

Be Careful of Your Neighbors: Injected Sub-prefix Hijacking Invisible to Public Monitors

Jun He*, Yahui Li*, Han Zhang*[†], Ming Wang*, Changqing An*, and Jilong Wang*[†]

*Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China

[†]Zhongguancun Laboratory, Beijing, China

Emails: {he-j21, m-wang20}@mails.tsinghua.edu.cn, liyahui_ego@126.com, zhhan@tsinghua.edu.cn, {acq, wjl}@cernet.edu.cn

Abstract—Prefix hijackings have always been a significant security issue in BGP and have continued to occur in recent years. Detecting prefix hijackings is a vital part of defending against them. Most detection approaches mainly rely on the feed from the monitors of public route collector infrastructures. We propose an injected sub-prefix hijacking that utilizes the BGP communities attribute and AS path poisoning to control the propagation of invalid sub-prefix routes. This attack only pollutes neighboring ASes, thus guaranteeing the invisibility to monitors. Then the attacker can stealthily hijack traffic passing through the polluted ASes. Through extensive simulations, we show that this attack has an enormous impact and propose the crucial indicator affecting the attacker's capability. Finally, we demonstrate that existing defenses are difficult to handle this attack and then propose several defense strategies against it.

Index Terms—BGP, Security, Hijacking, Invisibility, BGP Communities, AS Path Poisoning

I. INTRODUCTION

Border Gateway Protocol (BGP), as the standard inter-domain routing protocol in the Internet, is used by Autonomous Systems (ASes) to transmit routing information about IP prefixes. However, due to the lack of authentication, BGP is vulnerable to invalid routes announced by malicious ASes. An attacker can forge invalid routes about certain prefixes and then hijack traffic sent by other ASes. This phenomenon is known as BGP prefix hijacking. Moreover, prefix hijacking is further divided into exact prefix hijacking and sub-prefix hijacking, depending on whether the attacker hijacks an existing prefix or its sub-prefix.

BGP hijacking can cause great harm to the Internet, so many works have studied related defenses, which can be divided into two main categories. One class of defenses is to deploy preventive security mechanisms, such as ROV [1], BGPsec [2], and Path-end [3]. However, the cost of deploying them is high, and they depend on the coordination between global ASes, so they cannot be fully deployed in a short time. Another class of defenses relies on hijacking detection and manual mitigation after detection. Detection approaches are mainly divided into control-plane, data-plane, and hybrid approaches.

This work is partially supported by the National Key Research and Development Program (No. 2020YFE0200500) and the National Science Foundation (No. 62102020, No. 62002009).

Yahui Li and Han Zhang are the co-corresponding authors.

Among them, data-plane approaches [4], [5] require a lot of detection overhead, and the filtering standards are difficult to determine, resulting in low detection accuracy. Control-plane and hybrid approaches [6]–[9] are more effective and accurate, but they mainly rely on public route collector infrastructures, namely RouteViews [10] and RIPE RIS [11]. These two platforms establish collectors to peer with some ASes and collect BGP update messages from these ASes. The routing data are then archived for public access. These ASes that provide their routing information are called monitor ASes, or *monitors* for short. Control-plane and hybrid approaches need to use these public data to infer hijacking, so these detection methods are largely affected by the visibility of the hijacking to public monitors.

Some works [12]–[14] have considered how to control the impact and visibility of prefix hijacking, but their proposed attack methods rely on strict attack premises (*e.g.*, special BGP communities or preknowledge of routing policies). It is difficult for a realistic attacker to ensure the invisibility of the attack through these methods. Besides, the attack impact of these methods is minimal. Inspired by the hidden hijacking phenomenon in BGP, we consider how to control the visibility of sub-prefix hijacking in a feasible way.

Specifically, we propose injected sub-prefix hijacking. The attacker injects invalid sub-prefix routes into neighbor ASes and prevents these invalid routes from spreading out through elaborate BGP attributes. This attack method only pollutes specific neighbors, thus guaranteeing its invisibility to other ASes, including monitors. Then, all traffic destined for this sub-prefix will be hijacked as long as it passes through the polluted ASes. So this attack still has a significant impact. Most importantly, this attack can bypass most existing detection approaches and preventive security mechanisms. Overall, we make the following contributions:

- We propose a new injected sub-prefix hijacking that is invisible to public monitors.
- We evaluate the attack capabilities of different attackers through extensive simulations, demonstrating that such attacks can have a large impact.
- We analyze the resistance of our attack to existing defenses and propose some defense strategies for this attack.

