

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

ASRogue: Manipulating ASRank-Inferred AS Relationships

Yi Xu, Yihao Chen, Ke Xu, Qi Li, and Jianping Wu, *Tsinghua University*

https://www.usenix.org/conference/usenixsecurity26/presentation/xu-yi

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

ASRogue: Manipulating ASRank-Inferred AS Relationships

Yi Xu, Yihao Chen, Ke Xu, Qi Li, Jianping Wu
Tsinghua University
{y-xu22,yh-chen21}@mails.tsinghua.edu.cn, {xuke,qli01,wjp}@tsinghua.edu.cn

Abstract

Accurate knowledge of Autonomous System (AS) business relationships is critical for many Internet security and networking tasks. Since these relationships are not publicly visible, operators and researchers rely on algorithms that infer them from empirical routing data. Yet, despite decades of widespread deployment, the security of AS relationship inference against adversarial manipulation remains largely underexplored. To close this gap, we conduct a systematic security analysis of ASRank, a foundational algorithm that underpins many established methods and widely used datasets. We show that ASRank's output is highly sensitive to small, targeted perturbations of AS triplets extracted from BGP paths. Carefully chosen triplet changes can alter inference order and trigger cascading misclassifications. Building on this insight, we present ASRogue, a manipulation attack against ASRank-based inference. ASRogue adaptively crafts AS triplets to steer inference order and outcomes, scales triplet injection via prefix splitting and AS-path poisoning, and embeds forged triplets into policy-compliant BGP announcements that can evade common routing defenses. Extensive evaluation shows that when the attacker controls at least two ASes, ASRogue achieves an overall 96.7% manipulation success rate, exceeding 99% for small and medium-sized providers, and remains effective under constrained attack overhead and temporal variation. Finally, controlled experiments on the PEERING testbed validate the feasibility of launching ASRogue attacks in the real-world Internet.

1 Introduction

The Internet consists of over 80,000 Autonomous Systems (ASes) that interconnect via the Border Gateway Protocol (BGP). These interconnections are governed by economic agreements between AS operators, commonly referred to as *AS business relationships*, or simply *AS relationships*. In practice, two fundamental AS relationship types, namely peer-to-peer (P2P) and provider-to-customer (P2C), play a critical role in determining an AS's local routing policies in BGP. Accurate knowledge of these relationships, *i.e.,* whether two ASes are connected and the type of relationship between them, is therefore essential for understanding global routing behaviors and enables important tasks such as BGP operations [1–6], Internet measurement [7–9], anomaly detection [10–13], and network protocol design [14–16].

Notably, AS business relationships are not publicly disclosed by default, and therefore knowledge of them relies on algorithmic inference from empirical routing data. Among existing inference methods [17–21], CAIDA's ASRank algorithm [17] is a foundational approach that has shaped many subsequent techniques and underpins CAIDA's regular releases of AS-relationship datasets [22]. These ASRank-inferred datasets have been widely used in operation practice [1–6] and have supported over 800 scientific studies over the past decade, making ASRank arguably the most influential method in this regard.

Yet, despite its critical importance, little is known about ASRank's security against adversarial manipulation. To date, no prior work has systematically examined if ASRank's inferred relationships remain reliable under adversarial manipulation. This gap is particularly concerning given CAIDA's automated pipeline for producing its AS relationship datasets: real-world BGP data are periodically collected from a fixed set of public vantage points and processed by an open-source ASRank implementation with minimal preprocessing. Such transparency enables white-box exploitation of algorithmic weaknesses, which, if successful, could place a wide range of downstream analysis and operational systems at significant risk.

To address this gap, we conduct a detailed analysis of the ASRank algorithm and pinpoint components susceptible to adversarial manipulation. We find that ASRank's heuristics hinge on *empirical AS triplets* (*i.e.,* three consecutive ASes observed on BGP paths) in two key ways. First, ASRank infers connectivity between ASes based on their consecutive co-occurrence within triplets, thus selectively injecting or suppressing triplets can directly create or eliminate inferred links. Second, ASRank determines the relationship type of

each link using a greedy procedure ordered by decreasing AS transit degree, a metric derived from AS triplet counts. This ordering is crucial because earlier determined inferences directly constrain the classification of later, unresolved links. Since many ASes have similar transit degrees, even small perturbations to AS triplets can alter the inference order and, ultimately, the inferred relationships (see Figure 1).

These findings expose a concrete point of adversarial control in ASRank, *i.e., an attacker can steer inferred relationships by injecting carefully crafted AS triplets*. Turning this observation into a practical attack, however, entails three major challenges. First, targeted manipulation is non-trivial given the diversity of ASes and ASRank's order-dependent inference, and thus requires principled, adaptive methods for crafting AS triplets that reliably induce specific inference outcomes. Second, scaling the attack to inject multiple distinct AS triplets is difficult, because, to avoid conflicts in ASRank's input, each triplet must be carried by an independent route, which in turn requires control over significant IP address resources. Third, crafted triplets must be embedded into BGP routes that can survive real-world routing defenses, such as ROV and Peerlock, as well as ASRank's own sanitization.

In this paper, we demonstrate ASROGUE, a practical manipulation attack on ASRank-based AS relationship inference that overcomes the above challenges. ASROGUE supports two manipulation goals: (i) forcing ASRank to infer a P2C relationship for a target pair of ASes, regardless of their original inference outcome; and (ii) forcing ASRank to infer a P2P relationship between a pair of ASes that would otherwise have no inferred relationship. To achieve the goals, ASROGUE adaptively crafts AS triplets by (i) deriving the inference-order perturbations and forged connectivity required for the target outcome and (ii) realizing them under realistic operational constraints. To scale triplet injection, ASROGUE subdivides controlled prefixes and applies AS-path poisoning to map each triplet to a unique "prefix-VP" combination. Finally, ASROGUE embeds these triplets into policy-compliant BGP announcements with carefully constructed AS paths that can evade RPKI/ROV, Peerlock, and ASRank's preprocessing under realistic attack scenarios.

We mirror the CAIDA ASRank pipeline and extensively evaluate ASROGUE across a range of attack scenarios. ASROGUE achieves an overall manipulation success rate of 96.7%, exceeding 99% for relationships involving small and medium-sized providers, which constitute the majority of the Internet. We further show that the attack is highly resource-efficient. With modest overhead, injecting 100 crafted triplets can manipulate nearly 50% of all inferred relationships, and only 10-30 announcements can flip over 80–90% of inferred P2C relationships involving small or zero-size providers. The attack also remains effective on real-world RIB snapshots up to one week old (above 87% success) and can be improved to 94.7% by adding a small robustness buffer, indicating resilience to temporal drift. Finally, controlled experiments on the PEERING testbed confirm that forged triplets can propagate in the global Internet, validating the practical feasibility of ASROGUE.

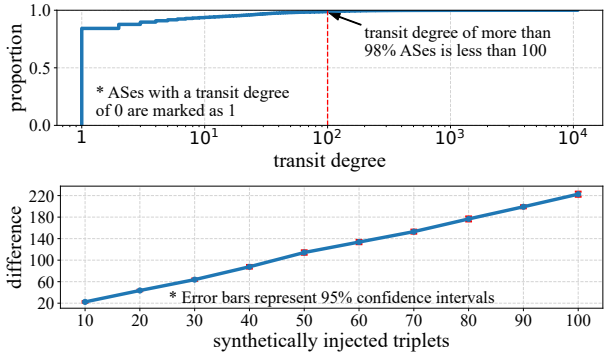


Figure 1: Transit degree distribution of ASes (top) and sensitivity of ASRank inference to injected AS triplets (bottom).

To summarize, our contributions are four-fold:

- We present the first systematic security analysis against AS relationship inference under adversarial manipulation, and uncover critical vulnerabilities in the widely used ASRank algorithm.
- We introduce ASROGUE, a practical manipulation attack that exploits these vulnerabilities to achieve adaptive, evasive, and scalable control over ASRank's inferred relationships across two realistic attack scenarios.
- We extensively evaluate ASROGUE at scale using CAIDA's AS relationship inference pipeline, demonstrating that it achieves a 96.7% attack success rate when the attacker controls at least two ASes.
- We validate the real-world feasibility of ASROGUE via controlled BGP experiments on the PEERING testbed, confirming that forged AS triplets can be successfully propagated and observed in the global Internet.

2 Background

AS Relationships and Their Inference. ASes interconnect under mutually agreed-upon business arrangements, commonly referred to as AS (business) relationships. The two fundamental relationship types[1] are provider-to-customer (P2C) and peer-to-peer (P2P) [17]. In a P2C relationship, the provider AS offers transit to the customer AS for payment, while in a P2P relationship, two ASes exchange traffic without settlement. These relationships largely determine local routing policies in BGP, including route preference and export

[1] Other relationship types, such as sibling or hybrid relationships, do not directly manifest in BGP or are rare in practice [20]. Therefore, we focus on inference methods for P2P/P2C relationships in this work.

behavior. Because AS relationships are typically considered proprietary and not disclosed by default, knowledge of them relies on algorithmic inference from empirical routing data. Existing methods [17–21] generally operate by analyzing BGP routes collected from public vantage points and inferring relationships using heuristic or statistical techniques.

CAIDA's ASRank Algorithm. Among existing methods for AS relationship inference [17–21], CAIDA's ASRank algorithm [17] is arguably the most influential and widely used. ASRank follows a multi-step workflow to infer AS relationships from empirical BGP routes, and we provide the details of the ASRank inference algorithm in Appendix A. It begins by discarding paths with artifacts or loops to ensure input integrity (Step 1). ASes are then sorted in descending order of transit degree, computed from counts of AS triplets observed in the data (Step 2). Based on this ordering, ASRank first identifies a clique of top-level ASes and infers the corresponding P2P relationships within the clique (Step 3). After filtering poisoned paths further (Step 4), it infers P2C relationships in a greedy, top-down manner (Step 5). During this process, newly inferred relationships serve as a basis for subsequent inferences, thereby progressively determining P2C relationships until no additional ones can be resolved. The algorithm then handles special cases, such as ASes with atypical transit patterns (Steps 6–10). Finally, any remaining unresolved links are inferred as P2P relationships (Step 11).

We refer readers to the original ASRank paper for full algorithmic details, and to Appendix B for a discussion of ASRank's widespread adoption and influence. Despite its central role, the security of ASRank has received little attention and, as we show later, is vulnerable to adversarial manipulation.

BGP Route Filtering. In practice, several classes of BGP routes are commonly filtered. First, as a matter of operational hygiene, ASes often discard routes for overly specific prefixes, *e.g.,* longer than /24 in IPv4, and those containing AS-path loops. Second, routes with invalid prefix-origin associations are filtered by ASes that deploy Route Origin Validation (ROV) based on the Resource Public Key Infrastructure (RPKI), or apply filtering based on Internet Routing Registry (IRR) records. Third, routes that violate inter-AS adjacency constraints are filtered by ASes enforcing link-level validation mechanisms, such as Peerlock(-lite), AS Provider Authorization (ASPA), and Path-End. Finally, more comprehensive schemes such as BGPsec reject routes that fail per-hop cryptographic attestations. To be effective, a forged route must evade potential filtering along its propagation. However, many of these schemes see limited real-world deployment due to incentive and performance constraints. In this paper, we therefore focus on the most practically relevant mechanisms, *i.e.,* ROV, Peerlock-lite, and hygienic filtering, and demonstrate that each can be evaded by our proposed attack.

AS-path Poisoning. AS-path poisoning is an established BGP route engineering technique in which an AS intentionally inserts selected ASNs into the AS-path of a route announcement to influence its propagation. Because BGP's loop-detection mechanism causes an AS to reject any route containing its own ASN, this technique can prevent specific ASes from accepting a route, thereby steering route propagation. In practice, AS-path poisoning is commonly implemented by embedding the ASNs to be avoided within the AS path, often between duplicated occurrences of the announcing AS's own ASN, which preserves a valid route origin while triggering loop-induced rejection at the targeted ASes. In this paper, we apply AS-path poisoning to steer the propagation of forged routes and therefore improve the scalability of our proposed attack.

