FEDERAL UNIVERSITY OF RIO GRANDE - FURG
CENTER FOR COMPUTATIONAL SCIENCE
GRADUATE PROGRAM IN COMPUTING
MASTER'S COURSE IN COMPUTER ENGINEERING

Master's Dissertation

Investigating the Security Implications of Traffic Engineering and Connectivity on Internet Routing

Renan Paredes Barreto

Master's Dissertation presented to Graduate Program in Computing of Federal University of Rio Grande - FURG, as partial requirement to obtain the degree of Master in Computer Engineering

Advisor: Prof. Dr. Pedro de Botelho Marcos
Co-advisor: Prof. Dr. Leandro Márcio Bertholdo

Rio Grande, 2025

B273i Barreto, Renan Paredes
Investigating the security implications of traffic engineering and connectivity on internet routing / Renan Paredes Barreto. – 2025.
98 f.

Dissertação (Mestrado) – Universidade Federal do Rio Grande – Programa de Pós-Graduação em Computação, 2025.

Orientador: Dr. Pedro de Botelho Marcos.
Coorientador: Dr. Leandro Márcio Bertholdo.

1. Computação. 2. Engenharia de tráfego. 3. Protocolos de comunicação. 4. Segurança de redes. I. Marcos, Pedro de Botelho. II. Bertholdo, Leandro Márcio. III. Título.

CDU 004

Catalogação na fonte: Bibliotecário Clériston Ribeiro Ramos CRB10/1889

Universidade Federal do Rio Grande
Centro de Ciências Computacionais
Programa de Pós-Graduação em Computação
Curso de Mestrado em Engenharia de Computação

DISSERTAÇÃO DE MESTRADO

Investigating the Security Implications of Traffic Engineering and Connectivity on Internet Routing

Renan Paredes Barreto

Banca examinadora:

gov.br
Documento assinado digitalmente
BRUNO LOPES DALMAZO
Data: 30/09/2025 16:37:16-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Bruno Lopes Dalmazo

gov.br
Documento assinado digitalmente
ANTONIO MARINHO PILLA BARCELLOS
Data: 01/10/2025 19:41:07-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Marinho Pilla Barcellos

gov.br
Documento assinado digitalmente
PEDRO DE BOTELHO MARCOS
Data: 30/09/2025 16:09:21-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Pedro de Botelho Marcos
Orientador

This work is dedicated to my family, who have offered me examples to follow, unconditional support, and the opportunity to pursue my dreams.

ACKNOWLEDGEMENTS

I would like to thank Prof. Dr. Pedro de Botelho Marcos for all the support, the time he dedicated, and the opportunities he gave to me. I would also like to thank Prof. Dr. Leandro Marcio Bertholdo for all the knowledge you shared. Thank you both for believing in my work, for advising me through it, and for your patience.

Sidney Delgado Barreto, Vera Maria Paredes Barreto, Christian Paredes Barreto, and William Paredes Barreto, thank you for the privilege of having you as my examples. Everything I have achieved was because of your support. I love you. I hope to make you proud and pass on what you taught me.

Juliana de Aguiar Alves, thank you for being my companion, despite my absence due to work or my degree. Thank you for being my safe haven and bringing me happiness every day. Thank you for everything. I love you.

”Be diligent in these matters;
give yourself wholly to them,
so that everyone may see your progress.
— HOLY BIBLE, TIMOTHY 4:15

ABSTRACT

BARRETO, Renan Paredes. **Investigating the Implications of Traffic Engineering and Connectivity on Internet Routing**. 2025. 98 f. Master's Dissertation – Graduate Program in Computing. Federal University of Rio Grande - FURG, Rio Grande.

The reliability and security of Internet routing are increasingly challenged by applications with strict service requirements, where connectivity and traffic engineering play a central role. While operators use traffic engineering to optimize performance and resilience, such decisions can inadvertently amplify routing security risks. Existing mechanisms such as BGPSec, RPKI, and ASPA remain insufficient due to limited deployment and technical vulnerabilities, leaving open questions on how traffic engineering practices themselves affect routing security.

This work addresses the gap in understanding how traffic engineering and connectivity influence Internet routing security. We propose a methodology that combines measurements from both the control and data planes. Using the PEERING platform to generate BGP announcements, we analyze the effects of AS Path Prepend, prefix length, and selective route announcements.

Our results show that while prepending can increase the impact of a hijack, from 17% to 67% in one scenario, connectivity alone may also expose an AS to prefix hijacks. We further demonstrate that hijacking via longer prefixes is particularly effective in achieving 100% of the data plane and control plane targets. Based on these findings, we provide a comprehensive view of the current state of the announced address space, showing that 93.3% could be victims of hijack with a longer prefix, and discuss practical implications for routing security.

Keywords: BGP, Traffic Engineering, Security, Routing.

RESUMO

BARRETO, Renan Paredes. **Investigating the Security Implications of Traffic Engineering and Connectivity on Internet Routing**. 2025. 98 f. Dissertação (Mestrado) – Programa de Pós-Graduaçõo em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

A confiabilidade e a segurança do roteamento da Internet são cada vez mais desafiadas por aplicações com requisitos rigorosos de serviço, nas quais a conectividade e a engenharia de tráfego desempenham um papel central. Embora os operadores utilizem a engenharia de tráfego para otimizar desempenho e resiliência, tais decisões podem, inadvertidamente, amplificar os riscos de segurança no roteamento. Mecanismos existentes como BGPSec, RPKI e ASPA permanecem insuficientes devido à adoção limitada e a vulnerabilidades técnicas, deixando em aberto questões sobre como as próprias práticas de engenharia de tráfego afetam a segurança do roteamento.

Este trabalho busca suprir a lacuna no entendimento de como a engenharia de tráfego e a conectividade impactam a segurança do roteamento da Internet. Propomos uma metodologia experimental que combina medições nos planos de controle e de dados. Utilizando a plataforma PEERING para gerar anúncios BGP, analisamos os efeitos de *AS Path Prepend*, prefixos mais específicos e anúncios seletivos.

Nossos resultados mostram que o uso de *prepend* pode aumentar o impacto de um sequestro de prefixo, de 17% para 67% em um cenário; entretanto, a conectividade do AS por si só também pode torná-lo vulnerável a sequestros de prefixos. Demonstramos ainda que sequestros utilizando prefixos mais específicos são particularmente eficazes, sequestrando 100% dos alvos no plano de controle e plano de dados. Com base nesses achados, apresentamos um panorama do estado atual do espaço de endereçamento anunciado, demonstrando que 93.3% podem ser vítimas de sequestro por um prefixo mais específico, e discutimos implicações práticas para a segurança do roteamento.

Palavras-chave: BGP, Engenharia de Tráfego, Segurança, Roteamento.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AS	*Autonomous System*
ASN	*Autonomous System Number*
ASPP	*Autonomous System Path Prepend*
BGP	*Border Gateway Protocol*
CDN	*Content Delivery Network*
ICMP	*Internet Control Message Protocol*
IP	*Internet Protocol*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
IRR	*Internet Routing Registry*
IXP	*Internet eXchange Point*
MRT	*Multi-Threaded Routing Toolkit*
RFC	*Request For Comments*
RIB	*Routing Information Base*
RIR	*Regional Internet Registry*
ROA	*Route Origin Authorization*
RPKI	*Resource Public Key Infrastructure*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
VP	*Vantage Point*

TABLE OF CONTENTS

1 INTRODUCTION

Traffic delivery is a fundamental component of current Internet operations. As the Internet evolves in both scale and importance, its underlying infrastructure faces increasing complexity and growing traffic volumes. This growth is driven by the need to meet application requirements, enhance users' quality of experience, and ensure network resilience.

The Internet emerges from the interconnection of multiple networks, known as Autonomous Systems (ASes). Each AS retains autonomy over its routing decisions when transmitting information. ASes expand their connectivity through Internet Exchange Points (IXPs) and transit ASes. IXPs are physical locations where network switches enable different networks to exchange traffic directly. Transit ASes, in turn, provide global Internet connectivity to other ASes, typically operating within a specific region. These transit ASes may adopt strategies that prioritize their own interests, often of an economic nature.

To coordinate the relationships and objectives of each AS, the Internet relies on the Border Gateway Protocol (BGP). BGP propagates routes across ASes and, when multiple paths exist to the same destination, applies a series of selection criteria. For outbound traffic, operators can influence routing through the *local preference* attribute, which defines preferred paths for traffic leaving their networks. For inbound traffic, however, BGP does not provide a direct parameter to enforce preferences. Network operators, therefore, rely on traffic engineering techniques to influence the choices made by neighboring ASes. These techniques include manipulating route advertisements through prefix length, selective announcements, prepending, and BGP communities.

Operators apply traffic engineering to make certain routes more or less attractive. While this practice helps them achieve their goals, it can exacerbate security risks regarding prefix hijacking events. In a hijack event, an AS announces a prefix it does not legitimately own, diverting traffic. Such incidents can disrupt services or serve as vectors for more severe attacks. For example, during a hijack of KLAYSwap [37], users were redirected to download malicious software, resulting in the theft of cryptocurrency assets.

Prefix hijacks can take multiple forms: forged-origin attacks, in which an adversary announces a prefix without legitimate ownership; AS path manipulation, where the at-

tacker inserts its AS number into an otherwise valid path; and configuration errors, which correspond to unintentional announcements caused by operator mistakes. Such events occur frequently—around 17.5 suspected forged-origin cases are observed per day [21]. Moreover, approximately 1.4% of ASes have been classified as serial hijackers through machine learning techniques [38]. Existing defense mechanisms, such as the Resource Public Key Infrastructure (RPKI) [7] and the Autonomous System Provider Authorization (ASPA) [1], provide only partial protection. RPKI mitigates a limited set of hijack types and still lacks universal adoption, while ASPA remains under development.

A systematic understanding of the factors that amplify the likelihood or severity of hijacks is still missing. In this dissertation, we seek to advance the understanding of how traffic engineering impacts prefix hijacks. Specifically, we investigate how connectivity and traffic engineering practices influence the success and impact of route origin hijacks.. To guide this investigation, we formulate the following research questions:

RQ1. How do different traffic engineering practices affect the impact of a prefix hijack?

RQ2. Which characteristics of the victim influence the outcome of a hijack?

RQ3. Which characteristics of the attacker influence the outcome of a hijack?