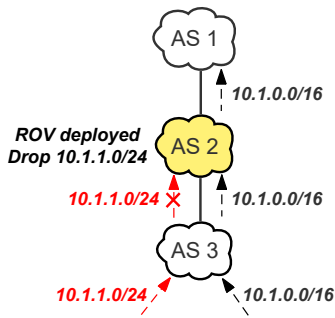


Fig. 1. Illustration of hidden hijacking.

II. BACKGROUND

A. RPKI

RPKI [15] is a public key infrastructure established to secure inter-domain routing. A prefix owner can register several ROA records for the prefix, and each ROA record indicates which AS has the right to announce this prefix. Moreover, each ROA record includes an additional field called *max_length*. All ROA records are stored in centralized databases.

ROV is the corresponding validation mechanism. An AS that adopts ROV will use the ROA records in the databases to validate whether the routing information received from the neighbors is legal. An update message whose origin AS does not match or whose prefix length exceeds *max_length* will be generally discarded or demoted.

B. Hidden Hijacking

Some works [16], [17] noticed the existence of the hidden hijacking phenomenon, which reflects the inconsistency between the control plane and the data plane. Theoretically, the impact scope of a sub-prefix hijacking is the entire Internet. That is, the invalid sub-prefix route will pollute all ASes. In fact, when the invalid route is being propagated, some ASes may not learn it due to the routing policies determined by AS relationship or the existence of other filtering mechanisms (e.g., ROV).

As shown in Fig. 1, suppose there is an invalid route of *10.1.1.0/24* and a legitimate route of *10.1.0.0/16*. AS 3 learns these two routes and propagates them to AS 2. However, if AS 2 deploys ROV and can filter the invalid route, then both AS 2 and AS 1 can only learn the legitimate route of */16*. Although AS 1 is not polluted, its traffic can still be hijacked. When it sends traffic to the */16* legal prefix, its traffic may pass through AS 3. Then the traffic will be forwarded to the attacker due to the longest prefix matching principle. However, AS 1 could not have foreseen this happening. From its point of view, this is a hidden hijacking.

C. BGP Communities

BGP communities are an optional transitive attribute defined by RFC 1997 [18], which is mainly used for recording information, delivering information, or behavior control. NO_EXPORT is a standardized well-known BGP community, meaning that any community-attribute-aware BGP speaker must comply.

When an AS receives an update message with NO_EXPORT label, it must not propagate this routing information to other ASes outside the BGP confederation.

D. AS Path Poisoning

There is a loop prevention mechanism in BGP. When an AS receives an update message containing its own ASN in the AS-path field, it will discard this message. AS path poisoning refers to prepending some specific ASNs to the AS-path field when advertising update messages. Therefore, these routing messages will be discarded after they reach the corresponding ASes. In this way, one can prevent routes from being learned by some specific ASes.

III. ATTACK METHOD

Although the existence of ROV mechanism can cause hidden hijackings, it is difficult for an attacker to do this intentionally. First, the attacker AS needs to know the ROV deployment of all ASes. Only in this way can it formulate an announcement method so that invalid routes will not be propagated to monitors. In addition, the target prefixes that can be hiddenly hijacked by exploiting ROV mechanism are limited to ROA prefixes (referring to prefixes covered by existing ROA records, currently accounts for about 40% of global routable prefixes according to NIST RPKI Monitor project [19]). In a word, due to the uncontrollability and target limitations, the impact of this exploit is small.

In this section, we propose a stronger attack called injected sub-prefix hijacking and describe its two implementations. Then we consider possible extensions of our attack method, by which the attacker can achieve more powerful attacks.

A. Injected Sub-prefix Hijacking

In exact prefix hijacking, only when an AS has learned the invalid route will its traffic be hijacked. This is because traffic will always propagate along the learned optimal route if no more specific sub-prefix routes exist at intermediate ASes. So exact prefix hijacking is essentially a route competition, where the attacker AS aims to get more ASes to choose the invalid route as the optimal route. Since many monitor ASes are large transit ASes, it is difficult to bypass these monitor nodes if more ASes are expected to learn the invalid route.

However, we can learn from the hidden hijacking phenomenon that sub-prefix hijacking may cause inconsistency between the control plane and the data plane. If an AS learns an invalid sub-prefix route, the traffic passing through it will be affected, even if the source AS of the traffic does not receive the invalid route. So if an attacker can control the propagation of invalid sub-prefix routes, it can achieve a large impact with less visibility.