3 Problem Statement

In this section, we analyze the inherent fragility of ASRank, formalize key concepts, and define the threat model addressed in this paper.

3.1 ASRank Fragility

ASRank infers AS relationships through a multi-step workflow, with the P2C inference step (*i.e.,* Step 5) playing a crucial role, as it accounts for approximately 90% of all P2C inferences. In Step 5, ASRank processes ASes in descending order of their *transit degree*. The transit degree of an AS is defined by the number of distinct neighbors adjacent to it in any observed *AS triplet* (X,Y,Z), where Y is the middle AS. During this step, when visiting an AS Z, ASRank examines triplets of the form (X,Y,Z) and applies the following heuristic: if the relationship between X and Y is already known (either P2P, denoted as $X-Y$, or P2C, denoted as $X>Y$), then ASRank infers the relationship $Y>Z$.

A key characteristic of this process is its *order-dependence*: the relationships inferred earlier in Step 5 influence which triplets can trigger further inferences. In practice, many ASes exhibit similar transit degrees, as evidenced by the strong clustering in the cumulative distribution shown in Figure 1 (top). As a result, even minor changes in the observed set of AS triplets can significantly alter the order of inferences. This is illustrated in Figure 1 (bottom), where the injection of just a few random synthetic triplets leads to noticeable changes in the inferred relationships by ASRank. We now demonstrate this effect with a concrete minimal example.

A Minimal Example. Consider the following observed BGP paths, where we highlight only the contiguous AS-path segments that are relevant for inference:

path 1: $\ldots\ M\ N\ A\ B\ C\ D$
path 2: $\ldots\ M\ N\ A\ B\ E$
path 3: $\ldots\ M\ N\ A\ F$
path 4: $\ldots\ M\ N\ A\ G$

These paths represent a typical scenario where multiple downstream ASes are accessible through a shared chain of transit

providers. ASRank initially infers a top-level clique of major transit ASes. For simplicity, we assume that M and N are part of this clique, with their mutual relationship inferred as P2P ($M-N$) prior to Step 5.

From these paths, ASRank extracts AS triplets and computes the transit degrees (denoted by TD). AS A appears as the middle AS in four triplets and has four distinct neighbors, yielding $\mathrm{TD}(A)=4$. Similarly, $\mathrm{TD}(B)=3$ and $\mathrm{TD}(C)=2$, while D, E, F, and G all have a transit degree of zero. ASRank thus visits ASes in the following order during Step 5:

$$A \to B \to C \to D \to E \to F \to G.$$

Starting with A, ASRank infers $N>A$ from (M,N,A) as $M-N$ is already known. It then moves to B, inferring $A>B$ from the triplet (N,A,B), based on the previously inferred $N>A$. Following this sequence, ASRank infers the following P2C relationships during Step 5:

$$\{N>A,\ A>B,\ B>C,\ C>D,\ B>E,\ A>F,\ A>G\}.$$

Small Perturbations, Different Outcomes. Now consider adding two additional BGP paths:

path 5: $\ldots\ X\ A\ B\ D\ C\ Y$
path 6: $\ldots\ X\ F\ D\ G\ Y$

These paths could arise in realistic, policy-compliant BGP deployments that are not filtered by standard sanitization processes. A detailed discussion will be provided in §3.3.

Due to the new introduced AS triplets, the transit degree of D increases to $\mathrm{TD}(D)=4$, surpassing that of C, whose transit degree becomes $\mathrm{TD}(C)=3$. As a result, D is visited *before* C in Step 5, resulting in the updated inference order:

$$A \to B \to D \to C \to \ldots$$

This reordered traversal leads to a different inference chain. When visiting D, the relationship $A>B$ has already been inferred, and (A,B,D) in path 5 therefore satisfies the heuristic and implies $B>D$. Later, when visiting C, (B,D,C) in path 5 together with the newly inferred $B>D$ again matches the heuristic and implies $D>C$. Consequently, the final set of inferred relationships includes:

$$\{N>A,\ A>B,\ B>D,\ D>C,\ B>C,\ B>E,\ A>F,\ A>G\}.$$

Compared to the original outcome, $C>D$ is replaced by $D>C$, and a new P2C relationship $B>D$ is inferred.

Takeaway. ASRank inference is highly sensitive to small changes in the observed AS triplets. Even minor perturbations can alter the inference order, triggering cascading effects that result in qualitatively different inferred relationships.

3.2 Problem Formulation

We now formalize the concepts underlying ASRank inference and define manipulation objectives that capture the behaviors illustrated in §3.1. We begin with the input to ASRank, namely observed BGP routes.

Definition 1 (BGP Route)**.** A BGP route is a tuple (p,v,π), where p denotes an IP prefix, v denotes the vantage point that observes the route, and $\pi=(a_1,a_2,\ldots,a_k)$ is the AS path. Each a_i represents an AS on the control-plane path, and a_k is the origin AS that supposedly announces prefix p.

ASRank operates on a collection of such routes after applying standard sanitization and filtering.

Definition 2 (Observed Routes)**.** Let $\mathbb{R}$ denote a finite set of BGP routes observed by ASRank. Relationship inference is performed solely based on the AS paths contained in $\mathbb{R}$.

From observed AS paths, ASRank extracts local path structures in the form of AS triplets, which serve as the basic evidence for inference.

Definition 3 (AS Triplet)**.** Given a route (p,v,π) with $\pi=(a_1,a_2,\ldots,a_k)$, an AS triplet is any contiguous subsequence (a_i,a_{i+1},a_{i+2}) for $1\le i\le k-2$. For a set of routes $\mathbb{R}$, let $T(\mathbb{R})$ denote the multiset of all AS triplets extracted from the AS paths in $\mathbb{R}$.

Based on AS triplets, ASRank derives transit degrees as the primary metric that determines the inference order.

Definition 4 (Transit Degree)**.** Given observed routes $\mathbb{R}$ and an AS y, the transit degree $\mathrm{TD}(y)$ is the cardinality of the set $\{x,z \mid (x,y,z)\in T(\mathbb{R})\}$.

Then, we model the overall inference process as follows.

Definition 5 (ASRank Inference)**.** ASRank inference is a deterministic mapping from observed routes to inferred AS relationships. Given a set of routes $\mathbb{R}$, ASRank outputs an inference result $I(\mathbb{R})$, which assigns each ordered pair of ASes (a,b) one of three labels: P2C (denoted $a>b$), P2P (denoted $a-b$), or none (no relationship inferred).

To represent ASRank's core inference logic, we further introduce the following notation.

Definition 6 (Triplet-Based Inference Heuristic)**.** For an AS triplet (x,y,z), we use $x>y\ ?\ z$ (resp., $x-y\ ?\ z$) to denote that the relationship between x and y is inferred as P2C (resp., P2P) while the relationship between y and z is unresolved. ASRank's core inference rules are ❶ $x>y\ ?\ z \Rightarrow y>z$ and ❷ $x-y\ ?\ z \Rightarrow y>z$.

We next distinguish inference results obtained under benign inputs from those obtained after adversarial manipulation.

Definition 7 (Baseline and Attacked Inference)**.** Let $\mathbb{R}$ be the set of routes observed under benign conditions, and let $\mathbb{R}_{\text{adv}}$ be an additional set of routes injected by an adversary. We define the baseline inference as $I_{\text{base}} = I(\mathbb{R})$, and the attacked inference as $I_{\text{att}} = I(\mathbb{R} \cup \mathbb{R}_{\text{adv}})$.

Finally, we define the manipulation objectives considered in this work.

Definition 8 (P2C Manipulation)**.** For a given target AS pair (a,b), the objective of P2C manipulation is to induce ASRank to infer a P2C relationship from a to b under adversarial inputs. Formally, the objective is to achieve

$$\forall a,b:\ I_{\text{att}}(a,b) = \textsf{P2C}. \tag{1}$$

Definition 9 (P2P Manipulation)**.** For a given target AS pair (a,b) with no baseline inferred relationship, the objective of P2P manipulation is to induce ASRank to infer a P2P relationship between a and b under adversarial inputs. Formally, the objective is to achieve

$$\forall a,b,\ I_{\text{base}}(a,b) = \textsf{none}:\ I_{\text{att}}(a,b) = \textsf{P2P}. \tag{2}$$

We note that P2P manipulation is only applicable to AS pairs without a baseline inferred relationship, since ASRank assigns P2P relationships only after all earlier P2C inference steps have been exhausted.

3.3 Threat Model

Public Measurement Setting. ASRank derives AS relationships by analyzing BGP routing data collected from multiple public vantage points (VPs), such as RouteViews and RIPE RIS. These VPs consistently capture BGP updates and periodically release routing snapshots, which are subsequently processed through CAIDA's automated ASRank pipeline. The methodology behind ASRank, including its sanitization rules and inference schedule, is fully documented and publicly available. This transparency provides attackers with comprehensive knowledge of when and where BGP routes are collected, as well as how the inferences are generated. Consequently, attackers do not need to compromise the VPs or interfere with ASRank's operations; they can simply manipulate the set of BGP paths observed at the VPs to influence the inference results.

Attacker's Goal. The attacker's objective is to manipulate ASRank's inferences, as defined in Def. 8 and 9, thereby affecting downstream, decision-relevant integrity. Specifically, by injecting carefully crafted AS triplets into ASRank's inference pipeline, the attacker induces directional and sustained changes in its inferred AS relationships, which are then distributed to and consumed by downstream systems. Although CAIDA cautions that ASRank outputs are not ground truth, downstream systems such as Cloudflare [3], GRIP [13], and BEAM [11] nevertheless rely on these inferences as trusted inputs for operational decisions, including anomaly detection, traffic engineering, and routing selection. Consequently, adversarially manipulating ASRank's inferred relationships can affect downstream decisions that depend on them. Accordingly, attack success is measured at the ASRank layer, by whether the attacker induces the intended ASRank inference changes, irrespective of direct data-plane impact.

This objective captures a failure mode distinct from ASRank's incidental baseline noise. Baseline noise stems from sampling artifacts and typically manifests as non-targeted inference uncertainty. In contrast, the adversarial errors studied in this paper are targeted and directional. Unlike incidental noise, adversarial errors are controllable by design, can persist as long as the attacker maintains the required capability, and can induce specific changes, such as P2C flips, that incidental noise is unlikely to produce.

Attacker's Capability. The attacker controls at least one AS, denoted by X, which is connected to the Internet and capable of originating BGP announcements. This capability is standard in prior BGP security research [9, 23, 24] and reflects realistic operational conditions; in practice, it can be obtained by exploiting known router vulnerabilities [25–28] or by legitimately acquiring Internet routing resources through existing commercial allocation and transfer ecosystems [9, 23]. The attacker can freely select the AS paths it announces, in accordance with standard BGP propagation rules and routing policies. Specifically, the attacker can (i) craft AS paths that embed targeted AS triplets, and (ii) control the timing and duration of these announcements. We do not assume the attacker has the ability to modify routing data at VPs or suppress routes announced by other ASes. A detailed discussion of the practicality of obtaining AS-level capabilities and the incentives for such attacks is deferred to §6.

Realization Scenarios. We consider two realistic scenarios in which the above capability can be realized, reflecting common operational constraints.

Scenario A (self-contained) In this scenario, the attacker controls two ASes, X and Y, and at least one IP prefix. The attacker configures ROAs so that Y is the valid origin of the prefix, and uses AS X to announce BGP routes of the form: X - (crafted AS triplet sequence) - Y. where attacker-chosen AS triplets are embedded. Because the origin AS and ROAs are consistent, these routes pass ROV and propagate normally. This scenario allows the attacker to inject forged AS triplets without causing data-plane disruption and represents the cleanest realization of our threat model.