RQ4. What leads an AS to accept a hijack announcement?

RQ5. Based on these results, what is the current state of traffic engineering employment on the Internet, and its possible impact on security?

To answer these questions, we use the PEERING platform [35], which enables controlled BGP experiments on the global Internet. By instantiating ASes and generating real announcements, we emulate scenarios that include the following traffic engineering practices: prepending, selective announcements, and more specific prefixes. We analyze their effects using both control-plane data and data-plane measurements, providing a systematic evaluation of how these techniques influence prefix hijack events.

While previous work has specified a method to analyze the effect of prepending on routing [12], this dissertation expands that study by investigating a broader range of traffic engineering practices and connectivity factors that influence routing security. To this end, we conduct experiments on the PEERING testbed, analyzing scenarios in which ASes differ in their level of connectivity and apply techniques such as prepending and prefix deaggregation. Our methodology combines control-plane and data-plane measurements, enabling us to evaluate how these factors affect the propagation and resilience of BGP announcements, both from the perspective of a victim and from that of a potential attacker.

The contributions of this dissertation are twofold. First, we provide empirical insights into how connectivity and traffic engineering affect the likelihood and impact of hijacks. Second, we present an updated view of the Internet's current exposure to such threats.

These results offer practical guidance for operators, who must weigh the operational benefits of traffic engineering against its security risks.

The results presented in this dissertation show that using a single AS path prepend increases the likelihood of a hijack to 17%; when additional prepends are applied, this value can rise to 67%. We also observed that hijacks with more specific prefixes are particularly damaging, in some cases capturing 100% of the control-plane targets. Furthermore, our analysis indicates that 93.3% of the announced Internet address space could potentially be compromised by an attack based on more specific prefix hijacking.

This dissertation is organized as follows. Chapter 2 introduces the concepts related to the Internet, ASes, and BGP. Chapter 3 reviews the main studies on traffic engineering and prefix hijacking. Chapter 4 details the methodology of our experiments, while Chapter 6 presents the results. Finally, Chapter 10 discusses the conclusions and final remarks.

2 BACKGROUND

In this chapter, we define the concepts used in this work. Section 2.1 introduces Autonomous Systems (ASes) and the Border Gateway Protocol (BGP), discussing how ASes are organized and how BGP enables their communication. Section 2.2 presents BGP route selection criteria and traffic engineering techniques. Section 2.3 addresses security challenges in routing and BGP. Section 2.4 discusses mechanisms for mitigating security gaps and validating routes. Section 2.5 focuses on BGP collectors and databases, describing their formats and limitations. Finally, Section 2.6 presents the PEERING research platform, explaining its objectives and operational principles.

2.1 Autonomous Systems and BGP

An *Autonomous System* (AS) is a collection of routers under a single administrative domain, operated by one or more entities, and governed by a uniform routing policy [24]. Each AS is identified by an *Autonomous System Number* (ASN), which is assigned by a *Regional Internet Registry* (RIR). RIRs are non-profit organizations responsible for distributing ASNs and IP address blocks within their respective regions. These address blocks, known as *prefixes*, are delegated by the *Internet Assigned Numbers Authority* (IANA) to the RIRs.

ASes can be classified according to their role in the Internet ecosystem. *Content ASes*, such as Google, Amazon, and Meta, host and distribute digital content. *Enterprise ASes* serve organizations' internal connectivity needs without providing public Internet access. *Access ASes*, or eyeball networks, connect end users to the Internet, usually through Internet Service Providers (ISPs) such as Vivo and Telefônica in Brazil. *Transit ASes* provide connectivity to other networks; Tier-1 ASes are a special case of transit providers that reach the entire Internet solely through peering agreements, without purchasing transit. These different roles are central to understanding the Internet's connectivity, resilience, and economic structure.

Interconnection between ASes takes place through two main arrangements: *peering agreements*, in which ASes exchange traffic without monetary compensation, and *transit*

agreements, in which one AS pays another for connectivity to the global Internet. While peering usually occurs between networks of similar size to improve performance and reduce costs, transit agreements are essential for smaller ASes that rely on larger providers for global reachability. Since transit is costly, many ASes use Internet Exchange Points (IXPs) to optimize traffic exchange. IXPs are physical infrastructures that interconnect multiple networks, reducing reliance on expensive transit providers, lowering latency, and improving resilience. Content Delivery Networks (CDNs), for example, frequently collocate infrastructure at IXPs to bring content closer to users.

The *Border Gateway Protocol* (BGP) is the standard inter-domain routing protocol of the Internet. It allows ASes to exchange routing information and decide which routes to use for forwarding traffic [32]. Each AS uses BGP to select routes to IP prefixes based on available announcements from its neighbors. For outbound traffic, operators influence the chosen path using attributes such as *local preference*. For inbound traffic, BGP does not provide a direct mechanism to define preferred paths, so operators rely on traffic engineering techniques such as prepending, selective announcements, and prefix length manipulation.

2.2 Route Selection and Traffic Engineering

BGP defines a sequence of decision criteria to select the best route to a given prefix [33]. While some steps are internal to protocol operations, several can be exploited for traffic engineering. Route selection follows, among others, the following criteria:

1. Longest prefix match (an IP-level criterion, not BGP-specific);
2. Highest *local preference*;
3. Shortest AS path;
4. Lowest origin type;
5. Lowest Multi-Exit Discriminator (MED);
6. Prefer eBGP over iBGP routes;
7. Lowest IGP metric to the next hop.

Figure 1 illustrates these key selection steps.

2.2.1 Prefix Length

The specificity of an IP prefix is the first criterion in route selection. Although not defined by BGP itself, but by IP addressing rules, routes with longer prefix matches are

Figure 1: Route selection criteria relevant to traffic engineering.

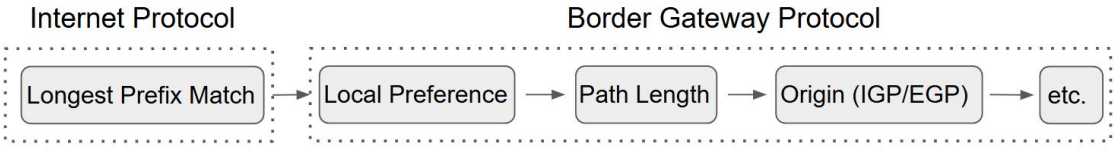
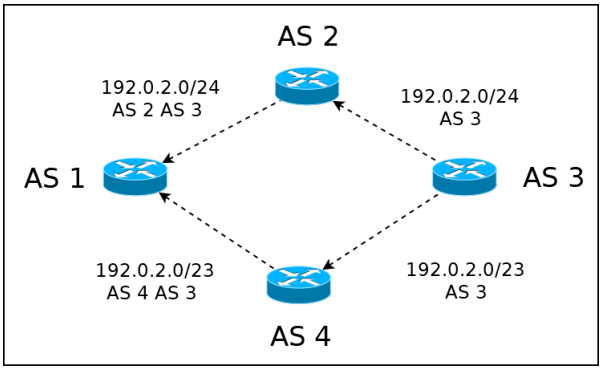


Figure 2: Example where AS 3 advertises a more specific prefix to AS 2, which AS 1 then prefers.



always preferred. For instance, an announcement of 192.0.2.0/24 takes precedence over 192.0.2.0/23, as shown in Figure 2.

While prefix deaggregation enables more precise traffic engineering, it increases the number of entries in global routing tables. Since prefixes longer than /24 are typically filtered, announcing more specific routes than /24 is generally ineffective.

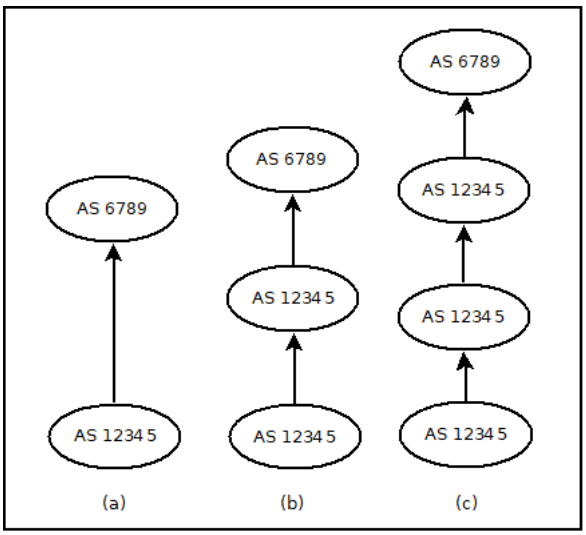
2.2.2 Local Preference

Local preference is an attribute set by an AS to choose among multiple outbound routes. Given two announcements of equal prefix length, the one with the higher local preference is selected. This mechanism allows operators to control how traffic exits their network, but does not directly affect how inbound traffic is received.

2.2.3 AS Path Length and Prepending

When prefix specificity and local preference values are equal, BGP prefers the route with the shortest AS path. Operators can exploit this behavior through *AS path prepending*, a technique in which an AS artificially lengthens its own path by repeating its ASN in the announcement. This makes the route less attractive to neighbors, as illustrated in Figure 3.

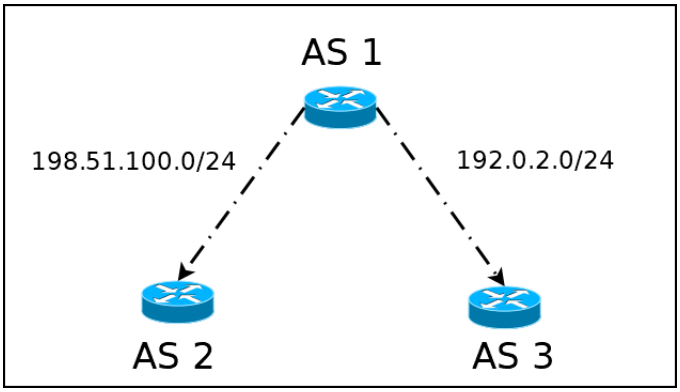
Figure 3: Example of path prepending: (a) no prepending, (b) prepend of one ASN repetition, (c) prepend of two repetitions.



2.2.4 Selective Announcements

Selective announcement refers to advertising a prefix only to a subset of neighbors, as shown in Figure 4. While this technique can control inbound traffic, it reduces the number of available paths to the announcing AS, potentially increasing vulnerability to hijacks.

Figure 4: Example of selective announcements, where AS 1 advertises different prefixes to different neighbors.

