

Global BGP Attacks that Evade Route Monitoring

Henry Birge-Lee^[0000-0001-7141-345X], Maria Apostolaki^[0000-0003-0342-2631],
and Jennifer Rexford^[0000-0002-0231-8165]

Princeton University, Princeton NJ 08544, USA

Abstract. As the deployment of comprehensive Border Gateway Protocol (BGP) security measures is still in progress, BGP monitoring continues to play a critical role in protecting the Internet from routing attacks. Fundamentally, monitoring involves observing BGP feeds to detect suspicious announcements and taking defensive action. However, BGP monitoring relies on seeing the malicious BGP announcement in the first place. In this paper, we develop a novel attack that can hide itself from all BGP monitoring systems we tested while potentially affecting the majority of the Internet. The attack involves launching a sub-prefix hijack with the RFC-specified NO_EXPORT community attached to prevent networks with the malicious route installed from sending the route to BGP monitoring systems. While properly configured and deployed RPKI can prevent this attack and /24 prefixes are not viable targets of this attack, we examine the current route table and find that 38% of prefixes in the route table could still be targeted (see Section 4). We also ran experiments in four tier-1 networks and found all networks we studied could have a route installed that was hidden from global BGP monitoring. Finally, we propose a mitigation that significantly improves the robustness of the BGP monitoring ecosystem. Our paper aims to raise awareness of this issue and offer guidance to providers to protect against such attacks.

Keywords: Network Security · BGP · BGP Monitoring.

1 Introduction

The Border Gateway Protocol (or BGP) allows for the exchange of routes between the independently-operated networks (known as Autonomous Systems or ASes) that comprise the Internet. BGP announcements contain reachability information for IP prefixes and are propagated from one network to the next to build the global IP routing table. However, BGP was designed with no form of message authentication, which allows adversaries to construct bogus BGP announcements for IP prefixes they do not control or legitimately route to. These incidents are known as BGP attacks or BGP hijacks. BGP attacks cause Internet traffic destined to a victim AS to be maliciously routed to the attacker’s infrastructure. BGP attacks can cause significant damage, particularly as these routing attacks can be used to target higher-level applications run on top of the Internet [26, 28, 29, 39, 54, 55].

While securing BGP is fundamental to the security of many critical applications that run on top of the Internet, robust BGP security is not practical today with available technologies. Relatively high-security proposals like BG-Psec require updates to existing hardware and have not yet seen any production deployment [51]. Alternatives like Resource Public Key Infrastructure (RPKI) do have significant deployment but are still not ubiquitous and can suffer from misconfigurations [35,38].

A pragmatic and commonly used alternative to comprehensive BGP security is BGP monitoring. BGP monitoring involves observing BGP updates and flagging updates that appear suspicious. One advantage of BGP monitoring is that it leaves the BGP announcement syntax untouched. However, BGP monitoring has a core limitation: it cannot detect a route it cannot see [46]. Existing BGP monitoring systems have overcome this limitation by combining data from many vantage points spread across the Internet. With two of the most popular monitoring services (RIPE RIS and RouteViews) each having over 1000 different BGP peering sessions [2,3] with various top networks, this problem appears to be solved. Previous work on attacks that evade BGP monitoring only succeeded in affecting 2% of Internet hosts at best [31,46].

In this paper, we demonstrate that even with the vast number of peering sessions with monitoring services, if an adversary is capable of bypassing prefix filters and installing its route in the tables of tier-1 transit providers,¹ the adversary can launch a stealthy BGP attack (i.e., not seen by public BGP monitoring) that still succeeds. We introduce a novel attack on BGP monitoring that allows an adversary to stop networks peered with monitoring services from exporting the adversary’s malicious announcements to monitoring services while still directing traffic via the adversary’s route. Our attack involves the adversary tagging its BGP announcement with a specific value (the RFC-specified NO_EXPORT BGP community) so that networks using the adversary’s route do *not* send it to BGP monitoring services. Even though this misuse of the NO_EXPORT community limits the spread of the adversary’s route, the adversary can still install this route into networks commonly used to forward traffic to the victim using a *sub-prefix* BGP attack. This makes the effect of this attack substantial, sometimes even affecting the vast majority of the Internet.

We ethically demonstrate this attack in the wild (i.e., attacking an IP prefix we control) and found it can be highly effective, in our case capturing traffic from **all** of a sample of 1k random Internet hosts. Even while affecting nearly the entire Internet, this attack was invisible to all BGP monitoring services we studied (which included RouteViews, RIPE RIS, Cisco Crosswork Cloud, and ThousandEyes).

RPKI is an effective and deployed protection against this type of attack, and we studied the viability of the attack under the current RPKI deployment. We found that 38% of prefixes in the route table are still vulnerable to this attack (as discussed in Section 4). We studied the behavior of the NO_EXPORT

¹ Several high-profile BGP attacks have involved adversaries directly installing routes into tier-1 transit networks [4–6].

community at four tier-1 networks and found all networks we studied hid announcements from all monitoring services we studied when the community was attached. In fact, assuming an adversary only installed its malicious route at networks we confirmed were susceptible to this attack, the adversary could hijack traffic between 23% of random send/receiver AS pairs, according to our simulation. Finally, we conclude with recommendations that can significantly reduce the viability of this attack with simple configuration changes.

2 Threat Model

We assume an adversary capable of making a malicious BGP announcement with the aim of redirecting traffic. This traffic redirection can be used to attack several different critical applications that run atop the Internet [26, 30, 55]. The adversary also aims to avoid detection by BGP monitoring (i.e., launch a stealthy BGP attack) to prevent the victim from taking defensive action. We assume the adversary’s announcement is not stopped by IP prefix filters (as is the case with observed real-world attacks [28, 36, 39]).

Furthermore, we assume the adversary will willingly seek new BGP transit or peer sessions to enable its attack. Internet eXchange Points (IXPs) have many networks colocated in a single location. This allows an adversary to cheaply establish many peer and transit sessions from a single Point of Presence. Furthermore, Virtual Machines (VMs) at IXPs can be rented quite cheaply (e.g., ~\$30 a month [7]) and have ports on the IXP’s peering LAN. Thus, these VMs can establish BGP sessions with other networks at the IXP [7–9].

Using an IXP VM, an adversary can approach colocated ISPs and request BGP sessions to propagate its malicious announcement. Social engineering or manipulation of Internet Routing Registry (IRR) data (as done in past attacks [36]) can persuade ISPs into white-listing a victim’s IP prefixes on an adversary’s BGP session. Many major BGP attacks have been launched by direct customers of tier-1 ISPs [4–6], and some adversaries have used multiple tier-1 ISPs to spread their attack [5].

3 Attack Details

Our attack relies on two key insights. First, the behavior of the RFC-defined `NO_EXPORT` community affects a network’s sessions with BGP monitors. Second, because of longest-prefix-match forwarding, a network may be directing traffic over a route without hearing it or installing it into its RIB.