Scenario B (hijack-based) In this scenario, the attacker controls only a single AS X, and announces a forged route for a prefix legitimately owned by an innocent AS Y, again using an AS path of the form:
X - (crafted AS triplet sequence) - Y. Unlike Scenario A, the attacker does not control the prefix or AS Y, and the announcement therefore constitutes a prefix hi-

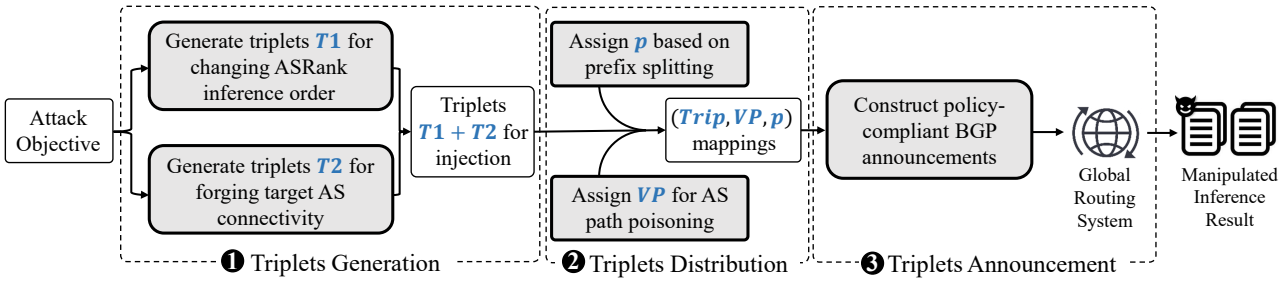


Figure 2: ASROGUE Overview.

jack. While this scenario may incur data-plane impact and is operationally riskier, it demonstrates that ownership of multiple ASes is not strictly required.

Assumptions of Defenses and Sanitization. We assume the presence of commonly deployed routing defenses, including ROV and Peerlock-lite. We further assume ASRank's standard sanitization procedures, which filter paths with AS loops, separated clique ASes, or obvious poisoning artifacts. We do not assume the deployment of path-validation mechanisms such as BGPsec or ASPA at Internet scale.

4 The ASROGUE Attack

In this section, we propose ASROGUE, a manipulation attack against ASRank-based AS relationship inference.

4.1 Overview

Figure 2 illustrates the high-level workflow of ASROGUE. Given a specified attack objective, ASROGUE injects carefully crafted AS triplets into ASRank's inference pipeline via the global routing system, ultimately manipulating the inference results. The attack proceeds in three stages: (i) generating the AS triplets needed for the target manipulation; (ii) assigning these triplets to prefixes and vantage points; and (iii) announcing them through policy-compliant BGP routes.

The core idea behind ASROGUE is to exploit ASRank's order-dependent inference process. By introducing a small number of strategically selected AS triplets, an attacker can rearrange the inference order and ensure that specific triplets are evaluated at the critical moment to trigger a desired inference. To achieve this, ASROGUE adaptively crafts two sets of AS triplets: those that disrupt the inference order through transit degrees, and those that forge the target AS connectivity once the order becomes favorable (§4.2). To facilitate the scalable injection of these crafted triplets, ASROGUE subdivides controlled prefixes and uses AS-path poisoning to steer announcements to different vantage points, assigning each AS triplet to a unique prefix–VP combination (§4.3). It then embeds the crafted triplets into policy-compliant BGP announcements that circumvent RPKI/ROV, Peerlock-lite, and ASRank's preprocessing, ensuring that the injected triplets are preserved in the observed data (§4.4). We now elaborate on each step in detail.

4.2 Triplets Generation

We first generate candidate AS triplets for manipulating ASRank. The central challenge is to craft triplets that steer ASRank toward a target inference outcome while introducing only minimal, hard-to-detect changes to the observed routing data. Because ASRank's decisions are jointly governed by (i) the inference order induced by transit degrees and (ii) triplet-based heuristics, successful manipulation must tightly control both factors simultaneously.

ASROGUE tackles this challenge by breaking triplet generation into two explicit technical goals. To control when a target inference is made, ASROGUE generates order-perturbation triplets that adjust transit degrees and position selected ASes favorably in ASRank's traversal order. To control what inference is made once that order is reached, ASROGUE generates connectivity-forcing triplets that satisfy ASRank's triplet-based heuristics and trigger the desired relationship inference. Guided by this principle, we devise triplet-generation strategies tailored to our two attack objectives.

P2C Manipulation. To generate AS triplets for P2C manipulation, ASROGUE operates against a *mirrored ASRank pipeline*, as illustrated in Figure 3. This pipeline locally replicates CAIDA's ASRank inference process using BGP data collected from the same set of vantage points during a recent reference period. Because ASRank's implementation, sanitization rules, and inference schedule are public, the attacker can precisely reproduce ASRank's full runtime state and inference order. This white-box view enables ASROGUE to reason explicitly about which AS triplets are required to induce a target inference. As we show later in §5.5, the freshness of the reference data is not a sensitive parameter.

Given a target AS pair (a,b) and the attack objective, *i.e.,* $I_{\text{att}}(a,b) = \text{P2C}$, ASROGUE generates two sets of AS triplets in parallel. The first set perturbs ASRank's inference order to ensure that a is visited before b during Step 5. The second set forges a valid inference chain that deterministically induces the desired P2C relationship once this order is established.

As shown in Figure 3(I), the goal of the first set is to ensure

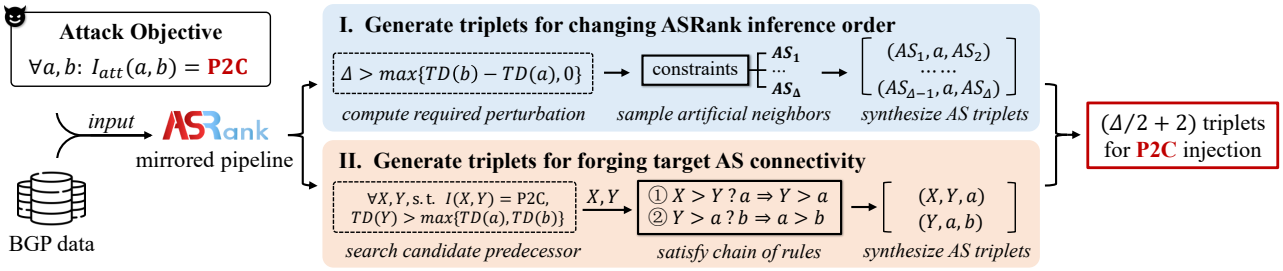


Figure 3: Triplets generation for P2C injection.

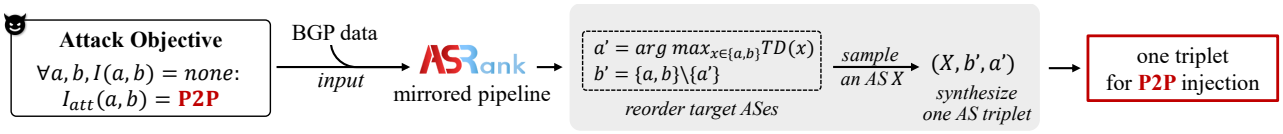


Figure 4: Triplets generation for P2P injection.

$\mathrm{TD}(a) > \mathrm{TD}(b)$ so that a precedes b in ASRank's traversal. Using the mirrored pipeline, ASROGUE computes the required transit degree perturbation:

$$\Delta = \min\{d \in 2\mathbb{N} \mid d > \max\{\mathrm{TD}(b) > \mathrm{TD}(a), 0\}\}. \quad (3)$$

where Δ is constrained to be even because each synthesized AS triplet contributes two distinct neighbors to the middle AS. Intuitively, Δ captures the smallest number of a's new neighbors required to deterministically flip the inference order between a and b.

Given Δ, ASROGUE samples Δ artificial neighbors $AS_1, \ldots, AS_\Delta$, each satisfying the following constraints:

1. It is not a Tier-1 or clique AS, since triplets involving such ASes are likely to induce non-consecutive clique adjacencies in propagated AS paths, which are filtered by Peerlock-lite or ASRank's sanitization.
2. It does not appear on any AS path from the attacker's announcing AS to any vantage point, as including such an AS would create an AS-path loop during propagation and prevent the forged route from reaching the VPs.
3. It is not already adjacent to a in any baseline AS triplet where a is the middle AS, since such ASes are already counted toward $\mathrm{TD}(a)$ and do not increase it.
4. Its transit degree is below a threshold (100 in our practice), which reduces operational visibility and lowers the likelihood of triggering routing defenses.

Under these constraints, ASROGUE synthesizes $\Delta/2$ order-perturbation triplets of the form: (AS_1, a, AS_2), (AS_3, a, AS_4), $\ldots$, $(AS_{\Delta-1}, a, AS_\Delta)$, which collectively increase $\mathrm{TD}(a)$ by Δ and guarantee that a is visited before b during inference.

The second set, as shown in Figure 3(II), constructs triplets that force ASRank to infer $a > b$ once the traversal order is favorable. Using the mirrored pipeline, ASROGUE searches for a candidate predecessor AS pair (X, Y) such that $I(X, Y) = \mathrm{P2C}$ under baseline inference and $\mathrm{TD}(Y) > \max\{\mathrm{TD}(a), \mathrm{TD}(b)\}$. This ensures that the relationship $X > Y$ is inferred before a is visited. Given such a pair, ASROGUE synthesizes two connectivity-forcing triplets that satisfy ASRank's inference heuristics in sequence: First, the triplet (X, Y, a) satisfies $X > Y$? a, which triggers the inference $Y > a$; Second, the triplet (Y, a, b) satisfies $Y > a$? b, which in turn triggers the target inference $a > b$. Together, these two triplets deterministically induce the desired P2C relationship once ASRank processes a and b.

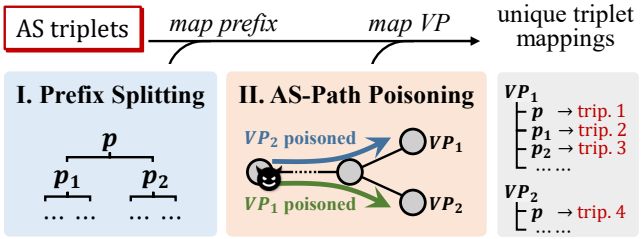


Figure 5: Triplets distribution over IP prefixes and VPs.

In total, P2C manipulation for a target pair (a, b) requires $\Delta/2 + 2$ synthesized AS triplets. When injected into ASRank's input data, these triplets jointly ensure that a is visited before b and that ASRank's inference heuristics necessarily conclude $a > b$, regardless of the baseline inference outcome.

P2P Manipulation. We next generate AS triplets for P2P manipulation, as illustrated in Figure 4. The attack objective is to induce a P2P relationship between a target AS pair (a, b) that has no baseline inference. Recall that, except for those among clique ASes, ASRank assigns P2P relationships mostly at the final stage (Step 11) after all P2C inference rules and special-case handling (Steps 6–10). The key challenge is therefore to introduce connectivity between a and b without

triggering any earlier P2C inference.

Using the mirrored ASRank pipeline, ASROGUE first orders the target ASes by transit degree:

$$a' = \arg\max_{x \in \{a,b\}} \mathrm{TD}(x), \quad b' = \{a,b\} \setminus \{a'\}. \tag{4}$$

This ensures that a' is visited before b' during Step 5. ASROGUE then randomly samples an AS X and synthesizes a single AS triplet (X, b', a'), which introduces an apparent adjacency between b' and a' while avoiding satisfying ASRank's P2C inference heuristics. When a' is visited, the relationship between X and b' is unresolved, and thus neither $X > b'$? a' nor $X - b'$? a' applies. As a result, no P2C inference between b' and a' is triggered in Step 5. Except with negligible probability due to special-case handling in Steps 6–10, the forged connectivity between b' and a' (equivalent to connectivity between a and b) is ultimately inferred as P2P at Step 11.