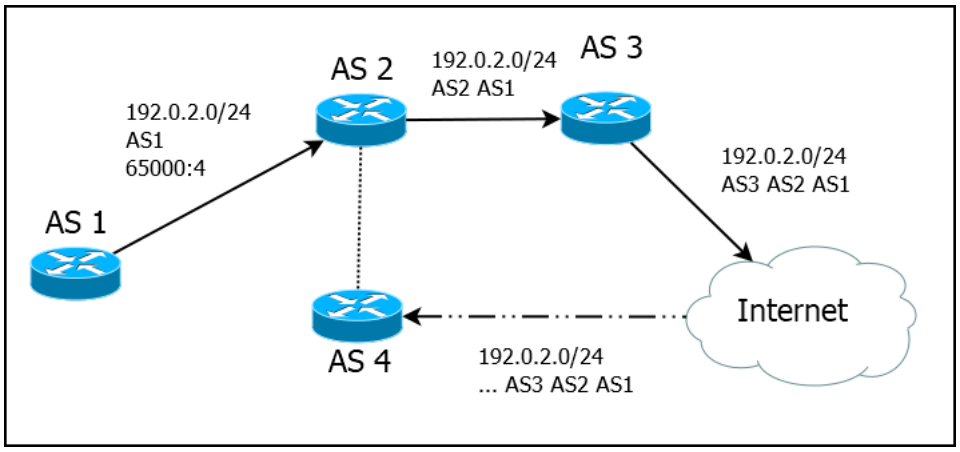


2.2.5 BGP Communities

BGP communities allow operators to tag announcements with instructions for downstream ASes. For example, a community may request that a route not be propagated to certain neighbors, or that prepending be applied automatically. Figure 5 shows an example of a community restricting propagation. Since community values are not standardized,

misconfigurations may occur.

Figure 5: Example where AS 1 attaches a community to its announcement so that AS 2 does not propagate it to AS 4.



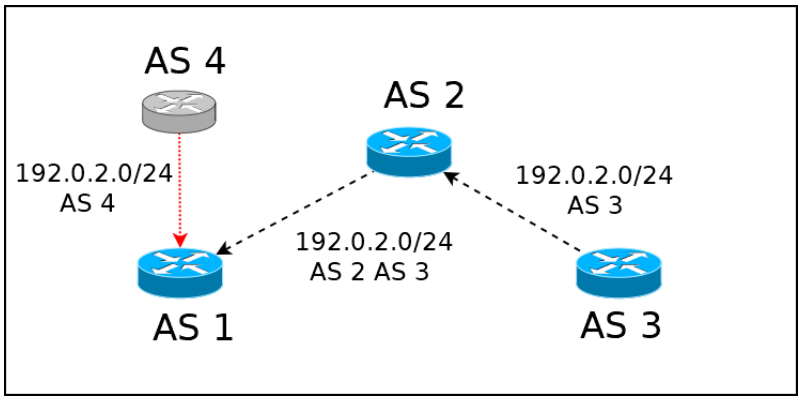
2.3 BGP Security

The security of the BGP protocol remains an unresolved challenge, despite attempts to mitigate its vulnerabilities. Among the problems faced are (i) denial of service, aimed at preventing services and applications from operating at full capacity; (ii) misconfigured announcements, for example, invalid routes can be announced; (iii) route leaks, where an AS announces a route it should not, causing traffic to flow through an undesired path. There is also a lack of route authentication and ownership verification of prefixes. Finally, during a prefix hijacking event, BGP does not verify whether a route is legitimate, allowing an AS to announce a prefix it does not own as the origin. This event is the focus of this work. These cases can be described as prefix hijacking events, in which not all are malicious, and some may be configuration errors.

BGP hijacking occurs when an AS announces a prefix that does not belong to it, or forges a path to the victim, diverting connections that should go to the correct destination to this AS, provided it is considered the route to be taken according to BGP route selection criteria, as shown in Figure 6. This can be used to waste requests that should go to the correct AS, or to intercept this data for monitoring or alteration, and then forward it to the correct destination [2].

The relationships between connected ASes may lead to the need to use traffic engineering tools as more routes become available to optimize the use of connections and their costs. However, the use of traffic engineering techniques can influence a prefix hijacking event. For example, the use of less specific prefixes than /24 can create vulnerabilities when it comes to prefix hijacking, as an AS announcing a /23 prefix may have its prefix

Figure 6: Example where AS 4 announces to AS 1 a prefix that belongs to AS 3.



hijacked by another AS announcing a more specific /24 prefix. Therefore, it is necessary to weigh the benefits of prefix aggregation with its vulnerabilities and the drawbacks of prefix disaggregation against the limitations of routing tables and filtering by ASes.

The use of *prepends* can create a vulnerability to intentional or accidental prefix hijacking events [12]. By announcing a path without *prepends*, an AS can hijack another AS prefix, provided the other criteria are equal, due to having a shorter path compared to the original owner of the prefix announcement.

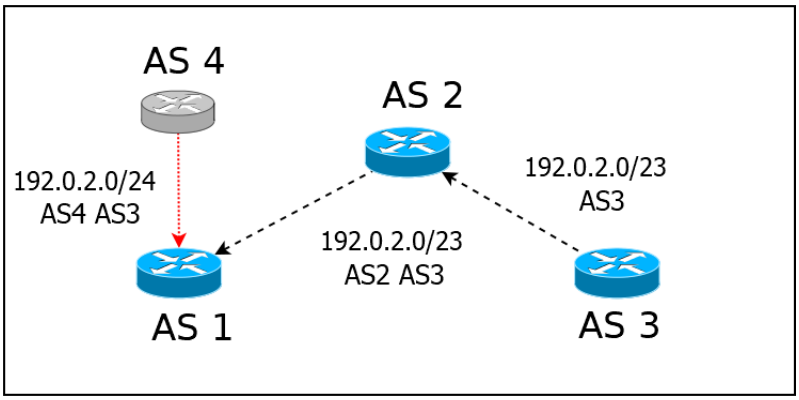
In the case of ASes with few neighbors to which they are connected, there may not be a need to optimize the use of their routes. However, a smaller number of neighbors—and consequently fewer available routes to the originating AS of an announcement—can make it easier to hijack its prefixes. For instance, if an AS has only one neighbor, a prefix hijacking event only needs to convince this neighbor to follow the false route to intercept all connections to the announced prefix. Conversely, an AS with multiple neighbors has a lower probability of having all its prefix routes hijacked, as the actor initiating the hijacking event would need to convince several ASes, using BGP criteria, to adopt their route.

2.4 BGP Security Issues Mitigation

By discussing various possibilities of traffic engineering techniques and the characteristics of the BGP protocol, it is evident that verifying the authenticity of announcements and addressing prefix hijacking events are crucial points, despite the protocol lacking built-in mechanisms for this. ROA (*Route Origin Authorization*) aims to address this issue by linking IP blocks to the ASes authorized to announce them. The creation of such records is managed by specific entities, such as RIRs or LIRs (*Local Internet Registries*). RPKI keys are employed to validate these ROA objects, ensuring that the announcement originates from an AS certified to make it [7].

One issue with the use of ROA and RPKI is the creation of objects with a broader max length attribute considering prefix lengths not currently being announced. For example, allow a /24 prefix even though the network only announces a /23. This practice simplifies the management of announcements by network operators, as it avoids the need to update ROA objects if changes in prefix aggregation or disaggregation are required. However, this same practice can pave the way for a malicious AS to announce a prefix contained within a valid ROA, falsely claiming to be a neighbor of the origin AS for that prefix.

Figure 7: Example of a hijack bypassing RPKI origin validation. In this scenario, the hijacker adds the victim ASN in the AS path even though the hijacker does not have a connection to the victim.



There is also the challenge of adopting RPKI, as not all ASes may adopt route verification, with at least 27% of ASes enforcing ROV, Route Origin Validation, strictly or partially [20]. This limitation extends to other implementations aimed at improving BGP protocol security.

ASPA seeks to enhance the foundation laid by RPKI by authenticating the entire path of a route, thereby making it more effective, but its adoption remains limited [40, 1]. BGPsec aims to address this by enabling update messages to be verified, ensuring that each AS in the path of the message has authorized the announcement of that route to the next AS in the path [25]. However, this approach also faces economic barriers, or lacks incentive, and, as a result, is not yet operational [30].

2.5 Data Collection

To verify the use of traffic engineering techniques, BGP collectors serve as one of the main sources of information. These collectors gather data about the state of the control plane based on the visibility of the collector, these collectors can be composed of multiple monitors that share this information. There are several collector projects, such as Route-Views [41] and RIPE RIS [28], which have collectors in various locations with different

levels of visibility. Consequently, it is likely that a single collector cannot access all the routes available on the Internet.

This limitation complicates the use of BGP data to accurately represent the network state. As a result, employing multiple collectors is a viable option; however, this approach may introduce data duplication and increase the volume of information. Another issue is that since BGP collectors provide the community with a snapshot of routes and relationships between ASes, those are published in specific periods of time, which can lead to difficulty synchronizing with experiments and other sources of information.

RIS Live is a BGP data service that provides real-time updates from network monitors [29] from RIPE. This format solves the synchronization issue but requires a connection to the API to receive data. In this case, the information is published in JSON format, including announcements, withdrawals, paths, communities, and more. RIS Live has approximately 400 monitors, which are used as the source of control plane data for the tool.

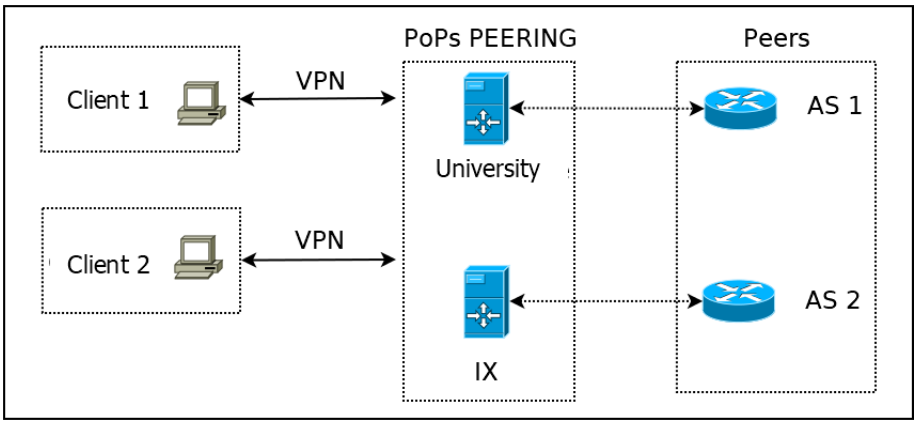
To increase data coverage for the experiments and mitigate the challenges of BGP data collection, we use measurements from the data plane. We use *pings*, more specifically *ICMP-echo-requests*, which were generated with the *nping* tool. By employing an IPv4 *hitlist* based on the ANT IP list [13], we perform *pings* to approximately 40,000 ASes and observe the response behavior during the experiments to define which targets were impacted by the hijack. To capture the responses, we used the tcpdump [19] tool.