3.1 Hiding the Route: `NO_EXPORT`

At the core of our attack is the ability to hide a route from BGP monitoring systems even when that route is installed in routers that provide a feed

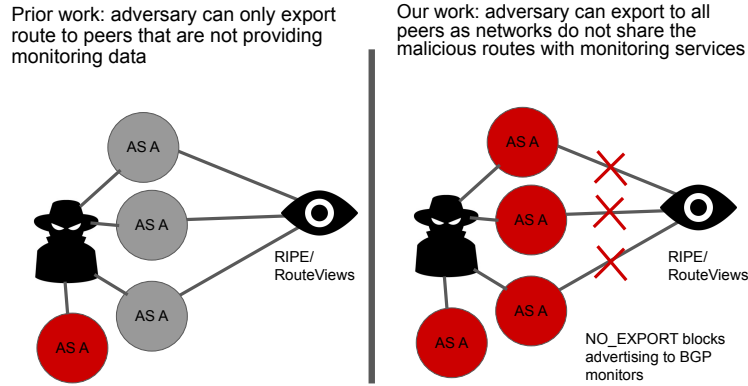


Fig. 1: Using our technique, an adversary can announce its malicious route to more peers while maintaining stealth. Even networks that directly provide data to monitoring services will not share the adversary’s malicious route due to the behavior of the NO_EXPORT community on BGP monitoring sessions.

to BGP monitoring systems. We accomplish this using an RFC-defined “well-known” BGP community called NO_EXPORT.² BGP communities are 32-bit tags that can be attached to BGP announcements and can impact how routers hearing those announcements behave [33, 40, 53]. When attached to a BGP announcement, the NO_EXPORT community instructs routers not to export that BGP announcement beyond a confederation boundary. Most routing software honors this behavior by default [10, 11], as support for the NO_EXPORT community is stipulated by the same RFC that defines BGP communities [33]. We utilize the key insight that **when the NO_EXPORT community is attached to a route, routers do not export that route on sessions with BGP monitoring systems.**

This is because most BGP monitoring systems use multi-hop External BGP (eBGP) sessions with participating peer networks to receive BGP updates [12–14]. To the routers that handle these sessions, they appear to be with an external neighboring network, and thus, the export restrictions of the NO_EXPORT community are applied. This creates the conditions that enable our attack: a router can have a route installed but not send it to monitoring systems. These conditions allow a malicious route to be announced directly to and used by networks that provide data to monitoring services while maintaining full stealth. This represents a significant improvement over previous work where a route would be detected if it was used by any AS sending data to BGP monitoring services [31, 46]. These differences are highlighted in Figure 1.

² The community NO_EXPORT_SUBCONFED has a similar behavior and can largely be used interchangeably with the NO_EXPORT community for this purpose.

3.2 Attracting Traffic: Sub-prefix Hijacks

Hiding the route from BGP monitors does not, by itself, allow for a widespread attack. The use of the NO_EXPORT community means the adversary can only affect as many networks as it has direct BGP sessions. This is because all ASes use the same RFC-defined value for the NO_EXPORT community, so the first AS that sees an adversary’s announcement will act upon the community and not export the adversary’s route any further. This makes it difficult to scale to thousands of ASes.

However, the adversary can attract more traffic using a *sub-prefix* attack where it announces a longer, more-specific prefix than the victim (i.e., true prefix owner) it is targeting. Longest-prefix-match IP forwarding means that routers will always use the most-specific (longest) IP prefix possible that matches a particular packet. Thus, in any network with the adversary’s sub-prefix attack installed, the adversary’s route will take priority over the victim’s. With this in mind, **the adversary can install its malicious sub-prefix route in large transit providers or upstream of the victim** that, due to their topological location, would naturally carry a large amount of the victim’s Internet traffic. Even though the propagation of the adversary’s announcement is limited by the NO_EXPORT community, the victim’s benign BGP announcement attracts traffic to networks with the adversary’s route installed. Longest-prefix-match forwarding then causes the traffic to detour to the adversary (see Fig. 2).³

3.3 Adversary Strategy

Combining these two techniques leads to the following attack strategy:

1. Select a victim prefix for the attack.
2. Use public BGP data to see which major transit providers carry traffic to that prefix (this can be trivially derived by seeing the most commonly used upstream ASes used for that prefix).
3. Rent a VM at an IXP where several of those transit providers collocate.
4. Approach these transit providers and request a BGP transit session.
5. Use social engineering and manipulation of routing data to trick these transit providers into whitelisting a sub-prefix of the victim’s prefix.
6. Announce a malicious sub-prefix of the victim’s prefix with the NO_EXPORT community.

This attack strategy is very effective as it affects traffic from a large portion of the Internet (potentially 100%, if an adversary installs the malicious route in all of a victim’s neighbors) while being completely stealthy. Furthermore, although some steps, like the establishment of BGP transit sessions, bear a cost, this attack’s stealth makes this infrastructure significantly more reusable. Presuming

³ One potential countermeasure to protect against sub-prefix hijacks is RPKI. However, RPKI deployment is still not ubiquitous [15, 43] and flaws in RPKI configurations still leave networks vulnerable to sub-prefix attacks [35].

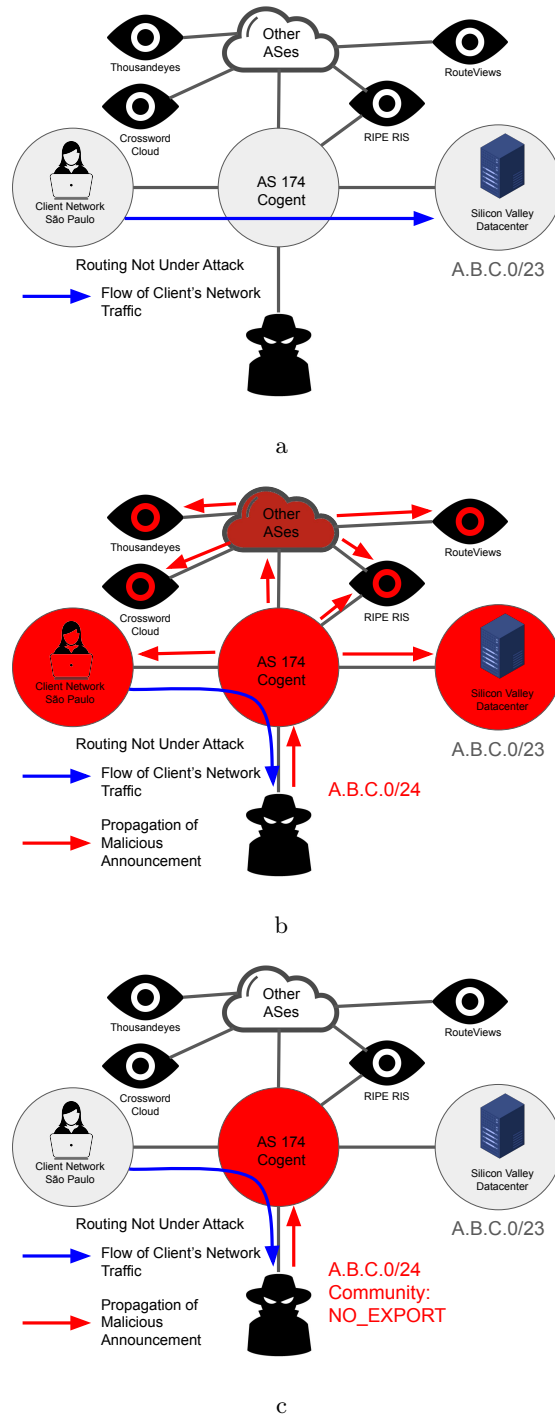


Fig. 2: Routing during benign operations (2a), a control non-stealthy attack (2b), and a stealthy attack (2c).

the attack goes undetected, the reputation cost of the attack is zero. This allows an adversary to potentially reuse its ASN, IXP VM, and transit links for many attacks. Stealthy attacks can avoid the negative reputation that often causes adversary infrastructure to get destroyed [5].