4.3 Triplets Distribution

We next need to realize the crafted AS triplets so they actually take effect. Specifically, each triplet must appear in the BGP routes observed by the vantage points (VPs) from which ASRank collects its input data, and the injected evidence must remain mutually non-interfering. As defined in Def. 1, each ASRank input is uniquely indexed by a prefix–VP pair and contributes AS-path–derived triplets independently. Consequently, multiple crafted triplets can coexist as long as they are carried by distinct prefix–VP combinations. The core challenge thus becomes provisioning a sufficient number of independent prefix–VP pairs to host all crafted triplets, while controlling which VP observes which triplet.

ASROGUE addresses this challenge by scaling the number of usable prefix–VP combinations using two complementary techniques. As shown in Figure 5, we first apply *prefix splitting* to expand the attacker-controlled prefix space under common route filtering constraints. We then use *AS-path poisoning* to steer route propagation such that different VPs observe disjoint sets of crafted triplets. Together, these techniques enable scalable, controlled, and non-conflicting mapping from AS triplets to prefix–VP pairs.

Prefix Splitting. ASROGUE exploits the hierarchical structure of IP prefixes to increase the available prefix space. Given an attacker-controlled prefix of length L, ASROGUE subdivides it into all routable subprefixes whose lengths do not exceed a maximum allowable length $L_{\max}$. In practice, prefixes longer than /24 in IPv4 or /48 in IPv6 are widely filtered and ineffective for propagation; we therefore restrict splitting to $L_{\max} = 24$ (IPv4) and $L_{\max} = 48$ (IPv6), unless tighter constraints apply.

Formally, for $L \le L_{\max}$, the number of usable prefixes after split is

$$S(L, L_{\max}) = 2^{(L_{\max} - L + 1)} - 1. \tag{5}$$

which corresponds to all nodes in the prefix's binary subtree down to depth $L_{\max}$. For example, an IPv4 /23 prefix yields one /23 and two /24 prefixes, resulting in $S(23,24) = 3$ usable prefixes. Each can be announced independently and contributes a distinct route when observed at a VP. Prefix splitting applies to both attack scenarios in §3.3. In the self-contained scenario, the attacker controls the prefix and can originate all subprefixes with valid ROAs, ensuring ROV compliance. In the hijack-based scenario, $L_{\max}$ is determined by the maximum prefix length authorized in the victim prefix's ROA, if present, and the same formulation applies.

By expanding a single prefix into $S(L, L_{\max})$ prefixes, ASROGUE substantially increases the number of routes available for carrying crafted triplets. One potential concern is that upstream providers may aggregate prefixes, reducing their visibility at VPs. In practice, such aggregation is neither universal nor enforced: /24 IPv4 and /48 IPv6 prefixes are widely propagated due to traffic engineering, multihoming, and policy requirements. Moreover, an attacker can intentionally avoid aggregation by announcing subprefixes via different upstreams or using standard BGP communities that discourage aggregation.

AS-path Poisoning. Prefix splitting alone does not increase the number of *distinct* AS triplets seen by ASRank: since ASRank operates on the union of triplets across all VPs, identical AS triplets observed at multiple VPs contribute no additional information. To inject multiple crafted triplets simultaneously, the attacker must therefore ensure that different VPs observe *disjoint* sets of triplets.

ASROGUE achieves this using AS-path poisoning to control route propagation at the granularity of individual VPs. Given a triplet intended to be observed by VP_k, ASROGUE additionally includes the ASNs of all other VPs $\{VP_i \mid i \neq k\}$ in the route that carries the triplet. By BGP loop detection, each poisoned VP_i rejects the route upon seeing its own ASN in the AS path, while VP_k, whose ASN is not included, accepts the route. As a result, each crafted triplet is observed by exactly one intended VP.

By combining prefix splitting with AS-path poisoning, ASROGUE creates a large set of independent prefix–VP combinations. If $S(L, L_{\max})$ prefixes are available after splitting and N VPs are selectively targeted, then over daily RIB snapshots of D days, the attacker can inject up to $D \cdot N \cdot S(L, L_{\max})$ distinct AS triplets without conflict, each mapped to a unique prefix–VP–day combination. In practice, N is bounded by AS-path length limits and the number of VPs used by ASRank, while D is bounded by ASRank's five-day aggregation window. However, the number of triplets required for a target manipulation is typically small, allowing ASROGUE to operate well within these bounds. We show in §5.3 that even modest values of N and D suffice for effective manipulation.

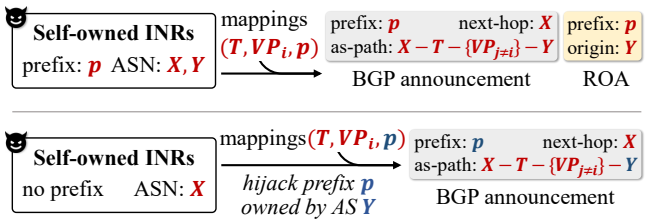


Figure 6: Triplets announcement under different scenarios.

4.4 Triplets Announcement

Once the crafted AS triplets have been generated and assigned to unique prefix–VP pairs, the final step is to instantiate these mappings as BGP announcements that ASRank will ingest. The key challenge is to engineer routes whose AS paths embed the intended triplets, propagate to (and ideally only to) the designated VPs, and remain effective in the presence of operational routing defenses and ASRank's input sanitization.

ASROGUE addresses this by translating each mapping (T, VP_i, p) into a policy-compliant BGP route whose AS path embeds triplet T and selectively propagates to VP_i. Figure 6 illustrates this process under the two realization scenarios from §3.3. In both cases, selective visibility is enforced using AS-path poisoning and prefix-origin consistency is ensured.

In the *self-contained scenario* (upper panel), the attacker controls two ASes, X and Y, and a prefix p, and registers an ROA that authorizes Y as the origin of p. For each triplet mapping, the attacker announces a route from X with AS path $X - T - \{VP_j \mid j \neq i\} - Y$. All non-target VPs reject the route due to loop detection, while VP_i accepts it. The route passes ROV and hygienic filtering, and the embedded triplets survive ASRank's preprocessing.

In the *hijack-based scenario* (lower panel), the attacker controls only AS X and forges a route for a prefix owned by an innocent AS Y, using the same AS-path structure. This leads to a prefix hijack and may incur data-plane impact. When announcing the same prefix (rather than a more specific one), the forged route may also be less favorable than the legitimate route due to its longer AS path, which can limit its visibility at VPs and thus reduce attack effectiveness. Nevertheless, this limitation can be mitigated by strategies such as targeting unrouted prefixes, which we leave for future exploration.

By converting abstract triplet–prefix–VP mappings into concrete BGP announcements, ASROGUE finalizes the attack pipeline and enables precise, scalable manipulation of ASRank inference using only standard routing mechanisms.

5 Evaluation

5.1 Experimental Setup

We evaluate ASROGUE primarily through offline execution against a locally mirrored ASRank pipeline using real-world BGP routing data, and complement this evaluation with live experiments to validate real-world triplet propagation.

Table 2: Top 10 ASes by transit degree.

ASN	Ownership	TD	Tier-1?
6939	Hurricane Electric	10884	✗
174	Cogent Communications	6870	✓
3356	Lumen (Level 3)	6747	✓
1239	Sprint Wireline (Cogent)	4874	✓
35280	F5 Networks SARL	4749	✗
38255	CERNET	4101	✗
137409	GSL Networks Pty LTD	3912	✗
15830	Equinix, Inc.	3370	✗
25091	IP-Max SA	3178	✗
9002	RETN Limited	2932	✗

Unless stated otherwise, our offline evaluation uses the October 2024 CAIDA Serial-1 AS relationship dataset, which includes 84,766 ASes and 490,318 relationships derived from 592,720,031 empirical routes. For each monthly release, CAIDA aggregates the first five days of BGP RIB snapshots from 33 fixed RouteViews and RIPE RIS collectors and processes them with ASRank. We replicate this process locally by retrieving RIB data from the same collectors and time windows via the official RouteViews and RIPE RIS archives, and running a Python-refactored version of ASRank. This refactoring is based on the methodology described in the ASRank paper [17] and the publicly available source code [22]. We validate our implementation by comparing its inference results on the October 2024 dataset with CAIDA's published AS relationship data and confirm that our implementation accurately reproduces ASRank's inference behavior.

To model attack execution, we randomly select a pair of P2C ASes as the attacker that announce forged routes and simulate their propagation under Gao-Rexford policies. A forged route contributes to inference only if it is visible at the intended VP under policy-compliant propagation, matching the constraints faced by a real attacker. All results reported in §5.2–§5.5 are obtained using this offline setup. In addition, we validate real-world triplet propagation separately in §5.6 through controlled experiments on the PEERING testbed.

Notably, while Gao-Rexford policies capture shortest-path preference, longer AS paths do not invalidate ASROGUE, because success in most cases does not depend on best-path selection. In Scenario A, the attacker originates its own prefix, so there are no competing routes other than the attacker's. In Scenario B, the attacker can announce a more-specific prefix, sidestepping competition with legitimate routes. However, when using same-prefix announcements, Scenario B is subject to best-path competition and should be interpreted as a stricter, more BGP-realistic setting.

Table 1: Inferred P2C relationships by AS size combination.

Cust. \ Prov.	Large	Medium	Small	0-Size
Large	0.37% (1988)	0.76% (4085)	0.05% (296)	0.05% (289)
Medium	6.16% (33304)	20.6% (111040)	1.99% (10777)	2.07% (11188)
Small	3.19% (17245)	11.0% (59316)	1.19% (6438)	1.08% (5819)
0-Size	13.8% (74315)	31.4% (169663)	3.79% (20489)	2.60% (14072)

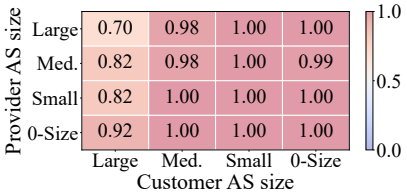


Figure 7: P2C manipulation succ. rate.

5.2 Manipulation Effectiveness

"How often can an attacker force ASRank to infer a relationship of its choosing?"

We evaluate the intrinsic manipulability of ASRank by measuring the success rate of ASROGUE without limiting the number of forged routes used to inject attacker-crafted AS triplets. This establishes an upper bound on an attacker's ability to control ASRank's inference outcomes; attack cost and resource constraints are analyzed separately in §5.3. An attack is considered successful if, after injection, ASRank infers the attacker-specified relationship type for the target AS pair, differing from the relationship inferred under benign inputs.

P2C Manipulation Effectiveness. We first evaluate the effectiveness of ASROGUE in inducing targeted P2C inferences. To obtain a finer view, we group ASes by customer cone size based on the established classification [29]: we classify ASes as *0-size* if they have no customers, *small* if they have 1 or 2 customers, *medium* if they have between 3 and 180 customers, and *large* if they have more than 180 customers, yielding 16 provider-customer size combinations. For each combination, we randomly sample 100 baseline P2C relationships and attempt to reverse the inferred direction using ASROGUE. Figure 7 shows that ASROGUE succeeds for more than 80% of targets in 15 out of the 16 combinations, and exceeds 98% whenever the provider is not large. In particular, relationships involving 0-size, small, or medium providers, which constitute the majority of inferred relationships (see Table 1), are almost universally manipulable, with success rates reaching 99.9%. Only the large-large category, accounting for 0.37% of inferred relationships, exhibits substantially lower success. Weighting each combination's success rate by its prevalence yields an overall manipulation success rate of 96.7% under Scenario A, where the attacker originates its own prefix and does not face competing routes. Scenario B is stricter: same-prefix announcements must win best-path selection, whereas more-specific announcements can avoid that competition.

