# Fooling BGP hijack monitors with ease

Anonymous Author(s)

## ABSTRACT

We argue that recent research on detecting BGP hijacks has taken a fundamentally flawed approach. Recent contributions indeed exclusively rely on data collected by public BGP monitors. Unfortunately, such data can be easily gathered, checked, and manipulated by the hijackers themselves.

This paper shows how hijackers can benefit from manipulating the BGP data that defenses use for hijack detection. We analyze the general architecture of monitor-based defense systems, abstracting from the specific techniques, and then pinpoint its fundamental limitations. We then exemplify how hijackers can exploit these limitations to circumvent or weaponize monitor-based defenses. We also demonstrate the effectiveness of the identified attacks with experiments on state-of-the-art systems. We finally take a step back and outline a research agenda for both offensive and defensive sides, toward building robust hijack defenses in the future.

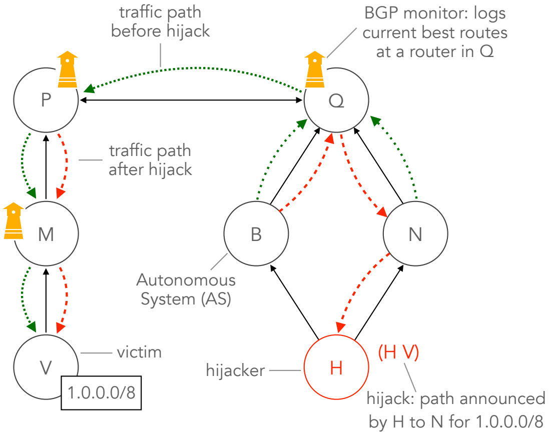

**Figure 1: Example of hijack for prefix 1.0.0.0/8 owned by AS $V$. Single-headed solid arrows indicate the direction of the money in customer-provider links; double-headed arrows represent charge-free links.**

## 1 INTRODUCTION

The Resource Public Key Infrastructure (RPKI) significantly increases Internet routing security by enabling BGP routers to filter out routes originated by unauthorized Autonomous Systems (ASes). Slow at first, RPKI deployment has clearly picked up steam, with an accelerating yearly growth rate above 10% [15]. As of June 2024, RPKI now covers the *majority* of the IPv4 (51%) and IPv6 (53%) prefixes advertised [9].

Although RPKI-based filtering is tremendously useful, it only protects against a subset of the prefix hijacks where the origin AS is incorrect. This means that malicious ASes can still easily evade RPKI-based filtering—even if *all* ASes deploy it—by making sure they advertise their hijacked routes with the correct origin set. Take the situation depicted in Figure 1 as an example and assume that *all* ASes filter routes using RPKI. A malicious AS (here, $H$) advertising `1.0.0.0/8` ($V$'s prefix) to $N$, with the correct origin ($V$) set, would successfully hijack the traffic coming from $Q$, $B$, and $N$.

Nowadays, the most practical solutions for detecting RPKI-evading hijacks are based on control-plane monitoring. These systems retrieve routes collected by monitors from hundreds of BGP sessions globally (e.g., using RouteViews [13] and/or RIPE RIS [12]) and build a knowledge base to classify whether new advertisements are malicious. The intuition here is that rogue advertisements—such as the route ($H$ $V$) advertised by $H$ to $N$ in the example above—will stand out with respect to previous advertisements. Indeed, $H$'s advertisement would be suspicious if $H$ had never advertised a prefix originating from $V$ to $N$ before (i.e., the link $H$-$V$ was previously unseen). Thus, based on classification, it is possible to filter out advertisements (akin to RPKI-based filtering) or alert a human operator for additional investigation. Recent examples of such monitor-based systems include proposals like Artemis [16], DFOH [7], or Beam [2], and commercial solutions such as Cisco's CodeBGP [17].