4 Viability of Launching a Sub-Prefix Attack

This attack strategy relies on being able to launch a sub-prefix BGP attack against the victim. While many major ISPs filter customer BGP announcements with white-list filters based on IRR data, there are cases of adversaries using social engineering and/or malicious IRR data to circumvent such filters [28,36]. An adversary can also claim to be the upstream provider of a victim by adding the victim to its AS-SET in its IRR entry [36]. Being able to circumvent such filters and propagate its malicious BGP announcement is a requirement of any BGP attack and frequently occurs in the wild [5,28,36,39]. Barring strict prefix filters, two primary mechanisms remain to limit an adversary's ability to launch a sub-prefix attack: Resource Public Key Infrastructure (RPKI) and 24-bit prefix length filters.

RPKI is a cryptographic database of IP prefix ownership that can be used for filtering BGP announcements through a process known as Route Origin Validation or ROV [32]. ROV can be evaded by an adversary putting the ASN of the true prefix owner at the end of the AS path of its announcement. However, RPKI data also contains the required IP prefix length of any valid BGP announcement for a covered IP address. Route Origin Authorizations or ROAs are signed entries in RPKI databases that contain information regarding an allowed prefix, its length, and its appropriate origin ASN. ROAs have an optional `max_length` attribute which specifies the maximum IP prefix length for any BGP announcements of a sub-prefix covered by the ROA [35]. If the `max_length` is left empty, only the exact IP prefix length specified in the ROA is permitted. This system can be used to defend against sub-prefix attacks because ASes that perform ROV will reject announcements that are longer than a permitted prefix length. However, there are issues of network operators sometimes configuring overly permissive max lengths, which sometimes hinders RPKI's ability to prevent sub-prefix attacks [35]. If a `max_length` is specified, which is longer than the IP prefix seen in BGP, then an adversary can make a sub-prefix announcement for the victim's prefix, which is still RPKI valid and will not be filtered by ROV.

The other primary defense against sub-prefix attacks is 24-bit prefix length filters. In an effort to fight route table bloat, many network operators began capping the permitted maximum prefix length for default-free zone BGP announcements at 24 bits. Thus, BGP announcements for prefixes longer than 24 bits are often filtered [27]. While this often holds in general for global route propagation, there is a chance an ISP may choose to violate this rule for `NO_EXPORT` prefixes as these are not propagated beyond the receiving AS (thus reducing the

impact of these smaller prefixes on the global route table).⁴ While this may be true, /24 customer prefix filters are still a recommended best practice [56] and need to be considered when addressing the viability of sub-prefix attacks.

To better understand the viability of sub-prefix attacks, we downloaded the Routing Information Base (or RIBs) of all RIPE RIS collectors on August 16th, 2024. We filtered the RIBs to exclude any prefixes that had a private ASN as the origin AS, as these were likely internal-only announcements. Using these filtered RIBs, we collected a database of any IP prefixes seen in any RIS collector to find 1.3 million IP prefixes in total. We also ran the RPKI tool Routinator [48] to fetch the current RPKI database and condense it into a CSV output. Using this data, we performed a count of:

- How many IP prefixes are covered by RPKI?
- How many IP prefixes are covered by RPKI and have a maximum permissible length that is equal to the length observed in BGP?
- How many IP prefixes are 24 bits long?
- How many IP prefixes are 24 bits long or are covered by RPKI with a proper maximum length?

We found of the 1.3 million prefixes 711k (54%) were covered by an RPKI record. This result largely aligns with the NIST RPKI monitor, which currently reads 53% as of Sep 9 2024 [15]. However, this result changes dramatically when the appropriate max length is considered. Only 501k prefixes (38%) were in RPKI with a proper max length. This large threat surface introduced by max-length is also supported by past research on the max-length attribute [35].

We additionally found that 638k (49%) of prefixes had a length of 24 bits.⁵ When we counted prefixes that had either a 24-bit length or were RPKI covered with a proper max length, we found that 811k (62%) prefixes had some protection against sub-prefix attacks. While this is substantial, it is a conservative estimate, assuming all ISPs perform ROV and filter /24 bit announcements. Several major ISPs are known not to do ROV [52], and prefixes longer than /24 are still propagated on occasion [27]. Regardless **at least 38% of the global route table is still vulnerable to sub-prefix attacks** making this stealthy attack still viable against a large number of victim prefixes.

5 Ethically Launching the Attack

We ethically launched this attack on the Internet using the cloud provider Vultr. Vultr allows customers with their own IP address space to make BGP announcements. Using space allocated by our affiliated institution Princeton University,

⁴ This case is somewhat similar to the well-established allowing of prefixes longer than 24 bits for black-hole announcements [44].

⁵ While this is a substantial percentage, it is worth noting that these are, by definition, some of the smallest prefixes in the route table so the fraction of IP addresses covered by 24-bit prefixes is likely significantly less than 49%.

we established BGP sessions with Vultr’s routers from virtual machines at various Vultr data centers. We relied on the BIRD BGP daemon v1.6.8 running on Ubuntu 22.04 to handle the sessions with the Vultr routers and make BGP announcements. Our routers were configured as part of AS400065 which was the origin AS of our announcements (AS-VULTR AS20473 was the upstream). Also, note that Vultr operates several topologically disjoint data centers and does not use any private backbone for customer traffic. Because of this, these data centers can be thought of as distinct BGP-speaking nodes. With this real-world attack, we confirmed that the NO_EXPORT community offers stealth from monitoring services while allowing us to install a route into a major transit provider’s network.

Our experimentation was completely ethical in that 1) all nodes making BGP announcements for these prefixes (including the ones being used as the adversary) had proper authorization to make these announcements, 2) the IP prefixes used were allocated specifically for research purposes and operated no real network services, 3) our experiment involved very few BGP announcements that did not pose any undue burden on production routers, and 4) we only used standard BGP attributes and community values that are commonly used in production routing configurations.

Normal Routing: We began by making a standard BGP announcement from Vultr’s Silicon Valley data center for the prefix A.B.C.0/23 to the provider Cogent, which provides transit to that data center. This announcement provided routing for our webserver that was hosted in the Silicon Valley data center and served as the hypothetical victim’s server. We confirmed this announcement had converged using various BGP looking glasses [16, 17] (services provided by networks that allow operators to query routes used by those networks), and we confirmed data-plane connectivity to the webserver. At this point we configured four BGP monitoring services (Cisco Crosswork Cloud, ThousandEyes, RIPE RIS, and RouteViews) to monitor the victim’s announcement and throw an alert if any sub-prefix announcements were noticed. We also configured a testing client in Vultr’s São Paulo data center that had a BGP session with Vultr’s upstream router that we used to 1) test connectivity to the victim’s webserver and 2) use “show route” on our BGP router to observe this client’s control-plane route to the victim’s prefix. As expected, this client correctly routed to the victim’s webserver and installed the proper control-plane route as announced by the victim.

Non-Stealthy Attack: To verify our monitoring configuration, we made a non-stealthy control BGP hijack from the Vultr data center in New Jersey (which served as the adversary in our experiment) for a sub-prefix of the victim’s prefix (A.B.C.0/24). This attack was effective in that traffic from our test client in São Paulo (as well as any other clients tested) was routed to the adversary. However, all control-plane monitors detected this attack. All four monitoring services generated notifications, and the Bird router at São Paulo had installed

the adversary’s prefix. Thus, while capable of hijacking the victim, this attack was highly noticeable.