Limiting Cases. Notably, the rare cases where ASROGUE fails are not random. They occur when at least one AS in the target pair has an exceptionally large transit degree, such that no predecessor relationship can be found to trigger the inference heuristics. In most failed cases, at least one AS has a transit degree exceeding 2,000. For example, attacks involving AS 6939 (TD = 10,884), AS 15830 (TD = 3,370), and AS 137409 (TD = 3,912) almost always fail; all three rank among the highest-transit ASes in the Internet (see Table 2). This pattern reveals a key insight: ASROGUE is not limited by ASRank's robustness, but by the absence of predecessor inference that could precede the target pair in ASRank's traversal order. When such extreme transit-degree ASes are involved, the traversal order becomes effectively fixed. Outside this narrow regime, however, ASRank's greedy and order-dependent inference logic remains highly susceptible to manipulation.

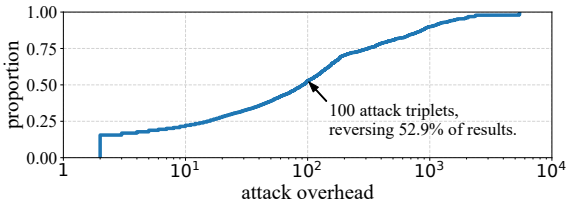


Figure 8: Fraction of manipulable ASRank-inferred P2C relationships as a function of attack overhead.

P2P Manipulation Effectiveness. We next evaluate the effectiveness of ASROGUE in inducing P2P relationships between AS pairs that have no baseline inferred relationship. Because over 84% of ASes have zero transit degree, uniform random sampling would overwhelmingly select pairs of zero-transit-degree ASes and underrepresent scenarios involving higher-degree ASes. To obtain balanced coverage, we construct 200 target pairs by design: 100 pairs where exactly one AS has non-zero transit degree, 50 pairs where both ASes have non-zero transit degree, and 50 pairs where neither AS has non-zero transit degree. ASROGUE succeeds in 99.5% of all cases, and all targets involving at least one zero-transit-degree AS are manipulated successfully. The only failure occurs for a pair where both ASes have identical non-zero transit degree, which triggers a corner condition in ASRank that infers a fallback P2C relationship. Such failures are rare and arise only under this narrowly defined condition.

Takeaway. ASROGUE can manipulate 96.7% of ASRank-inferred P2C relationships, with near-perfect success for small and medium ASes. Failures are confined to a set of ASes with among the highest transit-degrees. P2P relationships are even more reliable to manipulate, succeeding in almost all cases.

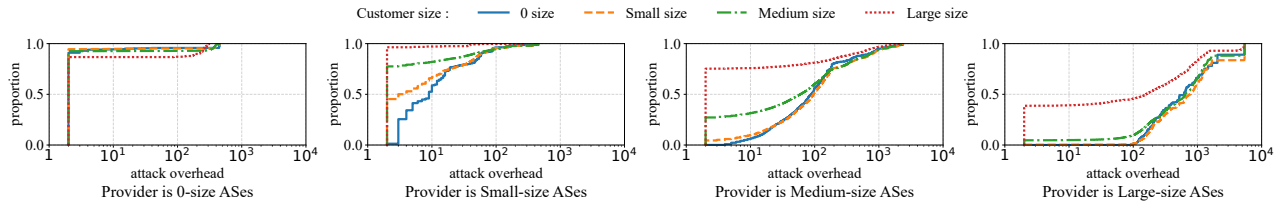


Figure 9: Fraction of manipulable ASRank-inferred P2C relationships under bounded overhead, grouped by provider AS size.

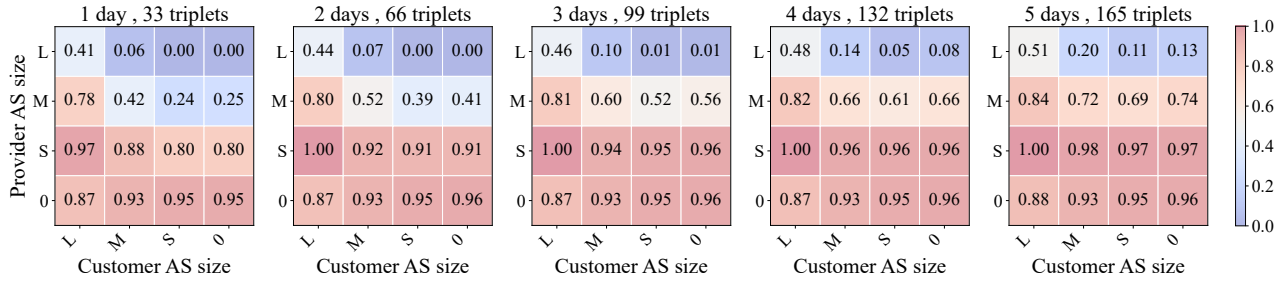


Figure 10: Fraction of manipulable ASRank-inferred P2C relationships under ASRank's aggregation bounds.

5.3 Attack Cost and Scalability

"How many forged announcements does an attacker need to manipulate ASRank at scale?"

We now evaluate the practical cost of ASROGUE by quantifying how many forged BGP announcements an attacker must inject to manipulate ASRank's inference results, and how the impact scales as attack overhead increases. In ASROGUE, each crafted AS triplet is carried by a distinct policy-compliant BGP announcement mapped to a unique prefix-VP pair; we therefore use the number of forged announcements as the operational measure of attack overhead. We focus on P2C manipulation in particular, because P2P manipulation requires exactly one triplet and succeeds almost universally (see §5.2). As derived in §4.2, manipulating a target P2C relationship requires $\Delta/2+2$ forged announcements, where Δ is the minimum transit-degree perturbation needed to flip the relative traversal order of the target AS pair in ASRank.

Effectiveness under Bounded Overhead. We first quantify how many ASRank-inferred P2C relationships become manipulable as the attack overhead is capped. Figure 8 plots the fraction of relationships that can be reversed as a function of the number of forged announcements. Even under tight constraints, ASROGUE scales quickly: 10 announcements make roughly 20% of inferred relationships manipulable, 30 announcements increase that fraction to about 30%, and 100 announcements suffice to compromise nearly half of ASRank's P2C inferences. In practical terms, this overhead is well within operational reach. For example, an attacker controlling a single /44 IPv6 prefix can split it into 31 routable prefixes; combining these prefixes with a few dozen VPs already yields hundreds of distinct prefix-VP combinations, sufficient to reach the 100-announcement regime shown here and influence a large fraction of ASRank's inference output.

We also evaluate a stricter setting that further limits the attacker's capability: the attacker controls only a single prefix, poisons at most four vantage points (VPs), and restricts forged AS paths to at most 10 hops. Under these constraints, the success rate drops to 25%, still partially effective. This 10-hop cap is conservative, as the October 2024 CAIDA RIB sample contains more than 100,000 AS paths longer than 10 hops, with a maximum observed length of 27.

Notably, the impact of bounded attack overhead is highly uneven across provider sizes. Figure 9 shows the manipulable fraction of P2C relationships by provider category. Relationships involving 0-size providers are particularly fragile: with only 10 forged announcements, over 90% can be reversed. For small providers, 30 announcements compromise over 80% of inferred relationships, while for medium providers, 200 announcements already affect over 70%. In contrast, relationships involving large providers demand substantially higher overhead, reflecting their dominant transit-degree advantage in ASRank's traversal order, yet this advantage does not imply immunity. With 300 announcements, more than 30% of such relationships can still be manipulated. Moreover, in 14.3% of all inferred P2C relationships, the customer already has a higher transit degree than the provider, allowing reversal with only two forged announcements.

ASRank's Aggregation Bounds. We further instantiate the above analysis under a deliberately strict model aligned with ASRank's inference pipeline. ASRank aggregates daily RIB snapshots over a five-day window ($D \leq 5$) from 33 collectors. To obtain a conservative lower bound, we assume at most one effective VP per collector contributes distinct forged routes,

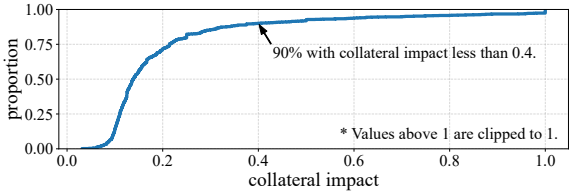


Figure 11: Collateral impact of ASROGUE.

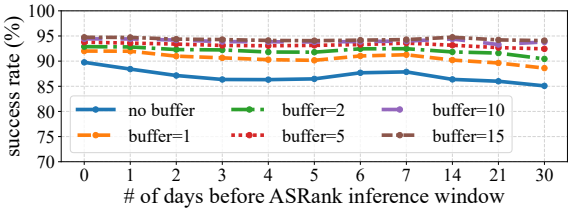


Figure 12: Attack success rate using stale RIB data.

yielding $N = 33$. Under this setting, a forged route observed throughout the aggregation window contributes 33 distinct AS triplets, while announcing different forged routes on different days accumulates additional triplets across snapshots. We therefore evaluate ASROGUE for $D = 1$ to 5, corresponding to total overheads from 33 to 165 forged triplets. As shown in Figure 10, even under this strict assumption, ASROGUE remains highly effective: a single day of injection already reverses over 80% of P2C relationships involving 0-size or small providers, while extending the injection window steadily increases manipulability for relationships involving medium providers, reaching approximately 70% with five days. The impact on relationships involving large providers remains limited under these bounds, but is consistently non-zero.

Takeaway. ASROGUE is highly effective at low cost: tens of forged announcements already make 20-30% of P2C inferences manipulable, and 100 announcements suffice to compromise nearly half. Only relationships involving high-transit-degree providers exhibit meaningful resistance.

5.4 Attack Precision and Collateral Impact

"How much unintended inference disturbance does ASROGUE *cause?"*

A natural concern is whether ASROGUE causes inference changes beyond those inherently required. We therefore evaluate the precision of ASROGUE by quantifying its *collateral impact*, defined as the number of unintended AS relationship inferences that change beyond the predictable effects induced by the attack design itself. This allows us to assess whether ASROGUE operates as a coarse disruption or as a targeted attack with bounded and predictable side effects.

Isolating Collateral Impact. We first account for the inference changes that are *inherent* to a successful P2C manipulation. According to §4.2, a target P2C manipulation requires injecting $\Delta/2+2$ crafted AS triplets. Each triplet, when embedded in a policy-compliant forged route, deterministically introduces four AS adjacencies. Excluding the known predecessor relationship, this results in exactly $4(\Delta/2+2)-1 = 2\Delta+7$ predictable inference changes, which we treat as the unavoidable, by-design footprint of the attack. We then execute ASROGUE against each target and count the total number of AS relationship inferences that differ from the baseline. The collateral impact is measured by the excess beyond $2\Delta+7$, normalized by $2\Delta+7$, thereby isolating unintended inference changes caused by secondary cascading effects, such as transit-degree reordering.

Figure 11 shows the distribution of collateral impact across all evaluated P2C manipulation targets. In the vast majority of cases, ASROGUE exhibits high precision: over 90% of attacks incur a collateral impact below 0.4, indicating that most inference changes are fully accounted for by the attack's unavoidable, by-design footprint. This confirms that ASROGUE typically induces only minor and predictable side effects. We do, however, observe rare corner cases with exceptionally large collateral impact. In the most extreme case, the collateral impact exceeds 22.67, caused by a cascading conversion of many P2P relationships into P2C relationships. Although uncommon, these cases highlight the inherent fragility of ASRank's greedy, order-dependent inference logic, where small perturbations can trigger large-scale cascades.