2.6 PEERING Testbed

The study of BGP announcement behavior can be conducted through topology simulations. However, this approach struggles with the challenge of modeling agreements between ASes, meaning simulated experiments might not yield results consistent with those observed when applied to the Internet. PEERING emerged as a tool to enable active routing research within the Internet topology. This tool provides the capability to instantiate ASes and generate BGP announcements to PEERING's *peers* using prefixes reserved for the platform.

The platform allows the use of BGP communities, traffic engineering techniques, and the parallelism of experiments, as well as the instantiation of multiple virtual ASes. Client connections to PoPs (Points of Presence) are established through VPNs, which are located in different places such as universities, Internet Exchange Points, and others. The BGP multiplexers, or mux, of PEERING are the points of presence from where clients can connect and originate BGP announcements, such as amsterdam01.

Figure 8: PEERING architecture: ASes peering to point-of-presence servers.



3 RELATED WORK

In this chapter, we will discuss relevant work in the context of prefix hijacking events and traffic engineering. These focus on vulnerabilities in the Internet that allow or facilitate routing attacks or address challenges related to inbound traffic engineering [1].

This chapter is organized as follows. In Section 3.1, we will present work that addresses challenges in traffic engineering. Finally, in Section 3.2, we will focus on work dealing with prefix hijacking events or their mitigation. In Section 3.3, we discuss the current scenario given the work found following the steps previously described.

3.1 Traffic Engineering

Traffic engineering techniques use the manipulation of BGP announcements according to route selection criteria, primarily to influence inbound traffic. While the use of ITE aims to accomplish ASes' objectives according to their policies, it can lead to security issues. As such, we must analyze previous studies to determine whether: (A) there are guidelines and prior research on ITE; and (B) the application of ITE is still occurring.

Research has already examined traffic engineering techniques, their impact on traffic, as well as proposals for objectives and steps in the traffic engineering process [15]. Models for applying routing policies have also been studied and proposed [16].

We can also observe solutions for routing challenges focused on specific operations, such as Azure, where approaches have been developed for route announcements with the aim of using ingress traffic engineering to improve performance and reduce latency of network services [23]. Although point (A) is confirmed, and there are indeed guidelines for ITE, we must still determine whether ITE is employed and whether its use follows these guidelines.

However, the continuous evolution of the Internet, as exemplified by the installation of new infrastructure [14], can lead to new challenges and require changes in ITE usage.

[1] Literature review in two stages: (i) databases IEEE (339), ACM (81), Springer (150), CAIDA (12), filtering post-2020, yielding 15 papers; (ii) conferences SIGCOMM (293), CoNEXT (149), PAM (130), IMC (254), NSDI (456), last 5 years, yielding 33 papers. After deduplication and relevance check, 16 were included.

Not only do topological changes require a reevaluation of previous studies, but they also demand a review of how ASes employ ITE [22], which shows that ITE is still in use. While point (B) is also true, the same study highlights that operators may not follow the proposed and published models and guidelines, a fact also observed by other authors [18].

Thus, it is possible to conclude that traffic engineering techniques are indeed being applied, as well as efforts to make their use more efficient, such as employing them according to defined standards. These challenges may increase as the Internet topology becomes more complex. However, during the use of traffic engineering, the security of BGP announcements in relation to prefix hijacking events may deteriorate.

3.2 Prefix Hijack Events and Mitigation

The literature on traffic engineering techniques reveals important security gaps, particularly concerning their application. For example, the prepend technique was first examined in a 2005 study, which analyzed its use and proposed strategies for effective handling [9]. More recently, in 2020, a study investigated the security implications of prepending [12], showing that longer prepends significantly increase vulnerability to prefix hijacking, with up to 94% of monitored traffic hijacked when a prepend of length three was employed.

To mitigate prefix hijacking events, several proposals have been made, some of which have been partially implemented. One such mechanism is DROP (Don't Route Or Peer), a list of prefixes deemed potentially harmful to the community. Studies have also observed the effectiveness of filtering techniques for hijack prevention [31]. However, the DROP list is limited, as it contains only a small subset of malicious prefixes. Additional mechanisms, such as Internet Routing Registry (IRR) records and RPKI, can complement these defenses. Both, however, present weaknesses: fraudulent entries are sometimes inserted into IRRs, allowing malicious prefixes to appear legitimate, while RPKI faces the risk of signed but unannounced prefixes being misused by attackers. For instance, one study reported a case in which a Russian AS announced a route to a Peruvian AS's signed prefix, as if the two networks were directly connected.

The adoption of RPKI has been growing within the AS community, and with fewer configuration errors, invalid announcements — including malicious ones — can be filtered out. However, challenges remain, such as the use of maximum signature sizes, lack of key updates, and false announcements of the type described previously [11, 39]. Another possible vulnerability arises when the certification authority for RPKI keys and ROA objects itself acts maliciously, since it could manipulate records to carry out attacks [42].

Despite increasing RPKI/ROA adoption, many ASes still do not use these mechanisms, even though they are aware of prefix hijacking events. Reported barriers include costs and technical difficulties [36]. These limitations allow hijacking to persist, and some

ASes have even been classified as serial hijackers due to their repeated malicious behavior [38]. For instance, in cases where hijackers simulated paths to the legitimate prefix owner, 17.5 suspicious cases per day were detected [21].

Research has shown that up to 20% of routes for new AS pairs may be forged [10]. This analysis is often based on BGP collector data, but attackers may deliberately manipulate announcements to avoid detection by well-known collectors and monitors [27]. Consequently, hijacking events not only become harder to detect but also complicate the broader analysis of BGP data.

3.3 Discussion

Thus, it has been observed in Section 3.1 that the challenges of routing continue to be present, with new routes due to infrastructure or new connections between ASes. This leads to the use of traffic engineering techniques to achieve the goals of the network, such as performance or costs. As such, not only guidelines have been proposed for ITE, but also solutions to automate ITE, aiming to decrease latency and increase resilience.

However, at the same time, the risks associated with the use of these techniques are not fully understood, for example, the risk of a hijack with a longer prefix. Leading to a scenario where it can cause issues instead of solving problems, ”the dose makes the poison”. Even though there are solutions being deployed, they do not resolve the issue completely. As seen in Section 3.2, RPKI adoption, despite growing, still does not cover all the routed address space.

To complete the situation, hijacks do occur [37, 21], inflicting damage on ASes, applications, and users. Therefore, it is necessary to expand the knowledge regarding the security of BGP announcements, given the use of traffic engineering techniques and AS connectivity, to provide network operators with the necessary information to achieve their routing objectives efficiently and securely.

4 METHODOLOGY

Considering the existing knowledge gap regarding the security of BGP announcements (Section 3.2) and the use of traffic engineering techniques (Section 3.1), we define a methodology to conduct experiments. This methodology assesses the potential impact of using ITE on BGP security (Section 2.3).

A prefix hijack event can be described in three stages: (1) the state of the network before the attack; (2) the state of the network during the attack; and (3) the state of the network during mitigation or post-attack. Any methodology designed to evaluate the impact of a hijack must reproduce at least the first two stages. To do so, it is necessary to announce the victim prefix and wait for its propagation (stage 1), followed by the propagation of the attacker's announcements (stage 2) across the Internet.

4.1 Announcements

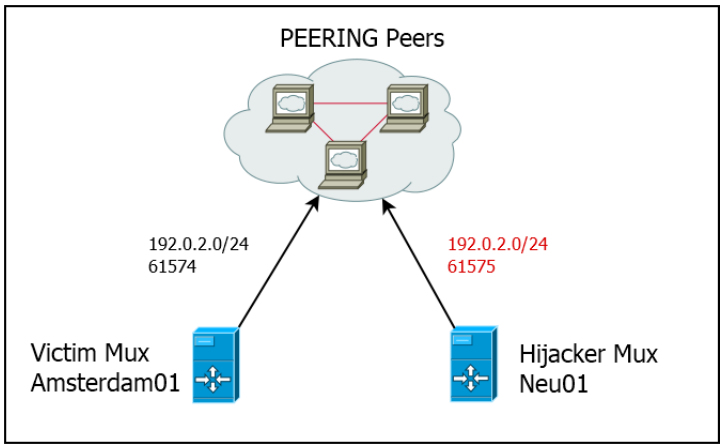
We use the PEERING [35] testbed to generate announcements that propagate on the Internet and manipulate how those announcements are executed. We define our experiment methodology to simulate the impact of prefix hijack events in three steps:

Original Announcement: The experiment begins with the victim AS announcing a prefix, referred to as the *original announcement*, as illustrated in Figure 9. A waiting period of 15 minutes follows to allow this announcement to propagate, during which control-plane data is collected. This propagation interval is consistent with the methodology adopted by Rizvi et al. [34]. After this period, a series of data-plane measurements targeting different hosts and ASes are conducted over 15 minutes. This step aims to analyze the routing behavior of the *original announcement* in isolation.

Hijack Announcement: After this period, the hijacker AS announces the same prefix, or a more specific one, using a different ASN as the origin. This step constitutes the *hijack announcement*. During this phase, we repeat the measurements and collect data from both the control plane and the data plane to evaluate the effectiveness of the hijacking attempt.