We consider an injected sub-prefix hijacking. When the attacker announces invalid sub-prefix routes to its neighbors, it uses certain tricks to control them not to propagate outward. Therefore, monitor ASes cannot learn these invalid routes. But when the traffic of other ASes passes through the injected neighbors, if the destination IP address belongs to the invalid

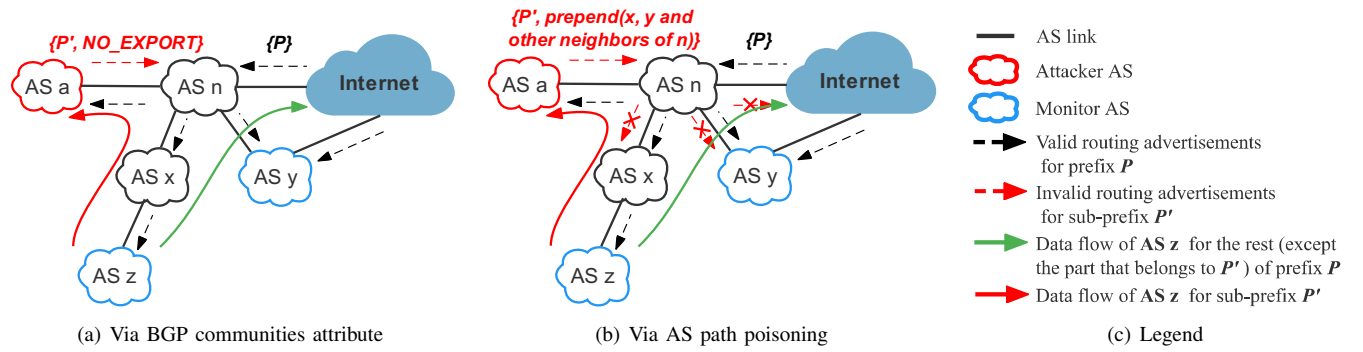


Fig. 2. Illustration of injected sub-prefix hijacking.

sub-prefix, it will still be forwarded to the attacker due to the longest prefix matching principle.

B. Via BGP Communities Attribute

To control the propagation of invalid sub-prefix routes, the first implementation is to utilize the BGP communities attribute mentioned in II-C.

Consider a simple attack example shown in Fig. 2(a), where $AS\ a$ represents the attacker and $AS\ n$ represents the neighbor AS that will be injected with an invalid sub-prefix route (note that $AS\ n$ is not the legitimate origin of the target prefix, unlike the victims mentioned in the simulations below). The blue frame indicates that the AS is a public monitor, so the attacker needs to ensure that these ASes cannot learn invalid routes. P represents a legal routable prefix propagated from other ASes in the Internet, and all ASes in the figure can learn legal routes about P .

Attacker $AS\ a$ now advertises an invalid route for the more specific sub-prefix P' of P to its neighbor $AS\ n$. Unlike regular BGP update messages, it adds the `NO_EXPORT` label to the communities field. In this way, when $AS\ n$ learns this invalid route, it will not advertise this route to its other neighbors. As a result, both routes for prefix P and sub-prefix P' exist in the routing table of $AS\ n$, while the routing tables of other ASes only contain routes for prefix P . Neither of the monitor ASes has learned this invalid route, so the attack is successfully concealed.

Next, we analyze the current behavior of the data plane. Take $AS\ z$ as an example, it only learns the route about P , and the AS-path is $\{x, n, \dots\}$. Therefore, the traffic sent by $AS\ z$ to prefix P will pass through $AS\ x$ and $AS\ n$ in sequence. But when the traffic reaches $AS\ n$, since $AS\ n$ has the route for a more specific prefix P' , the forwarding path of the traffic may be diverted. Specifically, if the destination IP address of a packet belongs to P' , then $AS\ n$ will forward it to the attacker $AS\ a$. In fact, not only traffic originating from $AS\ z$, but any related traffic passing through $AS\ n$ will be hijacked, such as traffic from $AS\ x$, $AS\ n$ or $AS\ y$ (if $AS\ y$ chooses to route through $AS\ n$). Therefore, $AS\ a$ successfully implements injected sub-prefix hijacking by manipulating the BGP communities field.

C. Via AS Path Poisoning

We already know from the previous introduction (II-D) that AS path poisoning can prevent certain ASes from learning the route. Inspired by it, another attack implementation method proposed by us is to use AS path poisoning. The attack process is shown in Fig. 2(b).

When attacker $AS\ a$ sends an invalid sub-prefix route to $AS\ n$, it prepends ASNs of all neighbors of $AS\ n$ ($AS\ x$, $AS\ y$, etc.) to the AS-path field. The route propagation process at this time differs from the situation in III-B. In the previous implementation, $AS\ n$ will not advertise the invalid route after learning it. But now, $AS\ n$ will add its own ASN to the AS-path field and continue to advertise it to its neighbors. However, the neighbors of $AS\ n$ will discard this invalid route after receiving it because their ASNs are already contained in the AS-path field.