Stealthy Attack: We withdrew this control announcement and updated our adversary’s configurations to launch a stealthy attack, as discussed above. To do this, we attached BGP communities supported by Vultr to instruct it to only export our route to the provider Cogent. Cogent (AS 174) offers a Cogent-specific version (value 174:990) of the RFC 1997 NO_EXPORT community as specified in their routing guide for customers [18]. This was crucial for enabling our experiments as we were not a direct customer of Cogent but instead had to propagate our announcements through Vultr first.⁶ Had we attached the RFC 1997 NO_EXPORT value to our BGP announcements, Vultr’s border routers would interpret this community and apply export restrictions coming out of *Vultr’s* network, preventing our announcement from ever reaching Cogent. On the other hand, the Cogent-specific NO_EXPORT value did not have any meaning to Vultr’s routers and was transparently transited to Cogent, where the Cogent routers interpreted this community.

The adversary announcement for launching this attack was for the prefix A.B.C.0/24 and included a set of communities supported by Vultr that instructed Vultr only to export the adversary’s route to Cogent and the 174:990 community supported by Cogent, preventing the exporting of the route beyond Cogent’s network. In this manner, the stealthy attack was only in the route tables of Vultr’s NJ data center and Cogent.

5.1 Attack Measurements

We confirmed this attack was both stealthy and effective via several measurements. We started by measuring the behavior of our São Paulo client, which routed traffic to the adversary instead of the victim. We used “show route” to examine the control plane route head by the client and found **the router still only showed the victim’s announcement, meaning this client was affected by the adversary’s attack while not hearing the malicious announcement in the control plane.** We expanded our measurements to look at the four BGP monitoring services we studied and found that none of them had thrown an alarm for the adversary’s route. We further confirmed that Cogent was a peer of RIPE RIS (as specified in the RIS peer list [2]) and during our control announcement, Cogent’s routers had exported the adversary route over this direct peering session. However, **when the NO_EXPORT community was applied, exporting of the route over the peering session with**

⁶ This was because we did not want to invest in obtaining a tier-1 provider just for this experiment. However, the cost of a BGP transit session is minuscule compared to the gains offered by a BGP attack [28, 39], making this a justifiable expense for the adversary.

RIPE RIS was indeed suppressed.⁷ We further confirmed that the adversary’s route was installed in Cogent routers via the Cogent looking glass [17]⁸ demonstrating the ability to infect a network with a malicious route and suppress that network’s reporting of the route to BGP monitoring services.

Having confirmed this announcement was fully stealthy to all the monitoring services we tested, we wanted to measure how much of the Internet was using the adversary’s route unknowingly. To do this we took a sample of 4096 random IP addresses and ping scanned them with the command “nmap -sn -iR 4096”. We are conscious of the ethical concerns with random scanning and ensured our technique was ethical by not performing any port scanning (preventing any behavior that could be seen as possibly probing for vulnerable services) and sending only a very small number (~ 3) of ICMP Echo requests to each host. We felt this scan was unlikely to raise any security alarms even on highly-monitored systems (as any public IP address is likely to be scanned several times as part of many projects) and imposed a negligible computational load even for under-provisioned systems. From this scan, we randomly selected 1000 responsive hosts to use in subsequent scans related to our attack.⁹ This sample is representative of IPv4 addresses that respond to ICMP Echo.

With the adversary’s attack active, we sent out ICMP Echo requests to each host in our sample from the victim’s machine using a source IP address in the hijacked prefix. Thus, the host’s ICMP Echo responses would be routed to the victim or the adversary, depending on whether that host’s network was affected by the attack (see Figure 3). When scanning the 1k sample, 739 hosts responded, all of which routed to the adversary (i.e., the victim was listening for responses but did not hear any while the adversary saw all 739 responses) **indicating that the attack had a global effect despite it being invisible to the route monitoring services we studied.** This was due to the victim exclusively using Cogent for transit and the adversary installing its malicious route into Cogent’s route table.

6 Broader Viability of the Attack

At a high level, there are three elements that need to be present for this attack to be viable: 1) the major ISPs the adversary intends to poison with its malicious route need to honor the NO_EXPORT community and apply it to their BGP

⁷ Cogent did not have a direct peering session with any of the other BGP monitoring services. Routes for the sub-prefix were suppressed in other monitoring services as well.

⁸ The looking glass is not a BGP monitoring service as it is intended for manual debugging of routes and requires a high-latency query to be run for each route lookup command. Some looking glasses even explicitly prohibit automated queries [16]. Thus, we do not consider the presence of the sub-prefix route in the looking glass as attack detection by BGP monitoring.

⁹ Not all hosts responded in subsequent scans, which reduced the effective sample size for each scan, but for consistency, we always sent to this same 1k host sample.

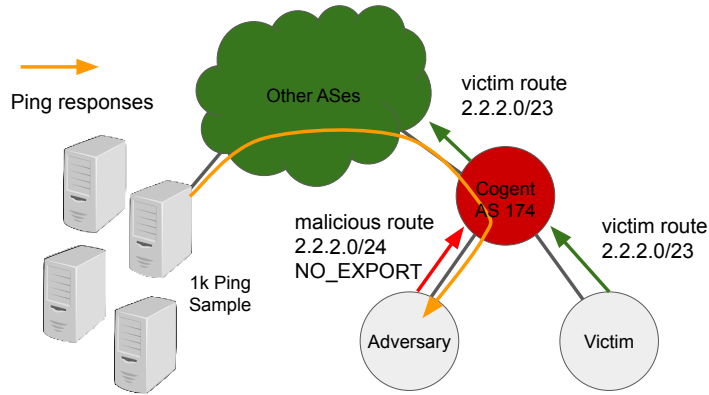


Fig. 3: The data plane during our real-world attack experiment. Because the victim exclusively used Cogent for transit and the adversary installed its malicious route into Cogent’s network, all ICMP Echo (ping) packets from the 1k random host sample were routed to the adversary.

monitoring sessions 2) the adversary needs to be able to launch a sub-prefix BGP hijack against its victim (discussed in Section 4) and 3) traffic from enough source ASes need to be routed through the networks poisoned with the adversary’s route for the attack to be effective. We analyze each of these aspects separately and find that each of these elements is viable in today’s Internet ecosystem.

6.1 NO_EXPORT Behavior at Tier-1 ISPs

We studied the NO_EXPORT community behavior required to launch this attack at several tier-1 ISPs and confirmed the vulnerable behavior at all networks we studied. Even though RFC 1997 states that any BGP speaker that understands communities shall implement well-known communities, networks could conceivably strip these communities from BGP route announcements, or filter routes with these communities. Checking for these practices would require a direct connection to the associated network—not possible with our Vultr test environment. Instead, we reached out to the operators of several other tier-2 providers to assist us by making BGP announcements tagged with the NO_EXPORT community to their tier-1 providers. We were assisted by the operators of RGNNet (AS3130 which used Sprintlink for transit) and NJEdge (AS21976 which used Telia and NTT for transit). These networks ran experiments using free IP blocks 192.83.230.0/24 and 130.156.183.0/24 respectively. Through this technique, we were able to confirm the behavior of the community at Arelion (AS 1299), NTT (AS 2914), and Sprintlink (AS 1239), in addition to Cogent (AS 174, which we tested through Vultr). At these networks, we confirmed the following behavior:

- These networks supported the NO_EXPORT community and installed routes with this community into their route tables (verified via looking glasses).
- These networks had a direct peering session with RIPE RIS [2] and RouteViews [3], except for Cogent, which only peered with RIPE RIS.
- These networks properly exported routes not tagged with the NO_EXPORT community to BGP monitoring but did not export our test announcements that contained the NO_EXPORT community.
- The announcements through these networks were *not* seen in Crosswork Cloud or ThousandEyes.

