) [21]. While many tier-1 providers offer public looking-glasses, support from smaller networks is less common. Knowing the full set of routes heard by an AS without a looking glass can be difficult, but knowing an AS's preferred route is easier because this route is exported to neighbors and can be seen from the looking glasses of other ASes (and in publicly-available BGP data like [44]). Interestingly, once an adversary knows an AS's preferred route to AS_{adv}, it can deduce all routes to AS_{adv} that that AS heard by suppressing the preferred route (via communities) and then observing what second-choice route that AS exports. In this manner, an adversary can find all routes heard and the preference of these routes at an AS that does not contain a looking glass.