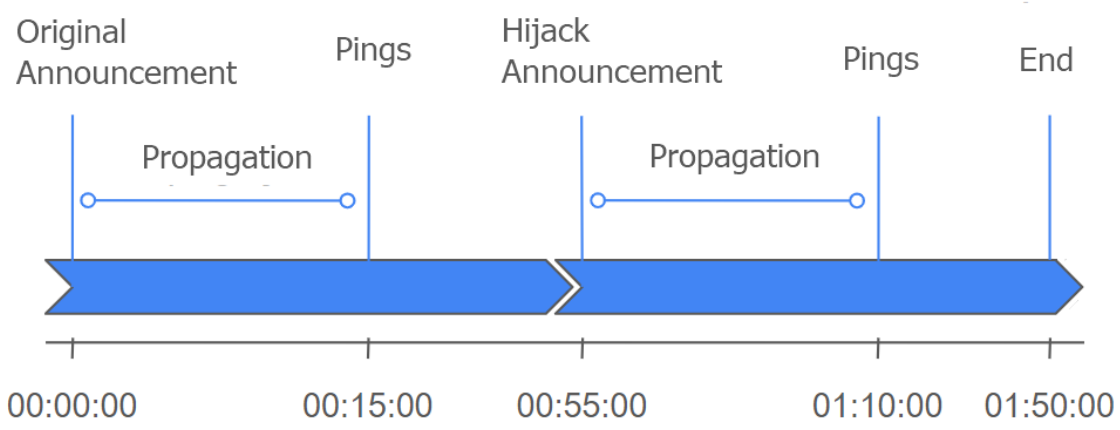
When propagating an announcement on the Internet, each AS must pass the update to its neighbors; not only is some time required until the announcement is passed to all

Figure 9: Example of hijack, where the victim AS (amsterdam01) announces the prefix 192.0.2.0/24 to PEERING peers and the hijacker (neu01) attempts to hijack the prefix with a different ASN as origin.



possible ASes, but we also need to wait until routes are stable, given each AS's routing policies and BGP route selection criteria. As such, we define for each step a waiting period of 15 minutes after a prefix is announced to wait for that route to propagate.

Figure 10: Timeline of steps executed for every experiment round.



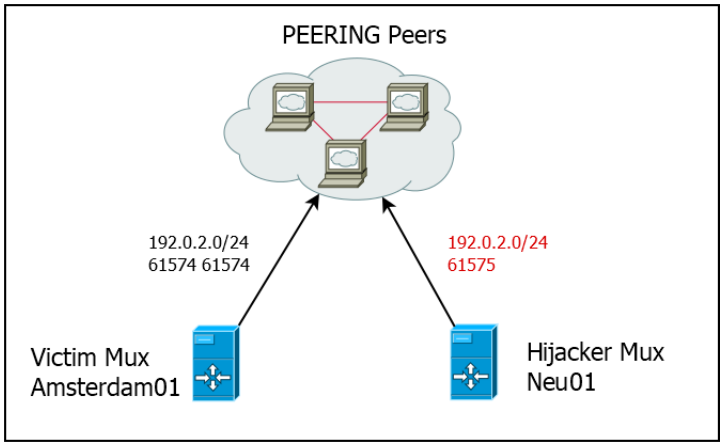
4.2 Internet Traffic Engineering (ITE) Techniques

The announcements were tailored to collect data and measure the impact of the following techniques: Prepend, Prefix Length, Selective Announcements. We define each ITE configuration in the following sub-sections. We use two PEERING muxes, amsterdam01 and Neu01 to exemplify scenarios for each ITE technique.

4.2.1 Prepend

Prepends aim to increase the AS path size, since BGP route selection criteria consider the path size. Manipulating prepends can lead to certain routes being less likely to be selected and used, and then move traffic according to the goals of an Autonomous System.

Figure 11: Example where the victim announce the prefix with prepend size 1, adding the ASN once again in the path. The hijacker announces without prepends.



Using prepend, we analyze the impacts considering different *prepend* lengths, ranging from 0 to 3, as illustrated in Figure 11. In the figure, we can see the example where amsterdam01 is the victim mux while neu01 is the attacker. The victim, AS 61574, announces its prefix, 192.0.2.0/24, with a prepend size of one, adding its own ASN one time. The attacker announces the same prefix, without prepends and changing the origin ASN to 61575.

4.2.2 Prefix Length

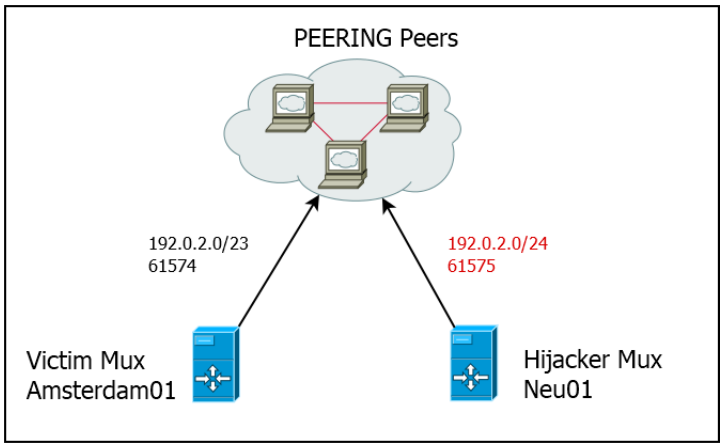
In the case of more specific announcements, we investigate how prefix specificity affects the success of a hijacking event. This involves performing the *original announcement* with a /23 or /24 prefix, as shown in Figure 12. The attack, however, always uses a /24 prefix.

In this example, the victim, amsterdam01, announces 192.0.2.0/23 using ASN 61574, without any other ITE. The attacker, neu01, then announces a longer, more specific, prefix of the victim, 192.0.2.0/24, with the origin ASN 61575.

4.2.3 Selective Announcements

For selective announcements, we restrict the prefix advertisement to a subset of the existing connections between the victim PEERING mux and its peers. In our experiments, propagation is limited to one of the following cases: (i) all neighbors; (ii) only IXPs; (iii) only transit ASes; (iv) subsets of transit ASes; or (v) subsets of IXPs. Since selective

Figure 12: Example where the victim announces the prefix as a /23, while the hijacker announces a /24 prefix disaggregated from the victim /23.



announcements restrict the set of neighbors that receive the route, this configuration also allows us to analyze whether the number of neighbors of an AS impacts the security of its prefixes against hijacking.

Figure 13: Selective announcement example, where the victim only passes the announcement to a single neighbor. The hijacker is announcing to all of its neighbors.

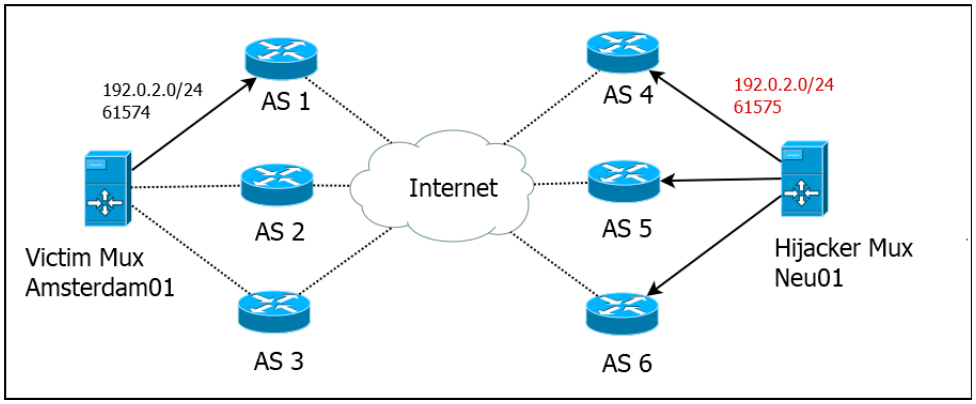


Figure 13 illustrates an example in which the victim, amsterdam01 (ASN 61574), announces the prefix 192.0.2.0/24 to one of its three neighbors (AS 1), thereby limiting inbound routes to the victim. The attacker, ASN 61575 on mux neu01, announces the same prefix with its ASN as the origin to all of its neighbors.

4.3 Data Collection

To measure the potential impact, we collect data from both the control plane and the data plane. Control-plane data provides insight into the routing decisions made by

ASes and the paths they select, allowing us to identify the causes of a hijack by comparing announcements and updates against BGP route-selection criteria. Although RIS Live provides real-time updates, it may lack the visibility required for a full assessment; therefore, we complement this with data-plane measurements.

Control-Plane Data Collection: During each experiment, we monitor control-plane information to track the routing behavior of both the victim and the attacker at every stage. This enables us to determine, for each round, how each actor behaves and to evaluate the resulting impact.

Data-Plane Data Collection: Data-plane measurements are used to capture traffic behavior, complementing the control-plane perspective. We actively probe targets using *ICMP echo requests* and observe the responses. While measurements from external hosts toward the victim are possible, they are constrained by the number of vantage points. To overcome this, we initiate probes from the victim toward a predefined list of hosts.

4.4 Metrics

To assess the impact of a hijack during an experiment, we define two metrics: (1) *Theoretical Impact* and (2) *Real Impact*.

Theoretical Impact: Extrapolates beyond the observed results by considering the customer cones of the affected networks, estimating the potential scope of impact that is not directly visible in our measurements. This metric captures how many networks *could* be affected and potentially divert their traffic toward the hijacker. Together with the real impact, it provides an estimate of the *minimum* and *maximum* impact for each scenario.

Real Impact: Is derived from control-plane and data-plane observations, identifying which networks actually accepted the hijack and what traffic was diverted toward the attacker. This metric allows us to quantify the number of ASes that accepted the hijack, the number of data-plane hosts that responded to the hijacker, and their proportions relative to the ASes and hosts observed in the original announcement.

5 EXPERIMENT TOOLS AND ENVIRONMENT

In this Chapter, we detail the tools and configurations used to evaluate the security implications of traffic engineering techniques following the methodology defined in Chapter 4. We define data collection in Section 5.2. Section 5.1 contains details regarding the PEERING muxes. Section 5.2 explores the configuration of tools for data collection. Finally, Section 5.3 consists of concerns regarding real traffic.

5.1 PEERING Muxes

In order to execute the announcements for the experiment, we must decide on which muxes to use. As such, we propagate a test announcement to check which muxes are responsive and if their announcement reaches the majority RIS Live monitors, out of the approximately 400 available. In Table 1 we present the results for each mux. Most of the PEERING muxes are capable of reaching more than 90% of the available monitors. The exceptions all stay below 10% of monitors. As such, we observed that a mux would either work as intended and reach most of the monitors or would fail noticeably, with very few monitors seeing the announcement.