The networks we tested were all tier-1 ISPs that are on a significant number of paths. Even though it is difficult to scale these tests to large numbers of networks, our results strongly suggest the widespread viability of this attack. Based on the many networks known to support BGP communities [53], the language in RFC 1997 (“[these communities] operations shall be implemented in any community-attribute-aware BGP speaker”), and instructions for peering from BGP monitoring services [12–14], we believe the vast majority of networks are likely susceptible to this attack.

6.2 Impact of BMP and iBGP-based Monitoring Systems

Our experiment focused on the two largest public BGP monitoring systems currently deployed: RouteViews and RIPE as well as two popular commercial offerings operated by Cisco: Cisco Crosswork Cloud and ThousandEyes. Based on customer configuration guides, all of these sources collect data primarily using eBGP peerings with participating networks [12–14, 19]. Other options do exist for providing BGP data to monitoring services, notably the BGP monitoring protocol (BMP) and iBGP. BMP is a protocol designed specifically for routers to send data to monitoring services [49] and can avoid the issues of the NO_EXPORT community. Additionally, some monitoring services use iBGP sessions to send BGP data, whereas peering routers still treat the monitoring service like another router but think the neighboring router is part of the same AS. This is achieved by the monitoring service establishing its BGP session using the same AS as the peer network. iBGP similarly avoids the export restrictions of the NO_EXPORT community, which is only applied at AS boundaries. The iBGP approach is currently used by the commercial network monitoring service Kentik [20].

Sadly, these methods lack adoption by the public BGP monitoring services, and data availability of commercial systems is significantly more limited. Many commercial systems only obtain BGP feeds from customers and provide data from those feeds for use by this individual customer. Even if commercial services do share feeds between customers, the viability of this approach relies on both the true prefix owner and the network with the malicious route in its table, both of which subscribe to the same commercial monitoring solution. Finally, as demonstrated by Cisco Crosswork Cloud and ThousandEyes, many commercial services are still dependent on data collected from eBGP sessions and some

even have largely based their monitoring products off of RIPE and RouteViews data [21]. We believe that to have reliable detection of these stealthy attacks, changes must be made to the public monitoring infrastructure to make them detectable.

6.3 Simulating the Effective Spread of the Attack

We ran Internet topology simulations using the Gao-Rexford model of routing policies [34] to study the AS-level paths of Internet traffic between 150 randomly-chosen ASes. We measured the fraction of paths that contained an AS with the adversary’s route installed under various different attack scenarios. We found that **by making its announcement to only five tier-1 ISPs, an adversary can on average hijack 37% of routes from random ASes to a victim destination**. Thus an adversary can still affect a large portion of the Internet even with the NO_EXPORT community limiting the spread of its announcement. Additionally, if the adversary announced to only the networks at which we confirmed the behavior of the NO_EXPORT community, the adversary could still, on average, hijack traffic from 23% of source ASes to an average destination AS. This vastly outperforms the capabilities of previous stealthy attacks (which often affected less than 2% of the Internet [46]).

We then ran Internet topology simulations using the Gao-Rexford model of routing policies [34] on the CAIDA AS-Relationships Data Set [1]. We simulated all routes between 150 randomly chosen ASes from the CAIDA topology. For each of the AS-level routes generated, we tested to see if one of the ASes the adversary could compromise with a stealthy attack was in the simulated path. If we so, we considered this path compromised as the source of the traffic would unknowingly send its traffic to an AS that had a stealthy attack installed in its route table (which would forward the traffic to the adversary). We grouped the data by origin AS, allowing us to obtain a percentage of traffic sources (i.e., source ASes) affected by the stealthy hijack for each origin AS.

We began by considering an adversary that only installed its route at the four networks we found vulnerable to the attack in Sec. 6.1: AS 1299, AS 2914, AS 1239, and AS 174. **Using only these networks, which we confirmed are vulnerable, an adversary could, on average, hijack traffic from 23% of the Internet**. The CDF of the percent of Internet traffic that can be hijacked by this adversary when targeting different victim destination ASes is shown in Fig. 4. Notably, some victim ASes (represented at the top right of the CDF) that used exclusively compromised providers for Internet connectivity have 100% of their traffic affected by the stealthy attack, representing the global effectiveness with 0% observability example we saw in our ethical real-world attack.

Given the behavior we exploit is stipulated in RFC 1997 (so all ASes capable of understanding communities should exhibit this behavior) and all ASes studied correctly conformed to this RFC, it is reasonable to assume the vast majority of major networks behave in this manner, particularly given the large amount of BGP community support [31, 53]. Thus, we ran our simulation considering a strategic adversary that chooses ASes based on the size of their customer

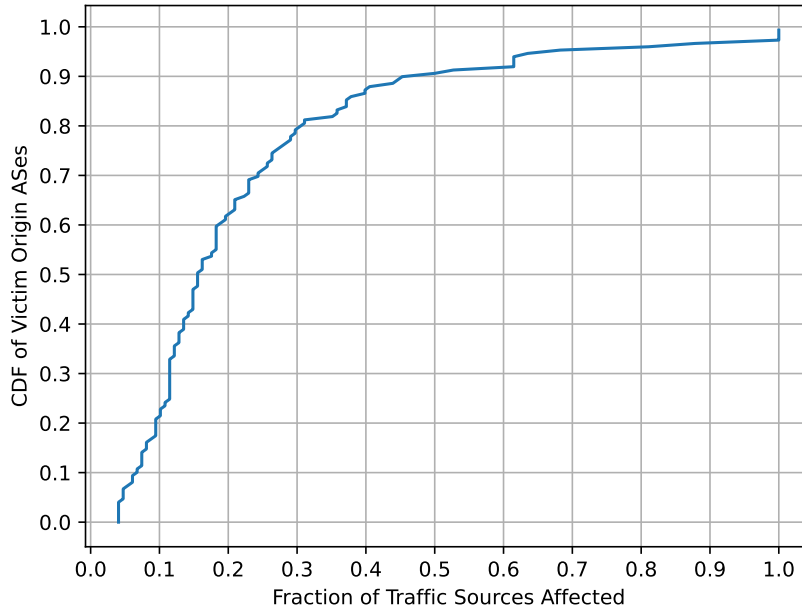


Fig. 4: A CDF of the fraction of traffic sources affected by a stealthy hijack launched by an adversary that installed malicious routes at the four ASes tested in this paper (that all are found to be vulnerable to the stealthy hijack).

cones (i.e., the number of direct and indirect customers that AS has). Based on the CAIDA AS-rank customer cone counts [22], we modeled an adversary that compromised a varying number of the top ASes (in order of customer-cone size: AS 3356 Lumen, AS 1299 Arelion, AS 174 Cogent, AS 2914 NTT, AS 6762 Telecom Italia Sparkle). If an adversary only installs its stealthy hijack at the largest AS: 3356 (Lumen), it, on average, can affect routes to 0.13% of destinations. Using only the top 2 networks, an adversary can affect 18% of destinations on average. This amount increases to 34%, 35% and 37% if the adversary installs its route in the top 3, 4, or 5 networks, respectively. A CDF of the effectiveness of these hijacks is shown in Fig. 5.