Monitor-based hijack classification, similar to many Machine Learning (ML) models, implicitly assumes that the training set (the historical routing data) and the test data (the new routes to assess) come from the same distribution. In practice, though, this assumption can be violated whenever an attacker can supply fabricated data, e.g., by poisoning the training set. Such attacks are known as "adversarial attacks" in the ML community, and countless issues have been uncovered [14]. So much so, in fact, that evaluating the robustness of ML models against adversarial attacks and minimizing the associated risks is a key requirement for any production-grade ML system. We believe it should also be the case when building systems on top of inter-domain routing data.

This paper shows that monitor-based hijack classification is indeed prone to adversarial attacks that can render it virtually useless. In particular, we demonstrate that adversaries can easily manipulate routing data so that state-of-the-art detection systems end up either missing actual attacks (false negatives) or reporting non-existent attacks (false positives). More concretely, we detail three attacks that exploit the inherent weaknesses of monitor-based systems. First, we illustrate how hijackers can uncover hundreds of thousands of links within historical data and use them to construct undetectable hijacks. Second, we present a poisoning attack that progressively undermines the accuracy of the defenses, enabling even a small AS to stealthily hijack half of the Internet. Third, we shed light on how hijackers can manipulate the output of monitor-based defenses to attribute the blame for hijacks to remote ASes.

In light of this, we argue that using public routing data for hijack detection is inherently flawed, as BGP routes used in a hijack cannot be differentiated from policy changes, and even legitimate routes may inadvertently expire. The adversarial attacks we uncover in this paper are fundamental and complement previous attacks against monitor-based systems (e.g., [1, 10]). These previous attacks prevent hijacked routes from reaching the monitors by tweaking specific BGP attributes or poisoning the AS path. Thus, they share the same spirit with our attacks concerning preventing the hijacked advertisements from appearing in the test data.

With faith in prompt hijack detection, we take a step back to acknowledge the merits of recent monitor-based systems. We then sketch an exciting future research agenda, outlining possibilities (i) to explore the attacking side thoroughly regarding potential vulnerabilities, strategies, and practicality; and (ii) to design the defending side to be more robust, potentially with ML-inspired techniques, such as diversifying the data source used for hijack detection.

## 2 THE UNDEFENDABLE DEFENSES

We describe the general architecture of hijack detection systems based on BGP public monitors and how it matches three state-of-the-art systems: Artemis [16], DFOH [7], and Beam [2]. We then highlight its fundamental limitations, which thus hold for all the monitor-based defenses.

### 2.1 General architecture

Figure 2 visualizes the main components and operations of monitor-based defense systems. Specifically, routes collected from BGP monitors are at the core of this architecture. Non-suspicious routes[1] collected in the last several weeks or months represent the *knowledge base* of the defense: most systems trust such historical routes. New routes are periodically received from the monitors, checked using the knowledge base, and added to the knowledge base if validated.

Processing new routes entails three main steps.

*Phase 1: trigger.* While each and every route can theoretically be checked, most systems trigger a proper check only on some data – e.g., for their own scalability. Re-validating the same data seems useless and resource-wasteful. So, a prevalent approach is to skip checking data fully consistent with the current knowledge base: hijack checks are typically skipped if a new route was seen in the past for the same prefix, or if all the AS links in it are also used in historical routes. The remaining routes are passed to Phase 2.

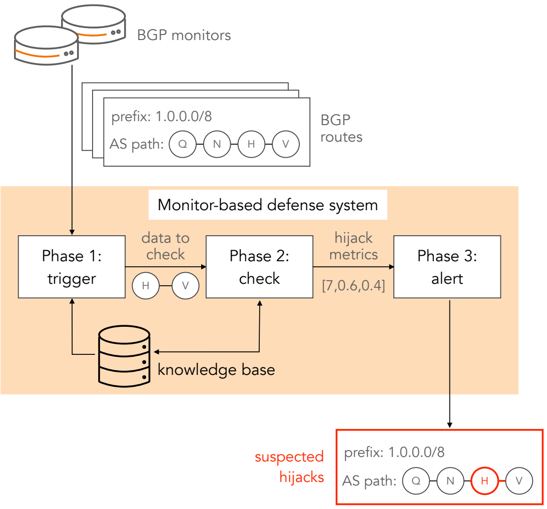

**Figure 2: Architecture of monitor-based defenses.**

*Phase 2: check.* Checking a new route generally maps to computing some metrics that capture the probability that the route is a hijack attempt. Such computation is typically performed by comparing the new route with the knowledge base. This comparison can, for example, include assessing the compatibility of new AS links with the AS-level topology and AS business relationships in the knowledge base [2, 7, 16]. The comparison results are passed to Phase 3.