Although the results for muxes in Vultr networks—one of PEERING's transit providers—were initially promising, we identified an issue when using a PEERING parameter to change the origin ASN of the announcement. In such cases, Vultr did not propagate the announcements. The matters identified with specific locations were reported and, in some instances, corrected; for example, muxes present in Vultr networks were reinstated as valid points to propagate announcements. A similar situation occurred with *ufmg01*, which in the first trial reached only eleven (11) RIS Live monitors.

We select muxes based on geodiversity, one mux for each of the following regions: Europe, North America, South America, Africa, and Asia-Pacific. Muxes are automatically selected for regions that only have a single mux available at the time; for example, vtrjohannesburg for Africa. For regions with multiple muxes available, we then considered the second hop for each mux and its variety.

Considering Europe, the mux with the most peers and connectivity options is amster-

Mux	Monitors	Percentage of Monitors Seen
amsterdam01	404	99.26%
seattle01	34	8.35%
saopaulo01	2	0.49%
ufmg01	11	2.70%
vtrseattle	377	92.63%
gatech01	379	93.12%
grnet01	389	95.58%
isi01	2	0.49%
neu01	379	93.12%
sbu01	1	0.25%
wisc01	379	93.12%
clemson01	380	93.37%
vtramsterdam	389	95.58%
vtratlanta	376	92.38%
vtrbangalore	407	100.00%
vtrchicago	376	92.38%
vtrdallas	376	92.38%
vtrdelhi	404	99.26%
vtrfrankfurt	389	95.58%
vtrjohannesburg	3	0.74%
vtrlondon	385	94.59%
vtrlosangelas	375	92.14%
vtrmadrid	377	92.63%
vtrmelbourne	374	91.89%
vtrmexico	375	92.14%
vtrmiami	379	93.12%
vtrmumbai	407	100.00%
vtrnewyork	379	93.12%
vtrosaka	375	92.14%
vtrparis	381	93.61%
vtrsaopaulo	381	93.61%
vtrseoul	380	93.37%
vtrsilicon	376	92.38%
vtrsingapore	381	93.61%
vtrstockholm	380	93.37%
vtrsydney	375	92.14%
vtrtokyo	376	92.38%
vtrtoronto	374	91.89%
vtrwarsaw	379	93.12%

Table 1: PEERING muxes and visibility of announcement on RIS Live.

dam01, being also the mux with the most variety of ASes as the second hop during the initial trials. The number of hops can be seen in Table 2. [1]

[1]Muxes with only a single AS as their second hop were omitted from the table. This includes all Vultr muxes, gatech01, grnet01, isi01, neu01, sbu01, clemson01 and wisc01.

Mux	Peers IPv4
amsterdam01	97
seattle01	61
saopaulo01	8
ufmg01	6

Table 2: Number of IPv4 Peers for each mux.

For South America, saopaulo01 was non-responsive, we are then left with ufmg01 and another mux in Vultr. We decided to use ufmg01 due to the second hop options in the mux, for example, IXP and Transit.

We are then left to decide which mux will be selected for North America and Asia-Pacific. While there are options of muxes in North America, some were unreliable since the first trials, for example seattle01. As such, we decided to continue with clemson01 and gatech01 despite them having only a single AS as second hop. Both muxes behaved and were successful in propagating announcements for the first rounds using ITE. Later, both muxes became unreliable and would not propagate announcements at times. We then selected neu01 to continue representing North America. For Asia-Pacific we decided on using vtrseoul.

Considering the mux selection for each region, we have selected the muxes in the following list: amsterdam01, ufmg01, neu01, vtrjohannesburg, vtrseoul.

5.2 Data Collection

Data collection for each experiment targets both the control plane and the data plane, as defined in Chapter 4.

To mitigate RIS Live's limited monitor coverage, we complement control-plane observations with data-plane measurements. At each experiment step, the victim sends *ICMP echo requests* to a list of targets, repeating each probe three times to reduce failures.

The target list is derived from the ANT IP list [13]. After filtering for responsive hosts, we obtain approximately two million active addresses across 40,000 ASes. To prevent overrepresentation of networks dominating the target set, we limit the sample to five addresses per AS. To enhance geographical diversity, we determine the location of each address using MaxMind[2]. The following criteria are then applied to select the final subset from ASes exceeding the five-address limit:

1. Addresses that cannot be located will be removed until there are 5 targets per ASes;
2. Limit to one target per town, allow more than one target in a single town if it's needed to achieve 5 targets per AS;

[2]Geolocation databases may contain inaccuracies.

3. Limit to one target per country, allow more than one target in a single country if it's needed to achieve 5 targets per AS;

4. Limit to one target per continent, allow more than one target in a single continent if it's needed to achieve 5 targets per AS;

With the steps enumerated above, we were able to limit ourselves to, at maximum, 5 targets per AS to a total of 127,421 targets. This provides diversity of location and diminishes the impact of large ASes with many targets in the dataset.

To conduct the data plane measurements, we used the *nping* tool [17], with the capture option disabled. When capture is enabled, nping will wait, until timeout, for a reply before sending the next ping; as such, we disable this option to send pings even to targets that are unresponsive at the time without needing to wait. Since the capture option was disabled, we required an alternative method to capture the replies from responsive targets.

Packet capture for the responsive targets will be handled via *tcpdump* instances, which will monitor packets on the interfaces used during the experiments [17, 19]. Through PEERING, the ingress route of a response from the data plane measurement, determining whether the hijacker or the original announcer received it, is identified using the MAC address or the interface on which the packet was received.

Consequently, all packets associated with the prefix used in the experiments are captured for analysis. Only one experiment will be conducted at a time, with a single PEERING client communicating with the involved muxes to specify the announcement configuration to be used. For example, only mux amsterdam01 and neu01 will be propagating announcements, with ping replies being received on each respective network interface.

5.3 Interference with Real Traffic

It is important to emphasize that none of the prefix announcements made during the experiments interfered with real user traffic, as the PEERING platform does not have any clients. Additionally, we include contact information in the data plane measurements to allow ASes to request exclusion from probing. We also limit the ping rate per second to avoid overloading vantage points or the measurement targets.

6 IMPACT OF TRAFFIC ENGINEERING ON HIJACKS

This chapter presents the results of our experiments, designed to evaluate how Internet Traffic Engineering (ITE) techniques influence the success and impact of prefix hijacks. Building on the methodology described in Chapter 4 and the configuration in Chapter 5, we analyze how specific techniques increase the likelihood of a hijack and examine the effects of forged-origin prefix attacks on the victim.

As introduced earlier, a prefix hijack involves three participants: the victim (the legitimate prefix owner), the attacker (who performs the hijack), and the impacted ASes (those that accept the hijacking announcement). We present our results following this structure: first, the impact on the victim; second, the characteristics that a hijacker can exploit; and third, the role of impacted ASes in shaping the AS path.

In addition, we investigate how the AS connectivity influences the probability of a successful hijack and analyze the characteristics of ASes that accept the hijacker's announcement, with the goal of identifying factors contributing to their vulnerability.

The experiments reported in this chapter provide evidence to address the research questions posed in Chapter 1: *(RQ1)* How do different traffic engineering practices affect the impact of a prefix hijack; *(RQ2)* Which characteristics of the victim influence the outcome of a hijack; *(RQ3)* Which characteristics of the attacker influence the outcome of a hijack; *(RQ4)* What leads an AS to accept a hijack announcement; *(RQ5)* Based on these results, what is the current state of traffic engineering employment on the Internet, and its possible impact on security.

This chapter is organized as follows. Section 6.1 discusses the results of the prepend experiments. Section 6.2 details the impact of hijacks when varying prefix length. Section 6.3 analyzes the effect of connectivity and selective announcements.

6.1 Prepend

We begin by analyzing the prepend technique, as described in Subsection 4.2.1, using an IPv4 /24 prefix as the *original announcement*. In this scenario, the victim applies different prepend lengths (0 to 3) on its announcement through PEERING, while the

attacker does not apply any ITE. We evaluate cases where each mux alternates as the victim or as the hijacker.

Figure 14: Impact of prepend experiments for each victim.

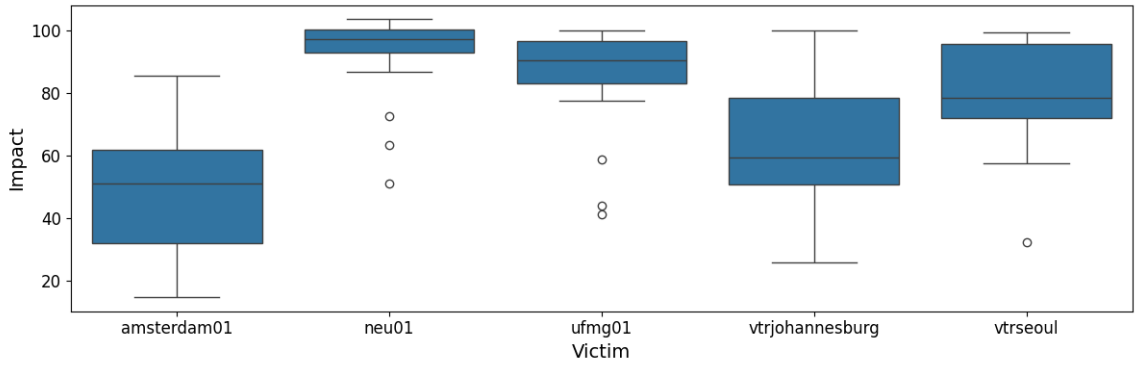


Figure 14 summarizes the overall results across PEERING muxes. The boxplots highlight that, in most cases, higher prepend values correlate with higher hijack impact: the minimums correspond to no prepend, and the maximums to three prepends. An exception is *neu01*, where all configurations already show high vulnerability, indicating that even without prepends this mux remains at risk.

6.1.1 Amsterdam01 as Victim

We start our analysis with *amsterdam01* as the victim. This mux has relatively strong connectivity in PEERING, which allows us to observe how prepend usage interacts with a well-connected origin. The goal of this experiment is to evaluate whether increasing the number of prepends alters the victim's resilience against different hijackers.

Table 3 summarizes the results when *amsterdam01* originates the prefix. Both control-plane and data-plane measurements show that prepend size directly amplifies the impact of a hijack. For instance, without prepends, between 57 (*neu01*) and 117 (*vtrjohannesburg*) monitors were hijacked; with three prepends, these numbers rose to 211 and 336, respectively. In the data plane, the proportion of hijacked targets increased from about 20% to more than 90%. These results confirm that prepend usage can substantially worsen hijack outcomes for the victim, even when the AS is well connected.