Although the process is different, from the results, only $AS\ n$ has learned the invalid sub-prefix route. The data plane behaves the same as the previous attack implementation. $AS\ a$ still achieves the purpose of hijacking while ensuring the invisibility of attack to public monitors.

D. Attack Extensions

The injected sub-prefix hijacking can be combined with other methods to achieve stronger attack capability.

Most attackers simply drop the traffic after hijacking it, similar to BGP blackholing. Although this manipulation method is more destructive, it is easy to find clues because the attack affects traffic reachability. It may be found that there is a problem with traffic propagation, but it is not sure whether the problem is caused by hijacking. Further, the previous work [20] considered how to conduct a man-in-the-middle hijacking, only intercepting the traffic and then sending it back to the legitimate origin AS of the target prefix. Our proposed attack can easily achieve this extension. The attacker needs to ensure that the legal route it learned before is not affected, thus only polluting those neighbors that the legal route does not pass through.

As shown in Fig. 3, the attacker $AS\ a$ learns legitimate routes about the prefix P from $AS\ n$ and $AS\ m$. Using our attack method, if $AS\ a$ wants to influence as many ASes as possible, it can inject the invalid sub-prefix route into all its

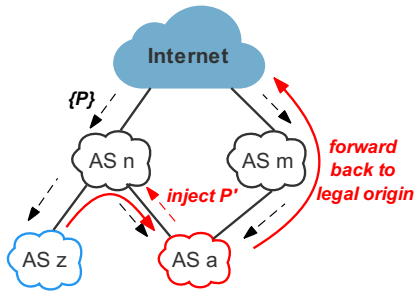


Fig. 3. Illustration of extended interception attack.

non-monitor neighbors. And at this time, $AS\ a$ chooses to inject the invalid sub-prefix P' into $AS\ n$, but not into $AS\ m$. Similar to the previous hijacking scenario, the traffic of $AS\ z$ will be forwarded to attacker $AS\ a$ after passing through $AS\ n$. But since a legitimate path is reserved in advance, attacker $AS\ a$ can send $AS\ z$'s traffic back to the legal origin of prefix P through $AS\ m$ after hijacking it.

In addition, attackers can forge the origin of invalid sub-prefixes or critical parts in the AS-path field to bypass some defense mechanisms, and we will detail this in section VI.

IV. FEASIBILITY OF ATTACK

In this section, we conduct related routing experiments to better understand the attack process and prove its feasibility. Furthermore, we analyze the challenges of implementing the attack and possible obstacles.

A. Routing Experiments

We installed Quagga routing software [21] on five hosts to run them as BGP speaker routers. We use each router to represent an independent AS, and the topology of the experiments is shown in Fig. 4. $AS\ 4$ announces the prefix $10.1.0.0/16$ as the legitimate origin, and attacker $AS\ 5$ tries to inject an invalid route about $10.1.1.0/24$ into $AS\ 3$.

The first implementation is to set the NO_EXPORT label, as described before. Another way is through AS path poisoning, and the AS-path field can be constructed as $\{5, 4, 2, 5\}$. We verified the result of these two attack implementations by viewing each router's routing table and active probing. Specifically, we found that only $AS\ 3$ has learned the invalid route, and there is no route about $10.1.1.0/24$ in the routing tables of other ASes. Then, we verified traffic diversion using *traceroute* command on the router of $AS\ 1$. When we *traceroute* to $10.1.2.1$, the packets will pass through the router interfaces corresponding to the legal AS-path $\{2, 3, 4\}$. And when we *traceroute* to $10.1.1.1$, we find that the packets are forwarded to the router interface of $AS\ 5$ (i.e., being hijacked).

B. Challenges

To implement this attack via BGP communities, the attacker needs to ensure that invalid prefixes are not filtered by neighbors. In addition to deploying preventive security mechanisms, some provider ASes filter the update messages

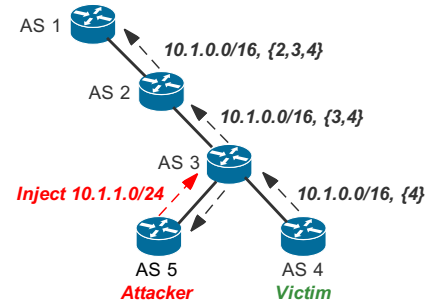


Fig. 4. Topology of our conducted routing experiments.

sent by their own customer ASes, accepting only those prefixes that have been publicly registered or assigned to customers by themselves. To the advantage of attackers, prefix filtering mechanisms are not widely deployed at present. As reported by MANRS [22], real hijackings are still happening in recent years. Furthermore, Cisco's BGPStream [23] reports thousands of possible hijackings on average each year.