*Phase 3: alert.* To provide scalable and manageable output for operators, the metrics computed in Phase 2 are post-processed and filtered. Commonly, the defense only alerts on suspicious routes, sometimes providing a (normalized) suspiciousness value [7]. For suspicious routes, defenses typically pinpoint the hijacker as the closest AS to new links.

Artemis, DFOH, and Beam all implement this architecture, albeit with variations. These range from the hijack detection algorithms employed in Phase 2 to the granularity of checks (per-link versus per-path), their scope (whether single-AS or a global service), and the timescale of detection (from near real-time to monthly). These differences have practical implications (e.g., the reactivity to hijacks, the practicality of hijack mitigation, and the time constraints for the detection algorithm), yet they do not exempt them from the fundamental limitations of their overall architecture.

[1] Typically, no route is considered suspicious at the system's bootstrap.

## 2.2 Fundamental limitations

Fundamental limitations of monitor-based defenses derive from the very nature of the routes collected by BGP monitors.

*It is impossible to identify invalid or expired BGP data.* To check for hijacks, monitor-based defenses compare new routing data (e.g., a new path or AS link) with their knowledge base. Suppose that the new data is not classified as a hijack. How long should this data be retained as valid?

There is no fundamentally sound answer to the above question. Some BGP paths and links are stable for months, while others are very short-lived. Worse, the BGP protocol provides no explicit information if paths or links are incompatible with the routing policies of some ASes. Hence, at any time, a previously announced path or link may still be valid even if not currently used – or it may not!

*BGP routes implementing a hijack cannot be distinguished from policy changes.* Operators can never be certain of links between remote ASes. Indeed, for any new BGP path, two scenarios always exist: one where the path results from a hijack attempt and another where it is a consequence of a BGP policy change; these scenarios are indistinguishable. For example, in Figure 1, AS $N$ has no means to know if a hijacker forges the route ($H$ $V$) or if it reflects a legit topology change (e.g., dictated by a new agreement between $H$ and $V$, or updated BGP policies at $V$). Monitor-based systems, thus, must check routes with intrinsically inaccurate heuristics (e.g., based on ASes' geography or business relationships).

*Hijackers can manipulate BGP data.* Monitors collect BGP routes regardless of their origin and nature. By definition, hijackers can inject BGP routes. Hence, hijackers can send arbitrary BGP routes in addition to their hijack attempts. Doing so likely has a low cost for hijackers: no infrastructure is needed besides the one required to launch hijacks, and attacks may be kept stealthy, especially if a few additional BGP routes are announced, without specific time constraints.

Altogether, the above limitations imply that monitor-based defenses can be forced to work on data partially crafted by hijackers and have no way to detect when this is the case.

# 3 EXPLOITING DEFENSES' LIMITATIONS

We now exemplify how hijackers can exploit the fundamental limitations of monitor-based systems described in §2.

*Attacker model.* We consider the attacker can inject BGP routes from a single AS, potentially using a single router. The attacker knows the internal algorithms used by monitor-based defense systems and has access to the data gathered by public BGP monitors. The primary objective of the attack is to circumvent the defense systems, such as evading their detection or manipulating their outputs.