6.1.2 Neu01 as Victim

We now consider *neu01* as the victim. Unlike *amsterdam01*, this mux has weaker connectivity, which allows us to assess how prepend usage interacts with a less resilient origin.

The results in Table 4 show that the prepend size produces little variation in some scenarios. For example, against *amsterdam01* as the attacker, 348 monitors were hijacked

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	0	391	57 (14.57%)	100171	19609 (19.57%)
amsterdam01	neu01	1	390	128 (32.82%)	99878	50246 (50.30%)
amsterdam01	neu01	2	393	186 (47.32%)	99918	71538 (71.59%)
amsterdam01	neu01	3	386	211 (54.66%)	99916	78831 (78.89%)
amsterdam01	ufmg01	0	391	92 (23.52%)	100180	32642 (32.58%)
amsterdam01	ufmg01	1	390	161 (41.28%)	100007	60404 (60.39%)
amsterdam01	ufmg01	2	388	219 (56.44%)	99752	80739 (80.93%)
amsterdam01	ufmg01	3	389	240 (61.69%)	100161	84170 (84.03%)
amsterdam01	vtrseoul	0	392	69 (17.60%)	99904	20379 (20.39%)
amsterdam01	vtrseoul	1	389	166 (42.67%)	99875	57586 (57.65%)
amsterdam01	vtrseoul	2	390	243 (62.30%)	100036	85256 (85.22%)
amsterdam01	vtrseoul	3	387	261 (67.44%)	100116	88473 (88.37%)
amsterdam01	vtrjohannesburg	0	392	117 (29.84%)	100175	34229 (34.16%)
amsterdam01	vtrjohannesburg	1	390	240 (61.53%)	99827	64593 (64.70%)
amsterdam01	vtrjohannesburg	2	378	318 (84.12%)	99876	89523 (89.63%))
amsterdam01	vtrjohannesburg	3	392	336 (85.71%)	99904	93529 (93.61%)

Table 3: Impact of 0, 1, 2, and 3 prepends on amsterdam01 as victim.

without prepends, while with one prepend the number rose only to 371. This suggests that *amsterdam01* already offers shorter paths to many monitors compared to *neu01*, making hijacking more effective regardless of prepend configuration. This finding underscores the importance of connectivity in routing security.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
neu01	amsterdam01	0	363	348 (95.86%)	99786	72218 (72.37%)
neu01	amsterdam01	1	363	374 (103.03%)	99573	87131 (87.50%))
neu01	amsterdam01	2	361	371 (102.77%)	99536	88874 (89.28%)
neu01	amsterdam01	3	360	372 (103.33%)	99629	89047 (89.37%)
neu01	ufmg01	0	363	187 (51.51%)	99690	52711 (52.87%)
neu01	ufmg01	1	361	350 (96.95%)	99507	94490 (94.95%)
neu01	ufmg01	2	361	349 (96.67%)	99698	94723 (95.00%)
neu01	ufmg01	3	358	349 (97.48%)	99583	94609 (95.00%)
neu01	vtrseoul	0	363	237 (65.28%)	99640	48552 (48.72%)
neu01	vtrseoul	1	361	314 (86.98%)	99656	81614 (81.89%)
neu01	vtrseoul	2	360	348 (96.66%)	99599	88606 (88.96%)
neu01	vtrseoul	3	361	351 (97.22%)	99636	89293 (89.61%)
neu01	vtrjohannesburg	0	362	273 (75.41%)	99475	55170 (55.46%)
neu01	vtrjohannesburg	1	361	366 (101.38%)	99487	94468 (94.95%)
neu01	vtrjohannesburg	2	361	366 (101.38%)	99384	94397 (94.98%)
neu01	vtrjohannesburg	3	360	367 (101.94%)	83721	78716 (94.02%)

Table 4: Impact of 0, 1, 2, and 3 prepends on neu01 as the victim.

Two key observations emerge from these experiments. First, approximately 20% of monitors or targets were not hijacked, suggesting that these ASes—or others along their paths—may rely on local preference policies, resisting the hijack despite the attacker's

announcements. Second, in some cases the number of hijacked monitors exceeded 100% of the number of monitors seen in the *original announcement*. We hypothesize that this anomaly is caused by filtering in *neu01* that is not present in *amsterdam01* or *vtrjohannesburg*, which makes announcements from those muxes more effective than *neu01*.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
neu01	amsterdam01	3	360	372 (103.33%)	99629	89047 (89.37%)
neu01	ufmg01	3	358	349 (97.48%)	99583	94609 (95.00%)
neu01	vtrseoul	3	361	351 (97.22%)	99636	89293 (89.61%)
neu01	vtrjohannesburg	3	360	367 (101.94%)	83721	78716 (94.02%)

Table 5: Impact of using 3 prepends on neu01 as victim

Table 5 further illustrates the severity of this scenario: when *neu01* announced with three prepends, every mux was able to hijack a large share of its traffic. This shows that ASes with weaker connectivity, such as *neu01*, can be severely affected even with minimal prepend usage. In practice, this not only amplifies the potential damage of a hijack but also limits mitigation options, since removing prepends from the victim would have little effect on traffic recovery when the victim AS has weaker connectivity.

6.1.3 Ufmg01 as Victim

In the configuration where *ufmg01* is the victim, the results in Table 6 show trends similar to those observed for *neu01*: the hijack impact grows with the number of prepends. However, the baseline impact without prepends is already higher than in the case of *amsterdam01* as the victim. This indicates that prepend usage does influence hijack outcomes, but its effect is relative to the connectivity of the origin AS.

In practice, *ufmg01* is not safe against hijacks even without prepends. With only one prepend, the proportion of hijacked monitors rises from about 78% to 96%, showing that limited connectivity amplifies the risk and leaves little margin for mitigation.

6.1.4 VtrJohannesburg as Victim

For the mux present in Johannesburg, the results show less impact compared to *ufmg01* and *neu01*. Table 7 indicates that with 0 prepends the hijack effect is significant, but it does not affect the majority of monitors—except when *amsterdam01* is the attacker. Excluding *amsterdam01*, the results for the other muxes with 2 and 3 prepends are very similar. Due to PEERING limitations, we could not test with more than three prepends, and therefore cannot determine whether this pattern would persist with larger prepend values.

These results highlight once again that hijack outcomes depend not only on the victim's characteristics but also on those of the attacker. In this case, the effectiveness of

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
ufmg01	neu01	0	362	157 (43.37%)	99733	39768 (39.87%)
ufmg01	neu01	1	364	285 (78.29%)	99772	79586 (79.76%)
ufmg01	neu01	2	363	315 (86.77%)	99793	86669 (86.84%)
ufmg01	neu01	3	364	319 (87.63%)	99816	88291 (88.45%)
ufmg01	amsterdam01	0	362	310 (85.63%)	99835	60619 (60.71%)
ufmg01	amsterdam01	1	363	349 (96.14%)	99752	81572 (81.77%)
ufmg01	amsterdam01	2	364	361 (99.17%)	99782	83813 (83.99%)
ufmg01	amsterdam01	3	363	361 (99.44%)	99767	84605 (84.80%)
ufmg01	vtrseoul	0	362	152 (41.98%)	99834	32995 (33.04%)
ufmg01	vtrseoul	1	364	291 (79.94%)	99859	55684 (55.76%)
ufmg01	vtrseoul	2	363	326 (89.80%)	99736	86501 (86.72%)
ufmg01	vtrseoul	3	363	327 (90.08%)	99791	82174 (82.34%)
ufmg01	vtrjohannesburg	0	362	212 (58.56%)	99580	26832 (26.94%)
ufmg01	vtrjohannesburg	1	364	338 (92.85%)	99687	62121 (62.31%)
ufmg01	vtrjohannesburg	2	363	346 (95.31%)	99587	88632 (88.99%)
ufmg01	vtrjohannesburg	3	362	351 (96.96%)	99665	89611 (89.91%)

Table 6: Impact of 0, 1, 2, and 3 prepends on ufmg01 as victim.

amsterdam01 demonstrates that the safety of using prepends is relative to the level of threat and the expected origin of a potential hijack.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
vtrjohannesburg	neu01	0	377	107 (28.38%)	99941	41520 (41.54%)
vtrjohannesburg	neu01	1	379	173 (45.64%)	99840	63519 (63.62%)
vtrjohannesburg	neu01	2	378	200 (52.91%)	99885	69876 (69.95%)
vtrjohannesburg	neu01	3	379	204 (53.82%)	99604	70769 (71.05%)
vtrjohannesburg	ufmg01	0	378	153 (40.47%)	99985	54377 (54.38%)
vtrjohannesburg	ufmg01	1	379	217 (57.25%)	99806	71531 (71.67%)
vtrjohannesburg	ufmg01	2	380	236 (62.10%)	99894	74927 (75.00%)
vtrjohannesburg	ufmg01	3	380	241 (63.42%)	99834	84834 (84.97%)
vtrjohannesburg	vtrseoul	0	379	118 (31.13%)	99996	33581 (33.58%)
vtrjohannesburg	vtrseoul	1	379	196 (51.71%)	99912	60838 (60.89%)
vtrjohannesburg	vtrseoul	2	378	249 (65.87%)	99884	68802 (68.88%)
vtrjohannesburg	vtrseoul	3	378	252 (66.66%)	99879	71159 (71.24%)
vtrjohannesburg	amsterdam01	0	378	312 (82.53%)	99974	59130 (59.14%)
vtrjohannesburg	amsterdam01	1	379	347 (91.55%)	99803	72787 (72.93%)
vtrjohannesburg	amsterdam01	2	379	349 (92.08%)	99815	74598 (74.73%)
vtrjohannesburg	amsterdam01	3	379	351 (92.61%)	99205	74711 (75.30%)

Table 7: Impact of 0, 1, 2, and 3 prepends on vtrjohannesburg as victim.

6.1.5 VtrSeoul as Victim

Although *vtrseoul* shows results similar to *vtrjohannesburg*, we can observe two key differences in Table 8: (1) the impact is slightly higher, even though both muxes are in Vultr networks, suggesting that geolocation *might* influence the outcome; and (2) both

amsterdam01 and *vtrjohannesburg* were effective in hijacking traffic from *vtrseoul*.