If the attack is carried out via AS path poisoning, in addition to prefix filtering mechanisms, it may also be affected by some AS-path filtering mechanisms. For example, tier-1 ASes and many large ISPs deploy Peerlock [24] to filter out apparently invalid AS-paths. If a transit provider receives a routing message from a customer whose AS-path has a hop that is a tier-1 AS, then this is most likely a hijacking or route leak. Because it means that this customer is providing transit service for the tier-1 AS, which is very unreasonable. Additionally, an update message is likely to be discarded when its AS-path field is too long. If there are many neighbors of the AS we want to pollute, then we need to prepend many ASNs in the AS-path, and so the greater the possibility of being filtered. By comparison, we believe the implementation method via BGP communities has stronger applicability and realistic feasibility.

V. IMPACT EVALUATION

In this section, we evaluate the impact of our attack through extensive simulations. We implement attacks via BGP communities, simulating both normal hijackings and extended interception attacks. First, we introduce the simulation method we use and how a victim AS (i.e., the legitimate origin of the target prefix) is selected. Then we calculate how many other ASes can be affected by a selected attacker, excluding the ASes that have already been affected (the routes of some ASes pass through the attacker before the attack, e.g., some customer ASes of the attacker).

Specifically, we analyze the attack impact mainly from the following three perspectives: (1) hijacking capabilities and interception capabilities (collectively referred to as attack capabilities) of different attackers (2) the indicator that best characterizes the hijacking capability of an AS.

A. Preliminary of Evaluation

Simulation Method. To simulate route propagation and path selection in the Internet routing system, we adopt methods

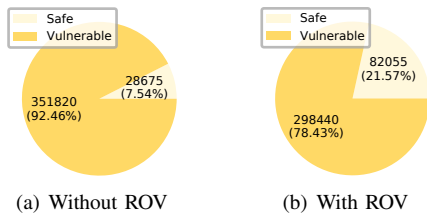


Fig. 5. Global routing prefixes (less specific than /24) that may be hijacked by sub-prefix hijacking.

commonly used in previous works (*e.g.*, [3], [9]). We use CAIDA's AS-relationship dataset [25] to establish the inter-domain topology. And we use the classic Gao-Rexford model [26] to simulate the routing policies of all ASes. Besides, we extracted information about the currently deployed monitors from RouteViews and RIPE RIS websites on September 23, 2022. From the number of RIB table entries reported by different monitor ASes, we identified a total of 364 full-feed monitors to reflect the monitor deployment in our simulations. **Selection of Victim ASes.** A target prefix can be attacked by sub-prefix hijacking if and only if its prefix length is less than /24, and there is at least one /24 sub-prefix not announced. On September 25, 2022, we extracted RIB tables recorded by route collector *amsix* of RouteViews to analyze the currently announced prefixes in the global routing table. Fig. 5(a) depicts the relevant statistics. There are 380,495 prefixes less specific than /24 in total, among which 351,820 (92.46%) have at least one /24 sub-prefix not announced (*i.e.*, may be vulnerable to sub-prefix hijacking). We randomly select 1000 ASes from the origin ASes of these vulnerable prefixes as victims. In subsequent simulations, they serve as legitimate origins of target prefixes.

B. Attack Capabilities at Different AS Levels

Because this attack method is implemented by affecting neighbors, there is a large gap in attack capabilities between ASes at different levels. We extracted the customer cone sizes of all ASes from CAIDA's AS Rank [27] and combined them with CAIDA's AS-relationship dataset for classification. We adopt the widely used UCLA classification [28], which divides different ASes into four categories: (1) *tier-1 ASes*: have no provider and peer with each other (2) *large ISPs*: have more than 50 customer cone ASes (3) *small ISPs*: customer cone size is between 5 and 50 (4) *stub ASes*: have fewer than 5 customer cone ASes.