*Experiments.* We provide evidence of the attacks' effectiveness by experimenting with state-of-the-art systems: Artemis, DFOH, and Beam. [2] Our experiments use topologies created by merging CAIDA AS-level graphs over 10 months (as done in [7, 16]). We consider hijackers located in 42 distinct ASes selected from 13 countries in America, Europe, and Asia. For each country, we select at least three ASes: a small AS with a degree under 10, a medium AS with a degree ranging from 100 to 300, and a large AS with a degree from 500 to 3000. We also consider the remaining ASes as victims, with hijackers fabricating links or paths to them. We assume that any hijacker's route reaches 50 random BGP monitors, where the recorded routes represent a combination of the hijacker's crafted routes and existing AS paths to the monitors.

## 3.1 Avoiding the trigger

A direct method hijackers use to neutralize monitor-based defenses is to ensure they skip the hijackers' routes. This compromises the correctness of Phase 1 in Figure 2.

*Recycling old data.* Hijackers can build non-existing BGP paths using the outdated AS paths or links that are still retained in the defenses' knowledge base. As a result, monitor-based defenses (e.g., [7, 16]) will skip checking them because all AS links were already seen. For example, the hijacker's route visualized in Figure 1 would not be checked at all if AS $H$ did have a link with AS $V$ in the past. Using past links with $M$ or $P$ in this scenario would also allow hijacker $H$ to evade the defenses. Indeed, it is impossible for defenses to identify invalid or expired BGP data (cf. §2.2).

**Table 1: Additional links from historical routes. The original topology (March 2024) contains 545K links.**

| 1 month | 3 months | 6 months | 10 months |
|---|---|---|---|
| 63K (+12%) | 136K (+25%) | 232K (+43%) | 350K (+64%) |

*Experiment results.* We confirm that Artemis and DFOH do not trigger hijack checks for routes that include the links present in the defenses' knowledge bases but are absent from the latest set of collected routes. Additionally, we discovered that hijackers could identify thousands of outdated links by mining them from routes previously collected by monitors; refer to Table 1. For instance, DFOH regards any link observed in the past ten months as valid within their knowledge bases, yet it is likely that 40% of these are no longer active (i.e., not observed in March 2024).

[2] Our code is available at https://github.com/dazzling-gauss/route-poison.

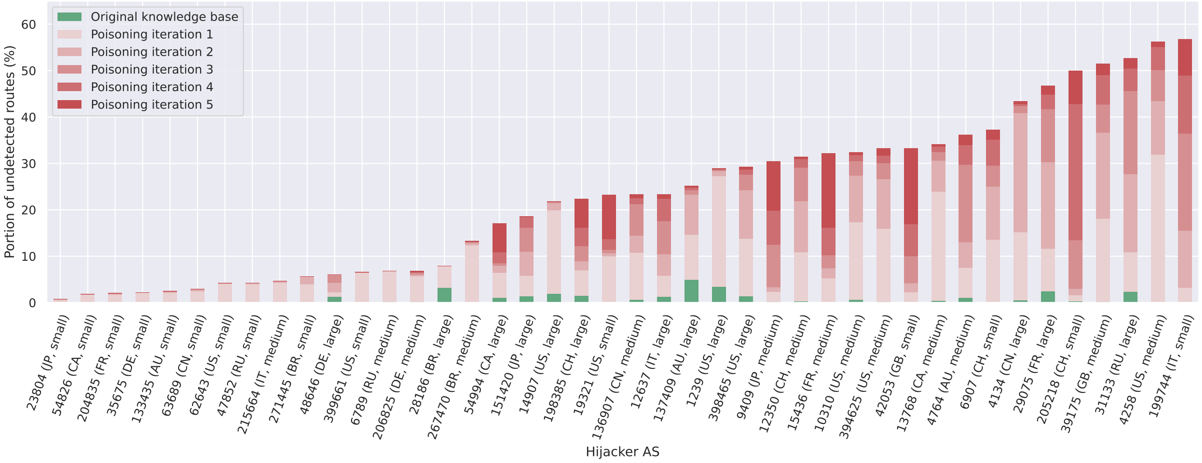

**Figure 3: Hijackers can force a progressive loss of accuracy of monitor-based defenses by progressively adding more and more fake AS links. The figure shows that even small ASes can pretend to have direct links with more than 50% of the entire Internet without raising any alert from DFOH.**