Overall, these simulations confirm that even if an adversary can only install its attack at a limited number of top networks, significant swaths of Internet traffic can be hijacked in a stealthy manner, making this hijack significantly more viable than previous techniques for launching stealthy BGP hijacks [46].

7 Defenses

7.1 Changes in BGP Configuration

At its core, this attack revolves around the fact that the well-known `NO_EXPORT` community restricts exporting to BGP monitoring sessions. The most immedi-

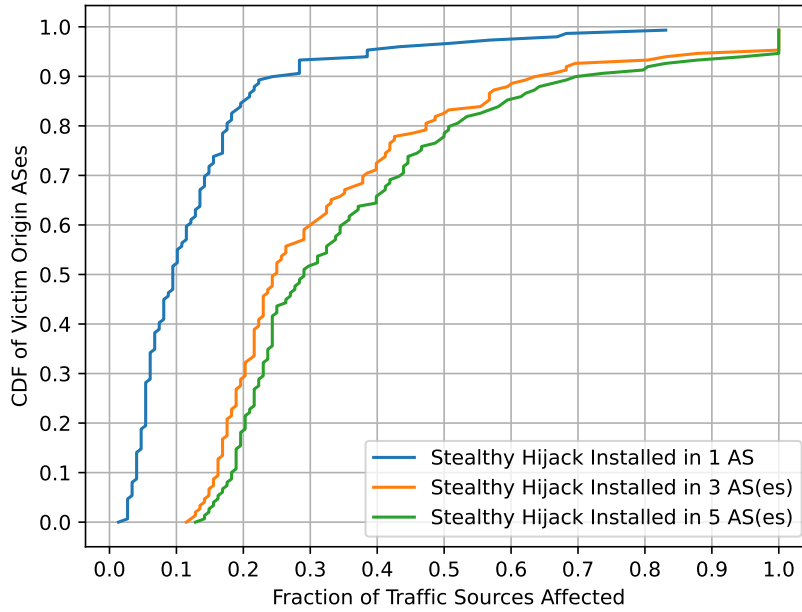


Fig. 5: A CDF of the fraction of traffic sources affected by a stealthy hijack launched by an adversary that installed malicious routes at the top 1, 3, and 5 networks by customer cone size.

ate mitigation to this attack is to change that behavior. This behavior can be changed without even restricting the use of the `NO_EXPORT` community or changing the way BGP monitoring is configured. Instead, the RFC 1997 value for the `NO_EXPORT` community can be rewritten (via an ingress policy on eBGP sessions with external ASes) to a different AS-specific community. This community can then be matched on egress sessions with customers, peers, and providers to restrict exporting. However, this community would not have any effect on BGP sessions with monitoring services. We discuss additional configuration options to mitigate these attacks and show configuration examples in Appendix A.

7.2 Use of BGP Monitoring Protocol (BMP)

The BGP Monitoring Protocol (BMP) [49] is a protocol specifically designed for the monitoring of BGP updates and RIBs that can expose prefixes tagged with `NO_EXPORT`. BMP could potentially be run between peer networks and monitoring services in place of the standard eBGP multi-hop sessions thus allowing for full visibility of the RIBs of participating networks by monitoring services. BMP is supported in many leading BGP implementations [23, 24] and has stand-alone implementations as well [25].

7.3 Increased Scope of BGP Monitoring

While the changes in BGP configurations and monitoring protocols proposed above vastly reduce the threat surface of this attack (as many major networks peer with BGP monitoring services), there is still a chance for an adversary to launch the attack at networks that do not peer with BGP monitoring services. Specifically, imagine the path between a traffic source and a victim AS pass through a transit provider T that does not peer with any BGP monitoring services. An adversary could still use the NO_EXPORT community to infect T with a malicious route and since T is not peering with a BGP monitoring service, the attack would remain stealthy. To this end, it is important that BGP monitoring services continue to expand the number of peers they use and attempt to peer with as many transit ASes as possible.

7.4 Use of RPKI

RPKI offers a cryptographic database of IP prefix ownership [42], which can be used to filter BGP announcements [32]. In addition to defining prefix ownership, RPKI defines the specific lengths of the IP prefixes that can be announced in BGP [35]. This can potentially prevent our stealthy BGP attack, which is based on a sub-prefix attack (although misuse of the max-length attribute can degrade RPKI's protection against sub-prefix attacks [35]). Limits on BGP prefix length hold even if an adversary bypasses traditional prefix ownership checks by prepending the victim's ASN to its announcement [38]. Thus, proper RPKI use is a strong mitigation for stealthy BGP attacks.

8 Related Work

8.1 Stealthy BGP Attacks

Milolidakis *et al.* studied smart BGP attacks that evaded public route collectors but used a vastly different mechanism to launch their attack (selective announcement to different BGP neighbors) and did not consider BGP communities or utilize any mechanism to prevent ASes that were peering with monitoring services from exporting the malicious route [45]. As these attacks must be limited only to ASes not peering with monitoring services, these attacks can only affect roughly 2% of the Internet in favorable circumstances.

Morillo *et al.* discuss hidden BGP hijacks that are not visible to networks affected by the hijacks because upstream networks perform ROV filtering on the malicious announcements [47]. Our attack similarly affects networks that do not have the malicious route installed, but in our attacks, the adversary intentionally limits the spread of the announcement via the NO_EXPORT community. Furthermore, while hidden hijacks are hidden from some ASes using the malicious route, there is no guarantee that they are hidden from monitoring services. Hidden hijacks involve the adversary announcing a standard, unrestricted, sub-prefix hijack (which usually has near-global visibility in BGP monitoring).

Heng discussed a BGP attack on a real network that evaded monitoring services by being localized to only a single source network (Yahoo mail) [37]. While this attack did affect a strategic target of interest to the adversary, it did not affect a significant portion of the Internet. Further, it can be seen as a real-world example of a smart BGP attack, as discussed by Milolidakis *et al.*. The malicious route used in the attack was not installed by any networks that were peering with monitoring services, and the adversary did not use any communities to control the propagation of its announcement. By use of the NO_EXPORT community, our attack has the potential to affect significantly more networks and remain stealthy.

Birge-Lee *et al.* discuss using BGP communities on equally-specific attacks to both enable BGP interception attacks and limit the spread of BGP attacks [31]. Our work has vastly different means and objectives than the previous work by Birge-Lee *et al.*. The stealthy attacks presented by Birge-Lee *et al.* aimed to affect a single target of interest and **as few other hosts on the Internet as possible**, and this was realized by using BGP communities to shape equally-specific BGP attacks. Furthermore, the ability of these attacks to specifically evade BGP monitoring systems was not studied. Our work has a completely opposite objective (i.e., to affect as much of the Internet as possible, not just a single host) and achieves this using a sub-prefix BGP attack.

Miller *et al.* use the NO_EXPORT community in the context of BGP attacks that exploit traffic blackholing [44]. They do mention the community’s ability to limit announcement spread and the use of the NO_EXPORT community on a sub-prefix BGP attack in conjunction with blackhole communities. However, they do not study the effect on BGP monitoring systems and, most crucially, do not come to the key insight that even ASes which provide data to BGP monitoring systems will inappropriately suppress information on the adversary’s attack when the NO_EXPORT community is attached. Furthermore, using the NO_EXPORT community on blackhole announcements (where it is already a recommended best practice [44]) is significantly different than the use of the NO_EXPORT community on BGP hijacking announcements that actually maliciously deliver traffic to the adversary.