Regarding the selection of attacker ASes, we selected all of the 17 tier-1 ASes, and randomly picked 100 large ISPs and 1000 small ISPs. Note that we did not choose stub ASes because they generally have little impact using this attack method, which will be explained in V-D. Therefore, we selected a total of 1117 attackers. For each selected victim AS, we simulate an announcement and calculate how many other ASes can be hijacked or intercepted by different attacker ASes. In total, we run 1000×1117 hijacking simulations to measure the attack capabilities of different attackers. Then we calculate the

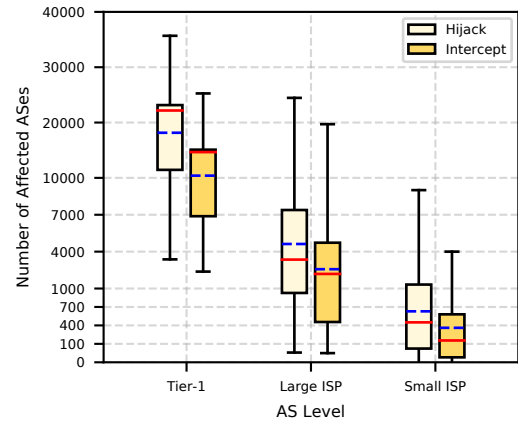


Fig. 6. Impact scope of attacker ASes at different levels (the blue dashed line represents the mean, and the red solid line represents the median).

average affected number of each attacker on 1000 victims, and the boxplot of the results is shown in Fig. 6.

While Sermpezis *et al.* [9] showed that prefix hijacking affecting $> 2\%$ of global ASes is always visible to public monitors, this threshold can be easily overridden using our injected sub-prefix hijacking. When these attacks are completely invisible to public monitors, tier-1 ASes, large ISPs, and small ISPs can hijack an average of 18154 (24.34%), 4623 (6.20%), and 629 ($< 1\%$) global ASes respectively. It shows that there are currently many ASes that can stealthily attack a wide range of global routing system. Consider more powerful interception attacks, tier-1 ASes, large ISPs, and small ISPs can intercept an average of 10403 (13.95%), 2571 (3.45%), and 361 ($< 1\%$) global ASes respectively. It should be noted that the average value of each attacker we calculated represents the general attack capability of the attacker, which means that it can achieve such an attack impact for a variety of different victims. In fact, for some specific victims, one attacker can often show influence far beyond its general attack capability.

C. Indicators of Attack Capability

The classic classification we considered in V-B is mainly determined by AS customer cone size. However, we can notice from Fig. 6 that some low-level ASes have stronger attack capabilities than high-level ASes. So we try to find which indicator is better for measuring the attack capability.

In addition to customer cone size, it is not difficult to find that AS degree greatly influences the general attack capability. Besides, we define an indicator called *adjacent cone size*. For an attacker, adjacent cone size represents the sum of the customer cone sizes of its non-monitor neighbor ASes. Based on the cone size rankings of the attackers selected in V-B, we re-select the corresponding attackers according to the rankings of AS degree and AS adjacent cone size. For example, suppose we previously selected an attacker with the i -th largest customer cone size. In that case, we select an attacker with the i -th largest AS degree and an attacker with the i -th largest adjacent cone size for comparison. In a word, we select 1117 attacker ASes

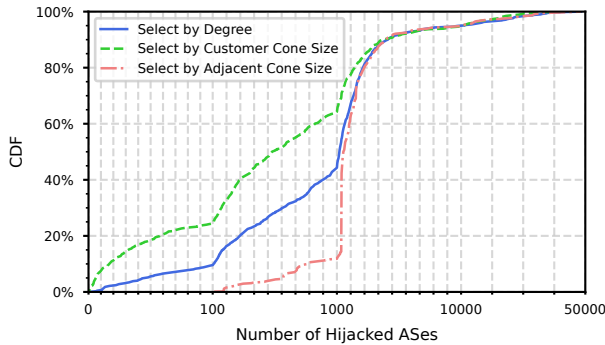


Fig. 7. CDF of general hijacking capabilities of three selected attacker sets.

for each indicator with the same rankings respectively, thus constructing three attacker sets to verify their differences in attack capabilities.

The CDF of general hijacking capabilities of different attacker sets is shown in Fig. 7. We can see that $> 85\%$ of the attackers selected by adjacent cone size can hijack more than 1000 other ASes on average, while the other two attacker sets are $< 40\%$ and $< 60\%$ respectively. Therefore, we believe that the adjacent cone size can better characterize the general attack capability of an AS.

D. Limitations

We admit that our attack method has certain shortcomings. On the one hand, the general attack capability of a stub AS is very weak because of its small adjacent cone size. But it can be found that in an injected sub-prefix hijacking, the attack impact strongly correlates with the distance between the attacker and the victim. If an attacker wants to hijack the prefixes announced by surrounding ASes, it can do more harm. Besides, an attacker can connect to more high-level ASes for stronger attack capability.

On the other hand, even if such an attack can stealthily affect more ASes, it hijacks only a sub-prefix of the target prefix. However, we think that this attack method may be used to attack specific IP targets to compromise some important servers or hosts.

VI. DISCUSSION

In this section, we discuss the resistance of our attack to various detection approaches and preventive security mechanisms. Then we propose some defense strategies against it.