*Is there an easy fix?* The chances of successful attacks depend on *when* defense systems discard historical routes. Thus, naively, defense systems can discard old routes and rely on only fresh data (e.g., collected over a few days). However, doing so may generate many false hijack alerts because valid links (e.g., backup ones) may re-appear periodically, depending on short-term commercial agreements or failures. Moreover, a smaller set of historical data may not capture well enough structural features and possible path diversity used by defenses to distinguish genuine new BGP routes and hijack attempts [7]. More fundamentally, unpredictable BGP dynamics are outside the control of defense systems because the frequency of policy changes and their impact on BGP routes seem an emergent property of Internet routing. In other words, there is no theoretically sound choice that can be made within monitor-based systems on how much history monitor-based defense systems should keep.

## 3.2 Poisoning the check

Hijackers can also evade monitor-based defenses by poisoning the monitor-collected routes so that the checks in Phase 2 (cf. Figure 2) are biased in the hijackers' favor.

*Building a castle of fake news.* Hijackers can exploit the inaccuracies of defenses to poison their knowledge base, ultimately biasing the outcomes of hijack checks. Indeed, since BGP routes generated by hijacks are indistinguishable from legitimate policy changes (cf. §2.2), monitor-based defenses always have imperfect checks. Thus, hijackers can introduce fabricated AS links and paths into the defenses' knowledge base undetected. Particularly, hijackers can *predict* the undetectable fabricated data by simulating defenses with known algorithms and inputs. Worse, once the fabricated data is in the defenses' knowledge base, it forces the defenses to rely on a distorted view of the Internet. This allows more fake paths and links to be accepted as legitimate, leading to more hijacks passing defenses' checks *and* also further polluting.

Let us consider again the example in Figure 1, where AS $V$ deploys Artemis [16]. Here, the knowledge base of Artemis consists of the links from *all* monitor-collected routes for all prefixes (i.e., including the ones that AS $V$ does not own). Following the above attack strategy, the hijacker $H$ can advertise a route ($H$ $M$) for a prefix that no other AS advertises, thus adding the link ($H$ $M$) to Artemis' knowledge base. As a result, the attack can inject a route ($H$ $M$ $V$) to hijack AS $V$'s prefix without being detected.

*Experiment results.* Here, we focus on the experiments with DFOH [7]. Particularly, we repeatedly pollute DFOH's knowledge base by injecting hijackers' routes consisting of their fabricated links with the victim ASes that DFOH misses.

Figure 3 shows the number of fake AS links that hijackers can add within five iterations of the above approach. Results for the first iteration look consistent with DFOH accuracy reported in [7]. However, within five iterations, hijackers can add direct links between their ASes and more than 20% of the entire Internet (i.e., ≈ 15,000 ASes) in roughly half of our experiments. Worse, the rightmost part of the figure shows that even small ASes can fabricate direct links with more than half of the Internet!

*Is there an easy fix?* The feasibility of the described poisoning attack stems from the fact that defense mechanisms derive their knowledge bases from all routes collected by all monitors. Thus, a naive approach might limit the information source to certain monitors or specifically chosen route prefixes. Nonetheless, this does not ensure that the hijackers' routes will be excluded from these monitors, and such a restricted knowledge base might prove inadequate for detecting hijacks [7]. Moreover, the inherent inaccuracy in detecting hijacks of monitor-based defenses is unavoidable. Worse, it tends to accumulate, allowing hijackers to force such a progressive accuracy reduction.

## 3.3 Weaponizing the alert

Without evading the defenses, hijackers can still render them impractical by deliberately increasing the false alerts in Phase 3 in Figure 2. Worse yet, a hijack detection system can be ignored or disabled if it persistently triggers minor alerts.

*Blaming remote ASes.* For any alert raised by a monitor-based defense, there is no certainty regarding the originating AS of the flagged BGP routes and if they are caused by hijack attempts or non-malicious policy changes – see again §2.2. Hijackers can exploit this uncertainty: instead of trying to dodge detection, they can aim for specifically crafted routes to be well visible and flagged as anomalous by monitor-based defenses. In particular, attackers can inject routes with non-existing links between remote ASes, blaming them for propagating such routes. Note that, to avoid being identified as the perpetrators by the defense, the hijackers blame only ASes they can reach with legitimate links.