Takeaway. Over 90% of ASROGUE attacks incur a collateral impact below 0.4, demonstrating high precision in practice. Rare outliers exhibit large cascades, revealing inherent fragility in ASRank's order-dependent inference.

5.5 Attack Robustness to Temporal Drift

"How sensitive is ASROGUE *to temporal drift between the attacker's view of BGP data and the actual ASRank input?"*

In practice, an attacker must inject forged routes *before* ASRank captures RIB snapshots. As a result, the attacker's view of routing data necessarily precedes and may differ from the data actually used by ASRank. We therefore evaluate how such temporal mismatch affects the effectiveness of ASROGUE, and whether an attacker can compensate for it.

Effectiveness under Stale RIB data. We evaluate ASROGUE using RIB data collected on the same day as ASRank's inference window, as well as data collected one to seven days, two weeks, three weeks, and one month earlier. Figure 12 reports the resulting P2C manipulation success rates (blue line) and shows that ASROGUE is highly robust to temporal drift. Using RIB data collected within the preceding week yields an average success rate of 87.5%, comparable to the approximately 90% success rate obtained using same-day data. Minor variations are observed across nearby time points due to natural routing variability. As the temporal gap increases further, success rates decline gradually, from about 90% to roughly

85% after one month, but remain high overall. These results indicate that the attack does not require precise or up-to-date knowledge of the exact BGP data used by ASRank.

Compensating with Triplet Buffering. An attacker can mitigate the effects of temporal drift by conservatively adding a small buffer to the number of crafted triplets for transit-degree perturbation. We evaluate buffer sizes of 1, 2, 5, 10, and 15 additional triplets. As shown in Figure 12, even a buffer of one triplet improves the success rate by about 3%. Increasing the buffer steadily improves manipulation effectiveness, and a buffer of 15 raises the success rate from about 90% to 94.7%, closely approaching the upper bound observed when using the exact RIB data ingested by ASRank. Importantly, this buffering strategy does not require knowledge of ASRank's internal state and incurs only modest additional attack cost.

Takeaway. ASROGUE is robust to temporal drift: using week-old RIB data still achieve about 87% success, and adding a buffer of 15 triplets raises success to 94.7%. Temporal uncertainty therefore provides little defense for ASRank.

5.6 Real-World Feasibility of ASROGUE

"Can attacker-crafted AS triplets be actually propagated and observed in today's Internet?"

We validate the real-world feasibility of ASROGUE by demonstrating that crafted AS triplets can propagate in the live Internet under standard routing policies. To this end, we conduct controlled experiments on the PEERING testbed [30]. PEERING allows us to validate the propagation of clean, short forged paths, but, for ethical and operational reasons, we cannot empirically test the propagation of heavily poisoned long paths or full-scale attacks. Such sustained poisoning is nevertheless plausible in the wild; for example, RouteViews records show that AS7619 poisoned 47 ASes over roughly three years, which supports the practical feasibility of this attack primitive. We obtain AS47065, AS61574, AS61575, and AS61576, and prefix 184.164.252.0/24. For ethical reasons, and to avoid any unintended impact on third-party networks, all forged AS paths are constructed exclusively from ASes under our control, and the announced prefix carries no production traffic.

In our setup, AS61574 is configured as the ROA-authorized origin for the prefix, while AS47065 acts as the attacker-controlled announcing AS. We embed crafted AS triplets via standard AS-path prepending, resulting in paths of the form "47065 61576 61575 61574" for announcement. These announcements are policy-compliant, loop-free, and consistent with common hygienic filtering practices, thereby faithfully reflecting the constraints faced by a real attacker operating within today's routing ecosystem. We conduct the experiment on January 13, 2026. Within minutes of the announcements, we observe the forged routes on multiple independent VPs across Asia, Europe, and North America. This confirms that the attacker-crafted triplets can propagate and become visible across the Internet without triggering filtering or rejection. We provide detail traces in Appendix C.

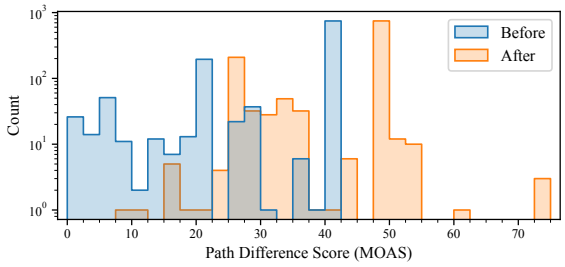


Figure 13: BEAM path difference scores before and after ASRank manipulation.

Takeaway. Forged AS triplets can effectively propagate in today's Internet using standard, policy-compliant BGP announcements, confirming that the core primitive of ASROGUE is operationally practical beyond offline evaluation.

5.7 Downstream Impact of ASROGUE

"What happens when manipulated AS relationships are consumed by downstream systems?"

We examine whether ASROGUE affects downstream systems that treat inferred AS relationships as trusted inputs. We study three representative usage patterns: model training in BEAM [11], rule-based anomaly classification in GRIP [13], and incident attribution in a real route-leak case reported by Kentik [31]. These case studies show that small, targeted ASRank manipulations can propagate into concrete downstream errors, including shifted detection thresholds, incorrect classifications, and misleading attribution.

BEAM Case. BEAM is a machine-learning-based BGP anomaly detection system that uses AS relationships as semantic priors [11]. We evaluate whether targeted ASRank manipulation can affect BEAM's learned detection behavior by training BEAM under two settings: one using baseline ASRank inferences, and the other using inferences after ASROGUE flips five strategically selected P2C relationships. We then use the trained models to compute the *path difference scores* of representative Multi-Origin AS (MOAS) route changes. BEAM uses this score to quantify how drastic (and therefore potentially anomalous) a route change is. Figure 13 shows that these five relationship flips substantially shift the path difference score distribution. If we set the detection threshold to produce no false positives under the baseline model, more than 25% of the same MOAS cases exceed this threshold under the manipulated model. This means that a small number of ASRank manipulations can materially alter BEAM's threshold-based decisions, causing otherwise benign route changes to be flagged as anomalous or changing how anomalies are prioritized.

GRIP Case. GRIP is a production BGP anomaly detection system that combines multiple signals, including IRR, RPKI, and AS business-relationship heuristics, to classify routing anomalies [13]. One such heuristic checks whether newly observed origin ASes have plausible business relationships with the affected ASes, *e.g.,* whether all newcomers are inferred as customers or providers. We evaluate whether ASRogue can alter such rule-based decisions by targeting suspicious MOAS events extracted from GRIP's October 2024 production logs. We identify 10 suspicious MOAS events for which changing a small number of ASRank-inferred relationships would affect GRIP's classification logic, and then apply ASRogue to induce those relationship changes. In all 10 cases, ASRogue successfully manipulates the required AS relationships and changes the classification outcome that GRIP would derive from this heuristic. The attack requires only modest overhead, with an average of 5.5 crafted triplets per event. These results show that ASRank manipulation can propagate into production anomaly-detection logic and cause events to be interpreted differently.

Kentik Case. We further examine a real route-leak incident reported by Kentik on February 11, 2025 [31]. The incident involved AS52863 (TD = 115), which leaked routes originated by AS7195 (TD = 2,558) to AS400282 (TD = 22). Under baseline ASRank inference, the relevant relationships are inferred as AS7195 $>$ AS52863 and AS400282 $>$ AS52863, which correctly identify AS52863 as the AS where the leaked route is propagated across inconsistent provider-customer directions. We show that ASRogue can alter this interpretation with only two crafted AS triplets. After injection, ASRank infers AS7195 $>$ AS52863 $>$ AS400282, changing the relationship between AS52863 and AS400282 and turning the inferred structure into a different provider-customer chain. If consumed by downstream incident-analysis systems, this manipulated inference could shift attribution away from the actual leaking AS52863 and toward the innocent AS400282, misdirecting operational response. This case illustrates that ASRank manipulation can effectively affect the interpretation of real routing incidents.

Takeaway. ASRogue has significant downstream impact. With only a few crafted triplets, ASRogue can shift BEAM's detection behavior, alter GRIP's anomaly classification, and mislead attribution in a real route-leak incident. These results show that attacks on AS relationship inference can affect operational security decisions, not merely offline topology datasets.

6 Discussion

Mitigations. Mitigating ASRogue requires both strengthening ASRank's inference robustness and constraining an attacker's ability to inject forged triplets at the routing layer. At the inference level, robustness can be improved by moving from single-observation decisions to persistence-weighted inference, in which triplets must remain visible across multiple days before they are allowed to meaningfully influence the final outcome. This reduces the impact of isolated forged observations, but it also comes at a cost: in our analysis, a 5-day persistence filter removes about 600,000 triplets (6.65%) and about 22,000 inferred relationships (4.93%). Such filtering therefore improves stability, yet it also suppresses a nontrivial amount of legitimate evidence. These mitigations raise attacker cost and exposure, but they do not make ASRogue infeasible for a persistent adversary; long-lived poisoning has occurred in practice, as reflected by RouteViews records showing AS7619 poisoned 47 ASes over nearly three years. A complementary defense is randomized-window corroboration, in which ASRank cross-validates triplets across randomly shifted windows and assigns greater confidence to adjacencies that recur under multiple sampling offsets. This approach reduces sensitivity to short-lived poisoning, but it also increases inference latency and computational overhead, since multiple windowed runs must be maintained and reconciled. Organization-level constraints provide another useful guardrail: incorporating AS affiliation and business structure can flag announcements that are implausible under known economic relationships, and can prevent isolated triplets from disproportionately influencing outcomes. Because ASRogue injects forged paths without suppressing legitimate ones, monitoring for internal inconsistencies, such as unexpected relationship reversals or abrupt changes that fail persistence checks, can expose manipulation and trigger further validation. Overall, these defenses improve robustness against targeted manipulation at the cost of reduced data coverage, delayed inference responsiveness, additional computational overhead, and increased deployment complexity. At the routing layer, broader deployment of path-validation mechanisms can further limit an attacker's ability to inject forged triplets. BGPsec [32] can cryptographically authenticate AS paths in principle, but currently sees limited deployment and degrades sharply under partial adoption [33]. More recent proposals such as ASPA [34] and ASRA [35] offer more deployable alternatives by authorizing legitimate AS adjacencies, yet still remain in early-stage deployment and are vulnerable to known evasion techniques [15].

Attack Feasibility and Incentives. ASRogue is feasible for motivated adversaries and requires no exceptional capabilities. Control over an AS can be obtained either by compromising routers via known vulnerabilities [25–28] or by legitimately acquiring routing resources. The growing IP leasing and Bring Your Own IP (BYOP) ecosystem makes obtaining ASNs and routable IPv6 address blocks inexpensive, fast, and subject to minimal vetting [9]. For example, an attacker can acquire an ASN and multiple IPv6 prefixes through a sponsoring LIR for 270 EUR and achieve full Internet connectivity within days [23]. Once an adversary controls an AS, whether compromised or newly registered, it can originate BGP announcements and inject forged AS triplets, making ASRogue both

practical and economically accessible.

Manipulating AS relationship inference also offers clear incentives. A higher ASRank position can yield tangible commercial benefits by increasing visibility to potential peers and attracting customers who associate higher ranking with reliability [36]. It can also enhance an AS's perceived reputation and influence, shaping its role in partnerships, coordination, and policy discussions. These incentives make targeted ASRank manipulation a plausible and attractive strategy for adversaries seeking to amplify their standing.