A. Resistance Analysis

Control-plane and Hybrid Approaches. Typical control-plane approaches (e.g., [6], [7]), the state-of-the-art third-party detection Argus [8], and even the self-detection framework ARTEMIS [9], rely on the routing updates provided by public route collector infrastructures. When they find multiple origin ASes (MOAS) conflicts in routing updates or abnormal links in AS-paths, further detection procedures will be triggered. Whereas our proposed attack ensures the invisibility to monitors, thus bypassing these detection approaches.

Data-plane Approaches. Like iSPY [4] and the approach proposed by Zheng *et al.* [5], most data-plane detection approaches inherently rely on the reachability information provided by a single IP address in the target prefix. But the selected IP address may not belong to the hijacked sub-prefix, and the active detection overhead is high, so these data-plane approaches are not suitable for detecting sub-prefix hijacking.

Preventive Security Mechanisms. More and more ASes begin to deploy ROV, which has a certain defensive effect on sub-prefix hijackings. Exact prefix hijacking can bypass ROV by forging the origin AS, while sub-prefix hijacking may be filtered by *max_length* even if the origin AS can be forged. But as shown in Fig. 5(b), assuming that all neighbor ASes deploy ROV, we still find that 298440 (78.43%) non-/24 prefixes can be hijacked by analyzing the existing ROA records of five major RIRs on September 1, 2022. In addition to ROV, path-end validation [3] can be bypassed by forging the first two hops in the AS-path. Although BGPsec [2] provides the strongest security guarantee, it is difficult for many ASes to deploy BGPsec in a short time due to its complexity and cost.

B. Defense Strategies

Stricter Filtering. If the AS finds that a neighbor's update message attempts to influence its propagation (through AS path poisoning or BGP communities) and none of these public monitors have a route for this prefix, it should verify that the neighbor is the legitimate owner of the prefix. Specifically, it can query the Internet Routing Registries (IRRs) and ROA databases to see if there are corresponding entries. If the information does not match, contact the legitimate owner for validation or drop the suspicious update message.

Stricter ROAs. Because of the special role of ROV in defending against sub-prefix hijacking, the deployment of ROV should be accelerated. Moreover, ASes may consider using stricter ROA records [29]. That is, the *max_length* of a ROA should not exceed the length of the existing legal prefix.

More Monitors. Deploying more monitors can somewhat reduce the general attack capabilities of some attackers. However, it should be noted that as long as the attacker's neighbors do not deploy monitors, there is no impact on the attacker's capability.

VII. RELATED WORK

Stealthy Hijacking. McArthur *et al.* [12] first considered reducing the impact of prefix hijacking to evade detection but did not consider how to bypass public monitors. Birge-Lee *et al.* [13] studied how to use BGP communities to achieve propagation control. However, their method overly relies on the support of each transit AS for special communities and the propagation of communities. Milolidakis [14] tried to use AS path poisoning and AS path prepending to affect the propagation of routes, thus making exact prefix hijacking invisible to monitors. However, this method relies on the preknowledge of the routing policies of each AS. So it is difficult for a realistic attacker to launch such an attack. Secondly, this attack method has a negligible impact scope, only 1.65% of the realistic attackers in his simulations affected more than 2% of the Internet.

Hijacking Defenses. In addition to the defenses we have mentioned before [1]–[9], Morillo *et al.* [16] proposed ROV++. It improves the security of ROV and reduces possible hidden hijackings, but only protects the ROA prefixes. Schlamp *et al.* [30] proposed a hybrid filter approach HEAP to evaluate hijacking alerts, but our attack method does not trigger alerts.

Measurements of Defense. Several works have studied how to measure the actual deployment of defense mechanisms in the Internet. The deployment of ROV mechanism was first measured by Glide *et al.* [17], who used publicly available control plane data for heuristic inference. Reuter *et al.* [31] measured ROV deployment using controlled experiments, but the measurement scope was limited. Gray *et al.* [32] and Chen *et al.* [33] measured ROV deployment using Bayesian inference techniques. Furthermore, McDaniel *et al.* [34] measured and inferred the deployment of Peerlock. They found that Peerlock is mostly deployed in the core area of the Internet but rarely deployed in the edge area.

VIII. CONCLUSION

We present a novel sub-prefix hijacking attack that evades observation by public monitors. Moreover, we propose two specific attack implementations and demonstrate their feasibility. Then we evaluate the attack's impact through simulations from different attacker perspectives. Further, we show that existing defense methods cannot defend against this attack, thus requiring organizations and network operators to adopt more robust defense strategies. Overall, our work is the first to study how to control the visibility of sub-prefix hijacking, and the proposed attack method can be easily implemented by a realistic attacker. In the future, we plan to investigate how to extend the scope of the attack and how to improve the existing defense mechanisms to enhance resistance to the attack.