For example, hijacker $H$ in Figure 1 can send a route with AS path ($H$ $B$ $M$ $V$) to $N$, which triggers an alert at any monitor-based defense. Subsequently, the defense (e.g., Artemis and DFOH) would report the fake link ($B$ $M$) as suspicious and identify AS $B$ as the hijacker because B appears as the first announcer of the fake link. If an alert is raised, operators in $N$ may investigate the incident without, however, being able to certainly identify which AS first injected the suspicious route and why.

We anticipate that this technique can be used by hijackers in at least two ways. First, it can enable attackers to perform hijacks while deflecting the blame. In the previous example, ASes $N$ and $Q$ would still prefer the route ($H$ $B$ $M$ $V$) because it was received from a customer, and alerts would focus on AS $B$, as already discussed. Second, routes can be engineered to cause reputation damage to specific remote ASes. For example, $H$ can inject multiple paths, including fake links between $B$ and other ASes for different prefixes.

*Experiment results.* To assess the practicality of this technique, we compute the number of ASes that a hijacker can blame, which is essentially the ASes the hijackers can reach with legitimate links. Figure 4 shows the percentage of ASes that can be blamed as perpetrators of a hijack, varying the attacker ASes and their distances (i.e., AS hops) from the blamed victims. We observe that hijackers can always divert the blame for a hijack to any other AS within a distance of 5 hops between them (except when located in single-homed stubs, such as the AS at the leftmost of the figure). Large hijacker ASes can reach many ASes with a few hops; hence, they can blame these ASes with short AS paths. Similarly, smaller hijacker ASes may need routes with up to 5 hops. In practice, injecting routes with long AS paths is not a problem — attackers can propagate routes for IP prefixes that are not announced by other ASes (e.g., not announced at all or only a less specific prefix is announced). Further, longer paths generally increase the number of possible hijackers if future defenses indicate all ASes after alerted links as suspicious.

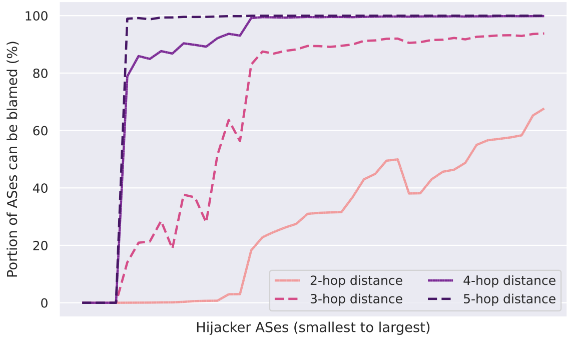

**Figure 4: The portion of ASes that can be blamed as the hijack perpetrators with varying attacker ASes and their distances from the victim to be blamed.**

*Is there an easy fix?* Blaming attacks can be mitigated if the defenses do not identify the perpetrators in the first place. However, this may incentivize hijackers to repeat their attacks. More importantly, attackers can always force monitor-based defenses to raise many alerts as they diligently report each suspicious new route. Indeed, the hijackers can simulate the defenses and construct routes containing only legitimate links corresponding to the defenses' false positives. With these routes, attackers can ensure that many alerts are raised – e.g., continuously over time or in spikes, effectively rendering the defenses impractical!

## 4 STEPPING BACK TO MOVE FORWARD

Monitor-based systems [2, 7, 16] can be effective against existing BGP hijacks. However, they are vulnerable to more sophisticated attacks by defense-aware hijackers, as exemplified in Section 3. How can we build more robust defenses, given the fundamental limitations of public routing data?