8.2 BGP Detection Systems

Given data from BGP monitoring, several works propose algorithms for detecting BGP attacks [38, 41, 50]. Our attacks’ ability to achieve stealth is *independent* of what detection algorithm is used for processing BGP data as *none* of the adversary’s malicious BGP announcements are seen by monitoring services. We do not exploit any specific BGP attack detection algorithm behavior but instead achieve stealth by preventing the adversary’s malicious announcements from appearing in the BGP data fed into these algorithms in the first place.

9 Conclusion

This paper challenges the assumption that stealthy BGP hijacks are not globally effective and highlights the need to harden BGP monitoring based on our recommendations. Stealthy BGP attacks can be launched by using the well-known NO_EXPORT community on a sub-prefix BGP hijack. These attacks are highly effective at hijacking vast amounts of Internet traffic while remaining invisible to leading BGP monitoring services. These attacks are also not seen by affected ASes that are forwarding traffic to the adversary. While this is due to the RFC-specified behavior of the well-known NO_EXPORT community, simple changes to router configurations can mitigate this attack. These changes cause ASes peering with BGP monitoring services to still report routes tagged with the NO_EXPORT community. However, even with these changes, it is essential that BGP monitoring services peer with as many transit providers as possible. More broadly, the push for fundamentally more secure interdomain routing is essential as BGP monitoring will always be limited by the number of monitors available and what routes those monitors see. However, as we wait for robust solutions, securing ASes against stealthy BGP hijacks is essential.

10 Ethical Considerations

We took care to ensure that all of the research was ethical. All of our simulations and measurements used publicly-accessible data and did not derive any conclusions about individual users (that could be considered a privacy violation). Our ethical real-world attack was achieved by **attacking ourselves** and using only nodes (including the nodes we used for the adversary) that were authorized to announce our IP prefix. We obtained an IP prefix specifically set aside for network research from our affiliated institution Princeton University (that ran no real network services). Authorization for the use and BGP announcements of this prefix was given in a Letter of Authorization (LoA) signed by the relevant office at Princeton University. This LoA, as well as the Internet Routing Registry (IRR) entries for this prefix, contained all origin ASes used to originate the prefix in our experiments and, based on the LoA and the IRR entries, all of our upstream allowed our announcements. We only made announcements at a reasonable rate. Furthermore, given the well-known NO_EXPORT community is RFC standardized and SHOULD be supported by all BGP speakers that understand communities, we did not see any risk in sending out a BGP announcement with this community attached.

We also conducted random Internet scanning in this paper, but we made sure this scanning did not pose a volume, security, or privacy threat to the networks being scanned or the providers we used for the scanning. We scanned with a small sample (4096 hosts initially that was reduced to 1k hosts based on which hosts sent responses to our scans) using an extremely small number of packets (roughly 3 ICMP Echo packets) that were sent at a reasonable rate. This small frequency of packets does not pose an undue load, even on dated or

under-provisioned equipment. We did not do any port scanning or scanning for support of protocols besides ICMP Echo, which prevented our scan from looking like an intrusion attempt. Finally, we did not record any information from the hosts that responded other than if they responded and whether their packets reached the victim or adversary during our attack.

Acknowledgments

We would like to thank Randy Bush at RGNNet and the networking team at NJEdge for assisting with our experiments. We would also like to thank the Princeton Office of Information Technology for their support in this project. We thank the reviewers for their helpful feedback and Robert Beverly for shepherding our paper.

References

1. The CAIDA AS relationships dataset. <https://www.caida.org/catalog/datasets/as-relationships/> (2023), <https://www.caida.org/catalog/datasets/as-relationships/>
2. (2024), <https://www.ris.ripe.net/peerlist/all.shtml>
3. (2024), <https://www.routeviews.org/peers/peering-status.html>
4. (2024), <https://mailman.nanog.org/pipermail/nanog/2022-February/217602.html>
5. (2024), <https://mailman.nanog.org/pipermail/nanog/2018-June/096034.html>
6. (2024), <https://mailman.nanog.org/pipermail/nanog/2022-August/220320.html>
7. (2024), <https://ifog.ch/en/ip/ixp-access>
8. (2024), <https://www.bandwidth.co.uk/connectivity/reseller-ports/ams-ix>
9. (2024), <https://netactuate.com/remote-peering/ams-ix/>
10. (2024), <https://community.cisco.com/t5/networking-knowledge-base/bgp-communities-and-no-export-lab-scenario/ta-p/3154194>
11. (2024), <https://community.juniper.net/discussion/no-export-community-question>
12. (2024), <https://www.routeviews.org/routeviews/index.php/peering-request-form/>
13. (2024), <https://www.thousandeyes.com/blog/monitoring-bgp-routes-thousandeyes>
14. (2024), <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-peering-policy/>
15. (2024), <https://rpki-monitor.antd.nist.gov/>
16. (2024), <https://www.gin.ntt.net/looking-glass-landing/>
17. (2024), <https://www.cogentco.com/en/looking-glass>
18. (2024), https://cogentco.com/files/docs/customer_service/guide/global_cogent_customer_user_guide.pdf
19. (2024), https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/crosswork-network-automation/b_cisco-crosswork-cloud-user-guide/m_configure-peers-external-routing-analytics.html#Cisco_Concept.dita_abaecc2a-8423-40b3-808f-77b789e87c2d
20. (2024), https://kb.kentik.com/v0/Bd01.htm#Bd01-Router_BGP_Considerations
21. (2024), <https://medium.com/thousandeyes-engineering/thousandeyes-bgp-monitors-fac4d272609c>
22. (2024), <https://asrank.caida.org/>