REFERENCES

- [1] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein, "BGP Prefix Origin Validation," RFC 6811, January 2013.
- [2] M. Lepinski (Ed.) and K. Sriram (Ed.), "BGPsec Protocol Specification," RFC 8205, September 2017.
- [3] A. Cohen, Y. Gilad, A. Herzberg, and M. Schapira, "Jumpstarting BGP security with path-end validation," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 342-355.
- [4] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP prefix hijacking on my own," in *Proceedings of the ACM SIGCOMM 2008 conference on Data Communication*, 2008, pp. 327-338.
- [5] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting IP prefix hijacks in real-time," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 277-288, 2007.
- [6] M. Lad *et al.*, "PHAS: A Prefix Hijack Alert System," *USENIX Security symposium*, vol. 1, no. 2, p. 3, 2006.
- [7] J. Li, T. Ehrenkrantz, and P. Elliott, "Buddyguard: A buddy system for fast and reliable detection of IP prefix anomalies," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*, 2012, pp. 1-10.
- [8] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the internet with argus," in *Proceedings of the 2012 Internet Measurement Conference*, 2012, pp. 15-28.
- [9] P. Sermpezis *et al.*, "ARTEMIS: Neutralizing BGP hijacking within a minute," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2471-2486, 2018.
- [10] University of Oregon, "RouteViews Project," <http://www.routeviews.org/routeviews/>, 2022.
- [11] RIPE NCC, "Routing Information Service (RIS)," <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris>, 2022.
- [12] C. McArthur and M. Guirguis, "Stealthy ip prefix hijacking: don't bite off more than you can chew," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, 2009, pp. 1-6.
- [13] H. Birge-Lee, L. Wang, J. Rexford, and P. Mittal, "Sico: Surgical interception attacks by manipulating bgp communities," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 431-448.
- [14] A. Milolidakis, "Understanding the Capabilities of Route Collectors to Observe Stealthy Hijacks: Does adding more monitors or reporting more paths help?," Licentiate dissertation, KTH Royal Institute of Technology, Stockholm, 2022.
- [15] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing," RFC 6480, February 2012.
- [16] R. Morillo *et al.*, "ROV++: Improved Deployable Defense against BGP Hijacking," in *NDSS*, 2021.
- [17] Y. Gilad, A. Cohen, A. Herzberg, M. Schapira, and H. Shulman, "Are we there yet? On RPKI's deployment and security," in *NDSS*, 2017.
- [18] R. Chandra, P. Traina, and T. Li, "BGP Communities Attribute," RFC 1997, August 1996.
- [19] NIST, "RPKI Monitor," <https://rpki-monitor.antd.nist.gov/ROV>, 2022.
- [20] H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 265-276, 2007.
- [21] P. Jakma, "Quagga Routing Software," <https://github.com/Quagga>, 2018.
- [22] MANRS, "MANRS Blog," <https://www.manrs.org/blog/>, 2022.
- [23] Cisco, "BGPSStream," <https://bgpsstream.crosswork.cisco.com/>, 2022.
- [24] J. Snijders, "NTT Peer Locking," http://instituut.net/~job/peerlock_manual.pdf, 2016.
- [25] CAIDA, "AS Relationships Dataset,"
- [26] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 681-692, 2001.
- [27] CAIDA, "AS Rank," <https://asrank.caida.org/>, 2022.
- [28] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (in) completeness of the observed Internet AS-level structure," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 109-122, 2009.
- [29] Y. Gilad, O. Sagga, and S. Goldberg, "Maxlength considered harmful to the rpki," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, 2017, pp. 101-107.
- [30] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack, "HEAP: reliable assessment of BGP hijacking attacks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1849-1861, 2016.
- [31] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt, and M. Wählisch, "Towards a rigorous methodology for measuring adoption of RPKI route validation and filtering," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 1, pp. 19-27, 2018.
- [32] C. Gray *et al.*, "BGP beacons, network tomography, and Bayesian computation to locate route flap damping," in *Proceedings of the ACM Internet Measurement Conference 2020*, pp. 492-505.
- [33] W. Chen *et al.*, "ROV-MI: Large-Scale, Accurate and Efficient Measurement of ROV Deployment," in *NDSS*, 2022.
- [34] T. McDaniel, J. M. Smith, and M. Schuchard, "Flexsealing BGP against route leaks: peerlock active measurement and analysis," in *NDSS*, 2021.