Securing the BGP protocol looks appealing. Unsurprisingly, this has been considered before, with proposals to extend BGP (e.g., BGPsec [8]) or to replace it altogether (e.g., SCION [4]). Despite efforts to establish incentives for the deployment of these solutions [5, 6], they have seen limited success, serving as a living testament to the practical hurdles when attempting to modify pervasive protocols like BGP.

We thus believe that prompt hijack detection and mitigation is a useful primitive that still deserves further attention in the future. Toward the goal of building such a robust defense primitive, we propose a three-step research agenda.

## 4.1 Knowing your enemy

Future research should first examine monitor-based systems from an offensive perspective to develop a more thorough understanding of possible attacks. This includes (1) conducting a more thorough vulnerability analysis of monitor-based approaches; (2) fully defining attack vectors based on specific combinations of identified vulnerabilities; and (3) demonstrating the practicality of the designed attacks regarding feasibility, alignment to realistic hijackers' goals, and so on.

Firstly, we suspect more potential attacks against monitor-based defenses than this paper has outlined, especially when targeting the defenses' specific design choices. For example, Artemis and DFOH classify bidirectional AS links as legitimate without any check [7, 16]. However, hijackers can still inject multiple BGP routes with remote bidirectional links instrumental to their attempts. Another instance is where Beam calculates a dynamic alert threshold using the standard deviations from the previous hour's data [2]. Consequently, hijackers could manipulate the threshold for the upcoming hour by injecting carefully crafted announcements into BGP monitors. Another example of design choices is the real-time checks in monitor-based defenses, which assess new routes in batches every few minutes. Thus, attackers may initiate a hijack while flooding routes with new data unrelated to the hijack, thereby overwhelming the defenses with numerous alerts, among which only one pertains to the actual hijack.

Secondly, we believe attackers can easily combine multiple attacks against single phases of the monitor-based defenses. For example, hijackers can inject BGP routes that do not trigger hijack detection (i.e., a Phase-1 attack) to pollute the defense's knowledge base (i.e., targeting Phase 2), representing the platform to launch hijacks at later times. This attack shares the same spirit as the recycling old data attack (cf. §3.1). Hijackers can also combine attacks in Phases 2 and 3 by adding a few fake links, allowing hijackers to reach more ASes and subsequently blame them.

Thirdly, we suggest that the feasibility of hijackers must be thoroughly assessed, considering practical limitations. For instance, hijackers may typically possess only limited knowledge of defense systems, which could include imprecise information about the locations of monitors or an incomplete ability to predict the results of defensive hijack checks.

## 4.2 Learning from machine learning

A fundamental cause of the vulnerability of monitor-based defenses is the nature of the data they rely on: BGP routes are deliberately opaque, noisy, and, most importantly, untrustworthy, as they can be manipulated by attackers. This setting is clearly reminiscent of problems other research communities (e.g., ML) face when focusing on data-driven approaches. We posit that interesting ideas can be borrowed from work that robustifies ML approaches against noisy training sets and adversarial input [14].

More concretely, a potential solution for monitor-based hijack defenses is to use multiple, diverse data sources. Artemis makes a first step in this direction by combining public BGP data with information (e.g., AS links) local to the protected AS. This additional data source indeed makes Artemis robust against attacks where hijackers pretend to have direct links with the protected AS.

We further envision more radical approaches relying on both control and data plane data. Regarding control plane data, an interesting direction is to identify practical privacy-preserving mechanisms that competing ASes can use to exchange minimal information targeted to hijack detection. Particularly, tailoring existing generic inter-AS collaboration frameworks (e.g., [11]) and policy-preserving schemes (e.g., [3]) to hijack defense may provide the right mix between high incentives and low data to share. Regarding data plane data, a potential solution is capturing significant changes in path-dependent traffic metrics, such as delays, to confirm or reject control plane inferences. Data plane signals can indeed complement routing information while also being harder to manipulate or learn by hijackers.