Detectability. Although ASROGUE may be detectable, such detection does not necessarily neutralize it. In Scenario A, the attacker originates its own prefix, which creates no obvious origin conflict and typically evades existing detection systems. In Scenario B, the attack involves a hijack and may therefore be flagged by operators or automated detection tools. However, ASROGUE only requires the forged routes to be observed by a sufficient number of VPs within a short time window, and does not depend on sustained global propagation. Thus, even if the hijack is detected, the polluted observations can already have entered the data collection and inference pipeline, and can still affect the final ASRank outcome.

Impact on Prior Research. A critical open question is whether historical AS relationship datasets have already been influenced. Addressing this requires the retrospective identification of forged or inconsistent AS paths, such as detecting anomalous structural shifts in datasets over time. If such manipulations have occurred, they could lead to false anomaly detections, inaccurate measurements, and skewed conclusions about AS relationships and Internet topology. A thorough examination of historical datasets and a reassessment of prior findings are essential avenues for future research.

7 Related Work

AS Relationship Inference. Prior work on AS relationship inference has focused on heuristic methods [17, 37–39] and, more recently, learning-augmented approaches [18, 19, 21]. Early methods are largely based on the valley-free assumption [37] and the Gao-Rexford model [40], later extended with additional heuristics leveraging BGP attributes [38]. Among these, ASRank [17] emerges as the de-facto foundation, applying valley-free inference at Internet scale and providing the primary AS relationship datasets used by the community. Building on ASRank, UNARI [39] constructs an initial set of relationships and refines the remaining ones using statistical features and logistic regression, while Problink [18] generates probabilistic relationship vectors and iteratively applies a Naïve Bayes classifier. TopoScope [19] applies ASRank to obtain voting-based seeds, and then use Bayesian models to infer unresolved relationships. Across these heuristic and learning-augmented methods, ASRank remains a central component, serving either as the primary inference engine or as the source of training and initialization data.

BGP Manipulation Attacks. Prior work has repeatedly shown that BGP routing information is highly vulnerable to manipulation. Stealthy hijacks occur far more often than previously documented, with thousands of malicious events observed in the wild [41]. Even protocol-compliant BGP announcements can be exploited to trigger persistent routing instability, as demonstrated by attacks such as BGP Vortex [24]. Other studies show that attackers can deliberately evade monitor-based detection by crafting hijacks invisible to public collectors [42]. Beyond hijacking, protocol-level weaknesses enable violations of routing and reachability guarantees [43]. More sophisticated interception techniques, such as SICO [44], further demonstrate how routing behavior manipulation can be achieved using BGP communities alone. Even with BGPsec, inherent vulnerabilities still permit traffic redirection and incorrect packet delivery [33]. Collectively, these works highlight that BGP remains remains fundamentally exposed to a wide range of manipulation attacks.

8 Conclusion

In this paper, we present the first systematic security analysis of ASRank, a foundational AS relationship inference algorithm. We show that ASRank's order-dependent inference is highly vulnerable to targeted manipulation via small perturbations of AS triplets. We present ASROGUE, a practical and scalable attack that injects forged triplets through policy-compliant BGP announcements to precisely control ASRank's inferred relationships. Our evaluation shows that when the attacker controls at least two ASes, ASROGUE can manipulate 96.7% of inferred P2C relationships, remains effective under realistic constraints, and is feasible in the real-world Internet, exposing a critical and previously unexamined weakness in real-world AS relationship inference pipelines.

Acknowledgments

We sincerely thank our Shepherd and all the anonymous reviewers for their valuable comments. This work was supported in part by National Natural Science Foundation of China under Grant 62132011 and 62425201, and Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM114. Qi Li is the corresponding author of the paper.

Ethical Considerations

We responsibly disclosed the identified vulnerabilities, the proposed attack methodology, and corresponding mitigations to CAIDA prior to publication. Across all stages, we took care to avoid any disruption to real networks, adhered to re-

sponsible disclosure practices, and complied with all dataset and PEERING testbed usage policies.

We structure our ethical considerations around three aspects of our methodology: the use of public datasets and simulated experiments, controlled testing on the PEERING testbed, and the broader intent and impact of this work.

Use of Public Datasets and Simulated Experiments. Our analysis of ASRank and the associated attack surface relied primarily on publicly available BGP datasets and offline simulations. All manipulations of AS paths, relationship inferences, and poisoning scenarios were performed entirely offline in a controlled local environment. No modified or forged AS paths were ever announced to the global routing system, ensuring that no real BGP routes, AS relationships, or production networks were affected. All datasets were used strictly in accordance with their terms of use. These simulations served solely to evaluate the robustness of ASRank under adversarial conditions and to highlight a previously overlooked security risk, not to facilitate or encourage malicious exploitation.

Experiments on the PEERING Testbed. For experiments requiring interaction with the real Internet, we used the PEERING testbed through its standard, approved resource-allocation process. We obtained ASNs and prefixes through PEERING's official mechanisms and followed all operational guidelines. Throughout these experiments, no traffic belonging to other ASes was influenced, rerouted, or otherwise affected. To avoid any unintended impact on third-party networks, all forged AS paths were constructed exclusively from ASNs under our control, and the announced prefix carried no production traffic. These experiments were designed to validate feasibility under realistic conditions while ensuring complete isolation from uninvolved networks.

Intent and Broader Impact. This work is intended to raise awareness of a structural vulnerability in ASRank-based AS relationship inference systems and to support the research community in designing more resilient methodologies. All experiments were conducted with caution and isolation, following responsible research practices to minimize risk and prevent harm.

In summary, our study was carried out under strict ethical safeguards: offline simulations for safety, controlled testbed experiments for realism, and responsible disclosure for community benefit. These measures ensured that our findings contribute constructively to improving the security and reliability of AS relationship inference systems without posing risks to the operational Internet.

Open Science

To support open science and ensure reproducibility of our results, we release all ASROGUE research artifacts at https://doi.org/10.5281/zenodo.20397489. The repository is organized to make it easy for researchers to inspect our methodology, reproduce our experiments, and extend our work. Specifically, this repository includes:

- **A Python implementation of ASRank.** This implementation serves as the inference engine used throughout our evaluation. It allows researchers to reproduce our baseline results and to examine how ASRank behaves under adversarial manipulation.
- **Scripts for generating forged AS paths and evaluating the performance of ASROGUE.** These scripts automate the generation of attack triplets and forged AS paths, including both P2C and P2P manipulation, followed by the injection of these forged paths into the ASRank inference pipeline and the measurement of inference deviations induced by adversarial manipulation.
- **All datasets used in our experiments.** We provide the CAIDA AS relationship data in the artifact. Due to Zenodo's storage limitations, only a subset of the RIB snapshots from RouteViews and RIPE RIS is included directly, but we provide stable download links in the README so that researchers can still reproduce our experimental setup and verify the robustness of our findings.
- **Baseline of inference results, AS sizes, and transit degrees.** These files used in our evaluation provide reference outputs for comparison with attacked results, enabling precise quantification of the inference manipulation effects introduced by ASROGUE.

Together, these artifacts provide a complete and transparent reproduction package, allowing the research community to validate our methodology, compare against our baseline, and build upon our work in future studies.

References

[1] G. Huston, "Nanog 95," 2025. [Online]. Available: https://labs.apnic.net/index.php/2025/11/28/nanog-95/

[2] M. Zhang, V. Giotsas, and C. Martinho, "How we detect route leaks and our new cloudflare radar route leak service," The Cloudflare Blog, 2022. [Online]. Available: https://blog.cloudflare.com/route-leak-detection-with-cloudflare-radar/

[3] M. Zhang and C. Martinho, "Cloudflare radar's new bgp origin hijack detection system," The Cloudflare Blog, 2023. [Online]. Available: https://blog.cloudflare.com/bgp-hijack-detection/

[4] Qrator Labs, "Addressing a particular internetworking misconception," 2021. [Online]. Available: https://blog.qrator.net/en/particular-misconception_125/

[5] A. Shah, "There and back again: a look at international routing detours," APNIC Blog, 2017. [Online]. Available: https://blog.apnic.net/2017/01/18/back-look-international-routing-detours/

[6] Internet Society, "Pulse documentation," 2026. [Online]. Available: https://pulse-docs.internetsociety.org/data.html

[7] W. Li, Y. Li, and T. Chung, "{ImpROV}: Measurement and practical mitigation of collateral damage in {RPKI} route origin validation," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 3631–3647.

[8] Y. Chen, Q. Li, K. Xu, Z. Liu, and J. Wu, "Understanding the stealthy bgp hijacking risk in the rov era," in *Network and Distributed System Security (NDSS) Symposium 2026*. Internet Society, 2026.

[9] B. Du, R. Fontugne, C. Testart, A. C. Snoeren, and K. Claffy, "Sublet your subnet: Inferring ip leasing in the wild," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, 2024, pp. 328–336.

[10] Y. E. Anahory, J. Kong, N. Scaglione, J. Furuness, H. Leibowitz, A. Herzberg, B. Wang, and Y. Gilad, "Suppressing {BGP} zombies with route status transparency," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 2025, pp. 1349–1366.

[11] Y. Chen, Q. Yin, Q. Li, Z. Liu, K. Xu, Y. Xu, M. Xu, Z. Liu, and J. Wu, "Learning with semantics: Towards a {Semantics-Aware} routing anomaly detection system," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 5143–5160.

[12] J. Li, J. Cao, Z. Meng, R. Xie, Q. Li, Y. Yang, and M. Xu, "Roll +: Real-time and accurate route leak locating with as triplet features at scale," *IEEE/ACM Transactions on Networking*, 2024.

[13] Georgia Institute of Technology, "Global routing intelligence platform (grip)," 2026. [Online]. Available: https://grip.inetintel.cc.gatech.edu/

[14] Z. Ni, Y. Xu, H. Zou, Y. Li, G. Cheng, and G. Xie, "From address blocks to authorized prefixes: Redesigning {RPKI}{ROV} with a hierarchical hashing scheme for fast and {Memory-Efficient} validation," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, 2025, pp. 1381–1397.

[15] J. Furuness, C. Morris, B. Wang, R. Morillo, A. Herzberg, and A. Kasiliya, "Securing bgp asap: Aspa and other post-rov defenses," in *Network and Distributed System Security (NDSS) Symposium 2025*. Internet Society, 2025.

[16] C. Morris, A. Herzberg, B. Wang, and S. Secondo, "Bgpisec: Improved security of internet routing against post-rov attacks," in *Network and Distributed System Security (NDSS) Symposium 2024*. Internet Society, 2024.

[17] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and K. Claffy, "As relationships, customer cones, and validation," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 243–256.

[18] Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy, and S. Shenker, "Stable and practical {AS} relationship inference with {ProbLink}," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019, pp. 581–598.

[19] Z. Jin, X. Shi, Y. Yang, X. Yin, Z. Wang, and J. Wu, "Toposcope: Recover as relationships from fragmentary observations," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 266–280.

[20] V. Giotsas, M. Luckie, B. Huffaker, and K. Claffy, "Inferring complex as relationships," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, 2014, pp. 23–30.

[21] T. Wu, J. H. Wang, J. Wang, and S. Zhuang, "Routeinfer: Inferring interdomain paths by capturing isp routing behavior diversity and generality," in *International Conference on Passive and Active Network Measurement*. Springer, 2022, pp. 216–244.

[22] CAIDA, "As relationships dataset," 2025. [Online]. Available: https://www.caida.org/catalog/datasets/as-relationships/

[23] L. Prehn, P. Foremski, and O. Gasser, "Kirin: Hitting the internet with distributed bgp announcements," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, 2024, pp. 19–34.

[24] F. Stöger, H. Birge-Lee, G. Giuliari, J. Subira-Nieto, and A. Perrig, "{BGP} vortex: Update message floods can create internet instabilities," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 3613–3629.

[25] CVE-2014-9222. https://nvd.nist.gov/vuln/detail/CVE-2014-9222, December 2014. Accessed 05 February 2026.