Global BGP Attacks that Evade Route Monitoring

23. (2024), https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9600/software/release/17-6/configuration_guide/rtng/b_176_rtng_9600_cg/m9_176_rt_bgp_monitoring_protocol_cg.pdf
24. (2024), <https://www.juniper.net/documentation/us/en/software/junos/cli-reference/topics/ref/statement/bmp-edit-routing-options.html>
25. (2024), <https://www.openbmp.org/>
26. Apostolaki, M., Zohar, A., Vanbever, L.: Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. In: IEEE Symposium on Security and Privacy (S&P) (2017)
27. Bayer, D.: Visibility of Prefix Lengths in IPv4 and IPv6. <https://labs.ripe.net/author/dbayer/visibility-of-prefix-lengths-in-ipv4-and-ipv6/> (2010)
28. Birge-Lee, H.: Attackers exploit fundamental flaw in the web's security to steal \$2 million in cryptocurrency. <https://freedom-to-tinker.com/2022/03/09/attackers-exploit-fundamental-flaw-in-the-webs-security-to-steal-2-million-in-cryptocurrency/> (2022)
29. Birge-Lee, H., Sun, Y., Edmundson, A., Rexford, J., Mittal, P.: Bamboozling Certificate Authorities with BGP. In: USENIX Security Symposium. pp. 833–849. USENIX Association, Baltimore, MD (2018), <https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee>
30. Birge-Lee, H., Sun, Y., Edmundson, A., Rexford, J., Mittal, P.: Bamboozling certificate authorities with BGP. In: USENIX Security Symposium (2018)
31. Birge-Lee, H., Wang, L., Rexford, J., Mittal, P.: SICO: Surgical Interception Attacks by Manipulating BGP Communities. In: ACM SIGSAC Conference on Computer and Communications Security (CCS) (2019). <https://doi.org/10.1145/3319535.3363197>, <https://doi.org/10.1145/3319535.3363197>
32. Bush, R.: Origin validation operation based on the resource public key infrastructure (RPKI). RFC 7115, RFC Editor (January 2014)
33. Chandra, R., Traina, P., Li, T.: BGP communities attribute. RFC 1997, RFC Editor (August 1996)
34. Gao, L., Rexford, J.: Stable Internet Routing without Global Coordination. IEEE/ACM Transactions on Networking **9**(6), 681–692 (2001)
35. Gilad, Y., Sagga, O., Goldberg, S.: Maxlength considered harmful to the RPKI. In: International Conference on Emerging Networking EXperiments and Technologies. p. 101–107. CoNEXT, Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3143361.3143363>, <https://doi.org/10.1145/3143361.3143363>
36. Goodin, D.: How 3ve's BGP Hijackers Eluded the Internet – and Made \$29M. <https://arstechnica.com/information-technology/2018/12/how-3ves-bgp-hijackers-eluded-the-internet-and-made-29m/> (2018)
37. Heng, L.: Invisible hijacking a case study of hijacking millions of ip address invisibly (2016), https://ripe72.ripe.net/presentations/45-Invisible_Hijacking.pdf
38. Holterbach, T., Alfroy, T., Phokeer, A.D., Dainotti, A., Pelsser, C.: A system to detect forged-origin hijacks. In: USENIX Symposium on Networked Systems Design and Implementation. USENIX Association (2024)
39. Kacherginsky, P.: Celer Bridge incident analysis. <https://www.coinbase.com/blog/celer-bridge-incident-analysis> (2022)
40. Krenc, T., Luckie, M., Marder, A., claffy, k.: Coarse-grained inference of BGP community intent. In: ACM Internet Measurement Conference. p. 66–72. IMC, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3618257.3624838>, <https://doi.org/10.1145/3618257.3624838>

41. Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., Zhang, L.: PHAS: A prefix hijack alert system. In: USENIX Security Symposium. USENIX Association, Vancouver, B.C. Canada (Jul 2006), <https://www.usenix.org/conference/15th-usenix-security-symposium/phas-prefix-hijack-alert-system>
42. Lepinski, M., Kent, S.: An infrastructure to support secure internet routing. RFC 6480, RFC Editor (February 2012), <http://www.rfc-editor.org/rfc/rfc6480.txt>, <http://www.rfc-editor.org/rfc/rfc6480.txt>
43. Li, W., Lin, Z., Khan, M.I.A., Aben, E., Fontugne, R., Phokeer, A., Chung, T.: RoVista: Measuring and Understanding the Route Origin Validation (ROV) in RPKI. In: ACM Internet Measurement Conference. Montreal, Canada (October 2023)
44. Miller, L., Pelsser, C.: A taxonomy of attacks using BGP blackholing. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) Computer Security – ESORICS 2019. pp. 107–127. Springer International Publishing, Cham (2019)
45. Milolidakis, A., Bühler, T., Chiesa, M., Vanbever, L., Vissicchio, S.: Poster: Smart BGP hijacks that evade public route collectors. p. 2 (2021)
46. Milolidakis, A., Bühler, T., Wang, K., Chiesa, M., Vanbever, L., Vissicchio, S.: On the effectiveness of BGP hijackers that evade public route collectors. *IEEE Access* **11**, 31092–31124 (2023). <https://doi.org/10.1109/ACCESS.2023.3261128>
47. Morillo, R., Furuness, J., Morris, C., Breslin, J., Herzberg, A., Wang, B.: ROV++: Improved Deployable Defense against BGP Hijacking. In: Network and Distributed System Security Symposium (NDSS) (2021)
48. NLnet Labs: Routinator 0.14.0. <https://routinator.docs.nlnetlabs.nl/en/stable/> (2024)
49. Scudder, J., Fernando, R., Stuart, S.: BGP monitoring protocol (BMP). RFC 7854, RFC Editor (June 2016)
50. Sermpezis, P., Kotronis, V., Gigis, P., Dimitropoulos, X., Cicalese, D., King, A., Dainotti, A.: ARTEMIS: Neutralizing BGP Hijacking within a Minute. *IEEE/ACM Transactions on Networking* **26**(6), 2471–2486 (Dec 2018). <https://doi.org/https://doi.org/10.1109/TNET.2018.2869798>
51. Siddiqui, A.: BGPsec – A reality now. <https://www.internetsociety.org/blog/2017/10/bgpsec-reality-now/> (2017)
52. Snijders, J.: Cogent RPKI invalid filtering. <https://mailman.nanog.org/pipermail/nanog/2021-April/213346.html> (2021)
53. Streibelt, F., Lichtblau, F., Beverly, R., Feldmann, A., Pelsser, C., Smaragdakis, G., Bush, R.: BGP communities: Even more worms in the routing can. In: ACM Internet Measurement Conference. p. 279–292. IMC, Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3278532.3278557>, <https://doi.org/10.1145/3278532.3278557>
54. Sun, Y., Apostolaki, M., Birge-Lee, H., Vanbever, L., Rexford, J., Chiang, M., Mittal, P.: Securing Internet Applications from Routing Attacks. *Communications of the ACM* **64**(6), 86–96 (2021). <https://doi.org/10.1145/3429775>, <https://doi.org/10.1145/3429775>
55. Sun, Y., Edmundson, A., Vanbever, L., Li, O., Rexford, J., Chiang, M., Mittal, P.: RAPTOR: Routing Attacks on Privacy in Tor. In: USENIX Security Symposium (2015)
56. Upadhaya, G.R.: BGP Best Practices for ISPs. <https://conference.apnic.net/22/docs/tut-routing-pres-bgp-bcp.pdf> (2022)

A Config Examples

This is an example of pseudocode for a router configuration that rewrites the NO_EXPORT community at ingress to an AS-specific community that matches the original NO_EXPORT behavior (except in the case of sessions with BGP monitoring services).

```

filter neighbor_in {
    if NO_EXPORT in bgp_communities {
        bgp_communities.remove(NO_EXPORT);
        bgp_communities.add(ASN:123);
    }
    # Similar blocks need to also be include
    # for NO_ADVERTISE and NO_EXPORT_SUBCONFED
    ...
    accept;
}

filter neighbor_out {
    if ASN:123 in bgp_communities {
        reject;
    }
    accept;
}

protocol bgp neighbor {
    ...
    import filter neighbor_in;
    export filter neighbor_out;
}

protocol bgp bgp_monitoring_service {
    ...
    import none;
    export all;
}

```

Another potential configuration change to mitigate this attack is to run BGP sessions with monitoring services as Internal BGP (iBGP) sessions instead of eBGP sessions. This way, routers would no longer consider BGP sessions with monitoring services as crossing an AS boundary, and thus, the export restrictions of the NO_EXPORT community would not apply. There still may be some attack surface left open by the RFC-standardized NO_ADVERTISE community [33] (which limits exporting to any other routers even in the same AS), but these attacks are likely significantly less effective. If an adversary were to use the

H. Birge-Lee et al.

NO_ADVERTISE community to launch an attack its malicious announcement would be localized to only a single router.