## 4.3 Turning active

While modifying BGP is challenging in practice, building external mechanisms outside of the BGP protocol (similarly to RPKI) seems much more viable for hijack detection and mitigation. Notably, we desire inter-AS querying mechanisms that allow an AS to directly query others about links or paths they use for given destinations. Path validation primitives that enable the validation of paths in the data plane (e.g., by using encryption) are also desirable. If employed to validate (suspicious or important) routes before using them to forward traffic, these mechanisms may even enable *hijack prevention*. That said, designing inter-AS route validation mechanisms may entail challenges such as ensuring quick and reliable information exchange regardless of the network conditions and guaranteeing that communications cannot be interrupted or tampered with by attackers even if they are on-path. Tackling these challenges is thus an exciting venue for future research.

# REFERENCES

[1] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. 2019. SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *Proc. ACM CCS*.

[2] Yihao Chen, Qilei Yin, Qi Li, Zhuotao Liu, Ke Xu, Yi Xu, Mingwei Xu, Ziqian Liu, and Jianping Wu. 2024. Learning with Semantics: Towards a Semantics-Aware Routing Anomaly Detection System. In *Proc. USENIX Security*.

[3] Marco Chiesa, Roberto di Lallo, Gabriele Lospoto, Habib Mostafaei, Massimo Rimondini, and Giuseppe Di Battista. 2017. PrIXP: Preserving the privacy of routing policies at Internet eXchange Points. In *Proc. IFIP/IEEE IM*.

[4] Corine de Kater, Nicola Rustignoli, and Adrian Perrig. 2024. *SCION Overview*. Technical Report. Internet Engineering Task Force.

[5] Phillipa Gill, Michael Schapira, and Sharon Goldberg. 2011. Let the market drive deployment: a strategy for transitioning to BGP security. In *Proc. ACM SIGCOMM*.

[6] Tomas Hlavacek, Italo Cunha, Yossi Gilad, Amir Herzberg, Ethan Katz-Bassett, Michael Schapira, and Haya Shulman. 2020. DISCO: Sidestepping RPKI's deployment barriers. In *Proc. NDSS*.

[7] Thomas Holterbach, Thomas Alfroy, Amreesh D. Phokeer, Alberto Dainotti, and Cristel Pelsser. 2024. A System to Detect Forged-Origin Hijacks. In *Proc. USENIX NSDI*.

[8] Matt Lepinski and Kotikalapudi Sriram. 2017. BGPsec Protocol Specification. RFC 8205. https://doi.org/10.17487/RFC8205

[9] Doug Madory. 2024. RPKI ROV Deployment Reaches Major Milestone. https://manrs.org/2024/05/rpki-rov-deployment-reaches-major-milestone/.

[10] Alexandros Milolidakis, Tobias Bühler, Kunyu Wang, Marco Chiesa, Laurent Vanbever, and Stefano Vissicchio. 2023. On the Effectiveness of BGP Hijackers That Evade Public Route Collectors. *IEEE Access* (2023).

[11] Cristian Munteanu, Oliver Gasser, Ingmar Poese, Georgios Smaragdakis, and Anja Feldmann. 2023. Enabling multi-hop ISP-hypergiant collaboration. In *Proc. ANRW*.

[12] RIPE NCC. 2024. Routing Information Service (RIS). www.ripe.net/data-tools/stats/ris/.

[13] University of Oregon. 2024. RouteViews Project. www.routeviews.org/routeviews/.

[14] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. 2018. Sok: Security and privacy in machine learning. In *Proc. IEEE EuroS&P*.

[15] Nils Rodday, Ítalo Cunha, Randy Bush, Ethan Katz-Bassett, Gabi Dreo Rodosek, Thomas C Schmidt, and Matthias Wählisch. 2023. The Resource Public Key Infrastructure (RPKI): A Survey on Measurements and Future Prospects. *IEEE TNSM* (2023).

[16] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. 2018. ARTEMIS: Neutralizing BGP Hijacking Within a Minute. *IEEE/ACM ToN* (2018).

[17] Joe Vaccaro. 2023. Cisco Announces Acquisition of Code BGP to Deepen ThousandEyes' BGP Capabilities. https://www.thousandeyes.com/blog/cisco-announces-acquisition-of-codebgp.