[26] CVE-2018-0037. https://nvd.nist.gov/vuln/detail/CVE-2018-0037, July 2018. Accessed 05 February 2026.

[27] CVE-2022-37035. https://nvd.nist.gov/vuln/detail/CVE-2022-37035, February 2022. Accessed 05 February 2026.

[28] X. Feng, C. Fu, Q. Li, K. Sun, and K. Xu, "Off-path tcp exploits of the mixed ipid assignment," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1323–1335.

[29] A. Dhamdhere and C. Dovrolis, "Twelve years in the evolution of the internet ecosystem," *IEEE/ACM Transactions on Networking*, vol. 19, no. 5, pp. 1420–1433, 2011.

[30] PEERING testbed, "Peering: The bgp testbed," 2026. [Online]. Available: https://peering.ee.columbia.edu/

[31] Doug Madory, "Beyond their intended scope: Bgp goof-upx," 2025. [Online]. Available: https://www.kentik.com/blog/beyond-their-intended-scope-bgp-goof-upx/

[32] M. Lepinski and K. Sriram, "BGPsec Protocol Specification," RFC 8205, 2017.

[33] Q. Li, J. Liu, Y.-C. Hu, M. Xu, and J. Wu, "Bgp with bgpsec: Attacks and countermeasures," *IEEE Network*, vol. 33, no. 4, pp. 194–200, 2018.

[34] A. Azimov, E. Bogomazov, R. Bush, K. Patel, J. Snijders, and K. Sriram, "BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects," IETF draft-ietf-sidrops-aspa-verification-24, 2025.

[35] K. Sriram, N. Geng, and A. Herzberg, "Autonomous System Relationship Authorization (ASRA) as an Extension to ASPA for Enhanced AS Path Verification," IETF draft-sriram-sidrops-asra-verification-03, 2025.

[36] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, "Profiling bgp serial hijackers: capturing persistent misbehavior in the global routing table," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 420–434.

[37] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Transactions on networking*, vol. 9, no. 6, pp. 733–745, 2001.

[38] V. Giotsas and S. Zhou, "Inferring as relationships from bgp attributes," 2011.

[39] G. Feng, S. Seshan, and P. Steenkiste, "Unari: an uncertainty-aware approach to as relationships inference," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 272–284.

[40] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Transactions on networking*, vol. 9, no. 6, pp. 681–692, 2001.

[41] P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind your blocks: On the stealthiness of malicious bgp hijacks." in *NDSS*, 2015.

[42] A. Milolidakis, T. Bühler, K. Wang, M. Chiesa, L. Vanbever, and S. Vissicchio, "On the effectiveness of bgp hijackers that evade public route collectors," *IEEE Access*, vol. 11, pp. 31 092–31 124, 2023.

[43] Y. Song, A. Venkataramani, and L. Gao, "Identifying and addressing reachability and policy attacks in "secure" bgp," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2969–2982, 2016.

[44] 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.

[45] CAIDA, "Resource catalog," 2025. [Online]. Available: https://catalog.caida.org/

A ASRank AS Relationship Inference Algorithm

Algorithm 1 outlines the multi-step workflow used by ASRank to infer AS relationships from BGP routing data. The algorithm proceeds as follows.

Step 1. Path sanitization. ASRank removes AS paths containing artifacts such as loops, reserved ASes, and IXPs.

Step 2. Initial ranking. ASRank uses the sanitized AS paths to compute each AS's node and transit degrees and derive an initial rank order.

Step 3. Clique identification. ASRank infers the ASes present at the top of the hierarchy, and links within this clique are inferred as P2P. In practice, this clique is often provided as prior input.

Step 4. Poisoning filtering. ASRank removes AS paths that exhibit signs of BGP path poisoning, such as loops or separated clique ASes, to avoid inferring relationships that do not exist in reality.

Step 5-10. P2C inference. ASRank infers P2C relationships using a set of heuristics applied in a greedy, top-down manner. These heuristics incorporate the ranking (step 5), VP behavior (step 6), atypical degree patterns (step 7), ASes with no providers (step 8), structural properties of stubs and clique ASes (step 9), and unresolved adjacent links (step 10). Newly inferred relationships are used iteratively to resolve additional P2C links. Notably, step 5 alone accounts for over 90% of all P2C relationships inferred in the results.

Step 11. P2P inference. ASRank then classifies any links that remain unresolved as P2P.

Algorithm 1 AS relationship inference algorithm

Require: AS paths, Allocated ASNs, IXP ASes

1: Discard or sanitize paths with artifacts.
2: Sort ASes in decreasing order of computed transit degree, then node degree.
3: Infer clique at top of AS topology.
4: Discard poisoned paths.
5: Infer P2C relationships top-down using the above ranking.
6: Infer P2C relationships from VPs inferred not to be announcing provider routes.
7: Infer P2C relationships for ASes where the provider has a smaller transit degree than the customer.
8: Infer customers for ASes with no providers.
9: Infer P2C relationships between stubs and clique ASes.
10: Infer P2C relationships where adjacent links have no relationship inferred.
11: Infer remaining links as P2P relationships.

B ASRank's Adoption and Influence

Among existing AS relationship inference methods [17–21], CAIDA's ASRank algorithm [17] is a foundational approach that has shaped both subsequent research and operational practice. ASRank underpins CAIDA's monthly releases of inferred AS relationship datasets [22], which are widely regarded as the de facto reference in the community. ASRank's influence extends beyond its direct use. Several prominent inference systems, including the state-of-the-art ProbeLink [18], TopoScope [19], and RouteInfer [21], either build upon ASRank or incorporate its core heuristics as a key component. In parallel, ASRank-inferred datasets are routinely used in operational and industrial settings [1–6], further underscoring their practical relevance. Over the past decade, CAIDA's ASRank-based datasets have supported more than 800 scientific studies across networking, measurement, and security research. As shown in Figure 14, these datasets have been consistently cited by over 20 publications per year, highlighting ASRank's sustained adoption and long-term influence on Internet research.

C Real-World Feasibility of Forged Triplet Propagation

To illustrate the real-world feasibility of forged triplet propagation, we conduct a controlled experiment on the PEERING testbed using the ASNs (AS 47065, AS 61576, AS 61575, and AS 61574) and the IPv4 prefix (184.164.252.0/24) allocated to us. Our goal is to validate whether a crafted AS triplet can propagate across the Internet when announced through standard, policy-compliant BGP mechanisms.

Experimental setup. We construct the forged triplet "61576

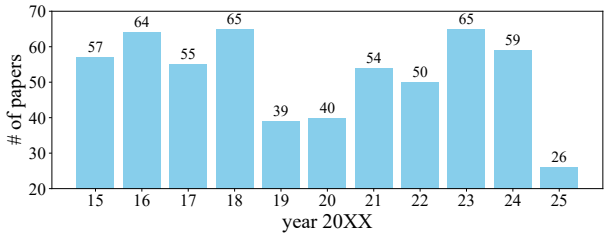


Figure 14: Number of research papers using CAIDA AS business relationship datasets over the past decade, showing consistently more than 20 publications per year [45].

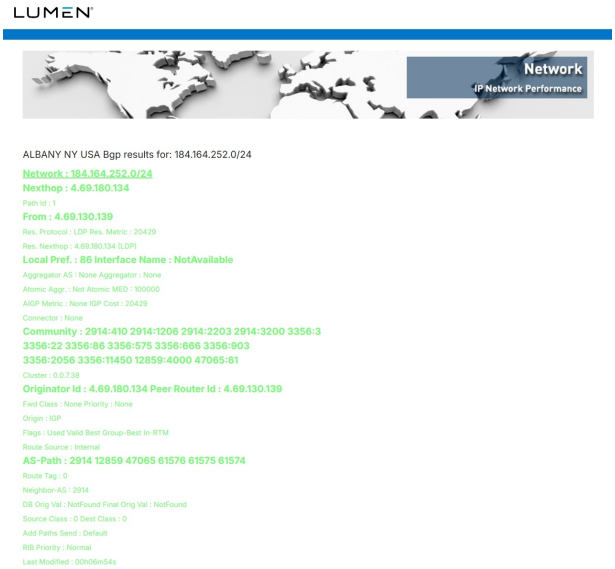


Figure 15: Routes to 184.164.252.0/24 observed in Albany (North America).

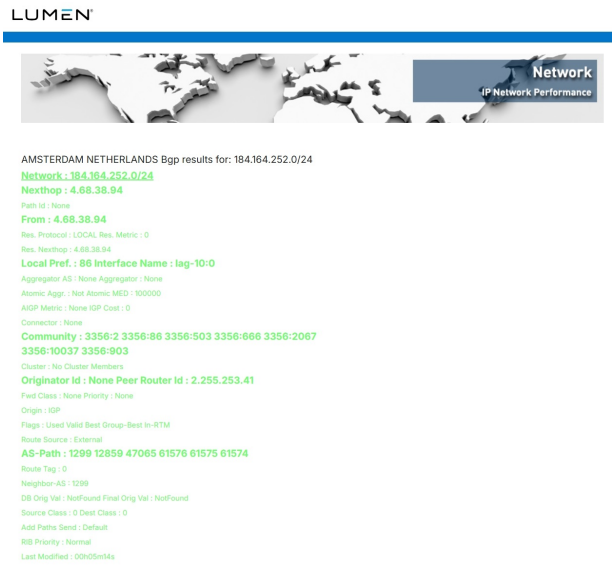


Figure 16: Routes to 184.164.252.0/24 observed in Amsterdam (Europe).

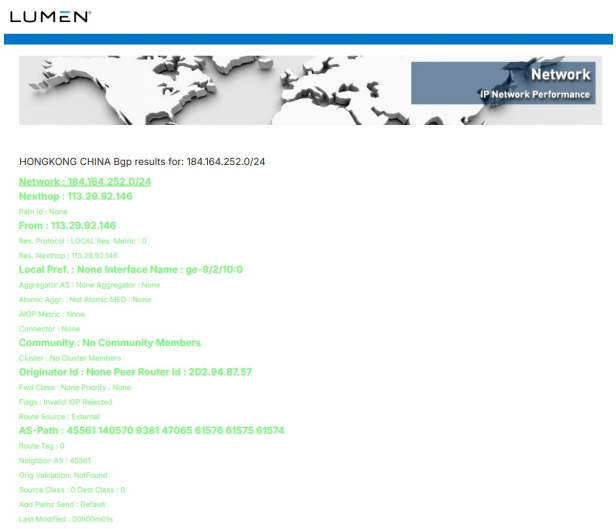


Figure 17: Routes to 184.164.252.0/24 observed in Hong Kong (Asia).

61575 61574", where AS 61574 serves as the ROA-authorized origin for prefix 184.164.252.0/24. AS 47065 is configured as the announcing AS. To embed the forged triplet into the AS path, we apply standard AS path prepending at AS 47065, resulting in an announced path of the form "47065 61576 61575 61574". This announcement is loop-free, syntactically valid, and consistent with common filtering practices, thereby reflecting the constraints faced by a real attacker operating within today's routing ecosystem. To ensure ethical compliance and prevent any unintended impact on third-party networks, all forged paths in this experiment were constructed solely from ASes under our control, and the announced prefix carried no production traffic.

Propagation observation. After issuing the announcement, we monitor its propagation using *Lumen Looking Glass servers*, which provide vantage points across multiple continents. The forged triplet announcement becomes visible in North America 15, Europe 16, and Asia 17 within minutes. These observations confirm that the crafted triplet is accepted and propagated by real Internet routers without triggering filtering or rejection.

Summary. This experiment demonstrates that a forged AS triplet, constructed solely from controlled ASNs and embedded via standard AS path prepending, can successfully propagate across the global Internet. The results reinforce that the core primitive of ASRogue is not only theoretically feasible but also operationally viable under real-world routing conditions.