We can also observe that when *ufmg01* is the attacker, the results for 1, 2, and 3 prepends are very similar. Similar to the case of *vtrjohannesburg* as the victim, PEERING limitations prevented us from announcing with more than three prepends, so we cannot determine whether the impact would increase further with larger prepend values.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
vtrseoul	neu01	0	364	117 (32.14%)	99993	37739 (37.74%)
vtrseoul	neu01	1	364	222 (60.98%)	96658	69131 (71.52%)
vtrseoul	neu01	2	365	270 (73.97%)	99973	83221 (83.24%)
vtrseoul	neu01	3	363	271 (74.65%)	99887	83978 (84.07%)
vtrseoul	ufmg01	0	363	209 (57.57%)	100102	61962 (61.89%)
vtrseoul	ufmg01	1	364	269 (73.90%)	99914	82363 (82.43%)
vtrseoul	ufmg01	2	365	288 (78.90%)	99913	85316 (85.39%)
vtrseoul	ufmg01	3	363	284 (78.23%)	99929	85521 (85.58%)
vtrseoul	amsterdam01	0	364	335 (92.03%)	100173	74981 (74.85%)
vtrseoul	amsterdam01	1	363	353 (97.24%)	100043	89601 (89.56%)
vtrseoul	amsterdam01	2	365	363 (99.45%)	99967	90143 (90.17%)
vtrseoul	amsterdam01	3	362	359 (99.17%)	99813	90074 (90.24%)
vtrseoul	vtrjohannesburg	0	363	241 (66.39%)	99608	56957 (57.18%)
vtrseoul	vtrjohannesburg	1	364	338 (92.85%)	99777	89763 (89.96%)
vtrseoul	vtrjohannesburg	2	365	350 (95.89%)	99831	92276 (92.43%)
vtrseoul	vtrjohannesburg	3	363	348 (95.86%)	99624	92318 (92.66%)

Table 8: Impact of 0, 1, 2, and 3 prepends on vtrseoul as victim.

6.1.6 Theoretical Impact

To better assess the extent of a prefix hijack, we extend our analysis beyond the direct visibility provided by RIS Live monitors by also considering the *AS customer cone* associated with each monitor from the AS-Rank dataset [8]. Analyzing these cones enables us to estimate the potential scope of networks impacted by a hijack, thus providing a broader perspective on the theoretical impact and complementing the observations from control-plane and data-plane measurements. The table reports three categories of ASes within the control plane:

- **Unaffected:** ASes that do not have any relationship with hijacked ASes. These are expected to remain loyal to the original announcement (*amsterdam01*) and to ignore the hijack.
- **Intersection:** ASes that have at least one transit or peering relationship that has accepted the attacker's announcement—we observe a BGP update from RIS monitor indicating it selected the attacker's route.
- **Affected:** ASes that maintain relationships only with other ASes that have been proven to be affected by the hijack.

The expected result is for targets that were hijacked to be present in either the intersection set or the affected AS set. For the data plane targets that were not hijacked, we expect them to be in the AS set of unaffected ASes or the intersection set.

Experiment Configuration			Control Plane			
Origin	Hijacker	Prepend Size	Total	Unaffected	Intersection	Affected
amsterdam01	neu01	0	74393	13474	55355	5564
amsterdam01	ufmg01	0	74297	10519	56487	7291
amsterdam01	vtrseoul	0	74308	15953	55271	3084
amsterdam01	vtrjohannesburg	0	74297	10497	55975	7825

Table 9: Control plane cone impact of 0 prepends on amsterdam01 as victim.

In Table 9 we show the AS customer cones when *amsterdam01* is the victim with 0 prepends. An interesting observation is that *neu01*, although generally the least effective attacker, has more networks visible only in the affected cone than *vtrseoul*. This suggests that, in theory, *neu01* should have a greater impact than *vtrseoul*. The same can be seen on the data plane, where we have 1354 ASes with hijacked targets that are in the *unnafected* cone, as seen in Table 10.

However, this expectation does not match the actual control-plane and data-plane results, where both attackers achieve similar outcomes for prepend 0. This discrepancy can be explained by the limited visibility of AS relationships: networks seen in the affected cone of *neu01* may also belong to the victim's cone, but such relationships are not publicly known or observable.

Experiment Configuration			Data Plane Safe			Data Plane Hijacked		
Origin	Hijacker	Prepend Size	Unaffected	Intersection	Affected	Unaffected	Intersection	Affected
amsterdam01	neu01	0	6053	25008	1299	643	6164	1642
amsterdam01	ufmg01	0	4490	21418	1566	858	10528	2206
amsterdam01	vtrseoul	0	7143	24198	815	1354	6358	741
amsterdam01	vtrjohannesburg	0	4586	21244	1221	1065	10678	2603

Table 10: Data plane cone impact of 0 prepends on amsterdam01 as victim.

6.1.7 Key Findings on Prepend

Consistent with previous studies [12], our results confirm that increasing the number of prepends directly affects the impact of a prefix hijack event. The impact of a hijack is influenced both by the prepend size and by the connectivity of the origin AS. Importantly, there is no number of prepends that can be considered universally safe; their effectiveness is relative to the victim's connectivity and to the attacker's position. While *amsterdam01* benefits from stronger connectivity, which reduces hijack impact, *neu01* shows more uniform vulnerability across prepend sizes. In addition, some ASes appear to rely on local preference policies to resist hijacks, underscoring the role of routing policies in mitigat-

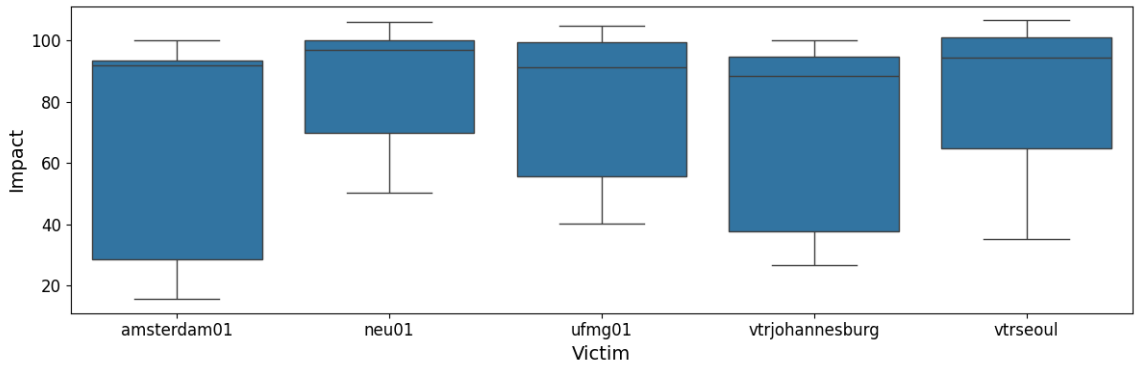
ing attacks. Overall, the findings show that attackers can exploit prepends to increase the impact of prefix hijacking.

6.2 Prefix Length

Prefix length plays a crucial role in the outcome of hijack events, since more specific prefixes are preferred in BGP route selection. To evaluate its impact, we designed the experiment configuration described in Subsection 4.2.2, defining two scenarios for each mux pair. In the first, the victim announces a /23 prefix for the *original announcement*. In the second, the victim announces a /24. In both cases, the attacker uses a /24 prefix to execute the hijack attempt.

Similar to the prepend experiments, Figure 15 presents a boxplot with an overview of the results for each mux. For all muxes, the maximum values are similar, confirming the effectiveness of a hijack using a more specific prefix.

Figure 15: Impact of prefix length experiments for each victim.



6.2.1 Amsterdam01 as Victim

When *amsterdam01* is the victim, a hijack with a more specific prefix impacts most, but not all, monitors, as shown in Table, this could be due to some monitors not having visibility of the announcement. 11[1]. When the prefix lengths are equal between the victim and the attacker, however, the impact is significantly reduced.

Announcing only /24 prefixes increases the size of RIBs, which can expose memory limitations in Internet devices and raise operational costs. Operational workload is also a factor, since managing only /24s raises the number of announcements and reduces the ability to use prefix length for traffic engineering, thereby forcing operators to rely on other techniques.

[1]Experiment failed to propagate the victim's original announcement for amsterdam01 vs. vtrseoul with a /24 prefix for the victim.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Victim Prefix Length	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	/23	391	364 (93.09%)	102631	102600 (99.96%)
amsterdam01	neu01	/24	388	60 (15.46%)	102591	22738 (22.16%)
amsterdam01	ufmg01	/23	390	365 (93.58%)	103050	103019 (99.96%)
amsterdam01	ufmg01	/24	393	93 (23.66%)	103020	35394 (34.35%)
amsterdam01	vtrseoul	/23	390	362 (92.82%)	102983	102963 (99.98%)
amsterdam01	vtrseoul	/24	N/A	73 (N/A)	N/A	N/A
amsterdam01	vtrjohannesburg	/23	389	380 (97.68%)	102801	102785 (99.98%)
amsterdam01	vtrjohannesburg	/24	389	119 (30.59%)	102745	36699 (35.71%)

Table 11: Impact of /23 or /24 prefix length for original announcement on amsterdam01 as victim. Hijack announcement uses a /24 prefix.

6.2.2 Neu01 as Victim

The results for *neu01* are presented in Table 12 for both the data plane and the control plane. A hijack with a more specific prefix is successful when the victim announces a /23 and the attacker announces a /24, with most monitors accepting the hijack announcement.

An important observation is that not all monitors that responded to the original announcement accepted the hijack announcement. This may occur because the peering agreement between PEERING and some ISPs *may* only allow /24 prefixes. Another possible explanation is visibility limitations: announcements made by the victim may not reach all RIS Live monitors, whereas the attacker's announcement may reach more monitors, thereby leading to more targets being hijacked than those observed in the *original announcement*.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Victim Prefix Length	Total	Hijacked	Total	Hijacked
neu01	amsterdam01	/23	363	389 (107.16%)	102782	102782 (100.0%)
neu01	amsterdam01	/24	364	351 (96.42%)	102858	73383 (71.34%)
neu01	ufmg01	/23	364	366 (100.54%)	102583	102583 (100.0%)
neu01	ufmg01	/24	365	185 (50.68%)	102935	55573 (53.98%)
neu01	vtrseoul	/23	364	363 (99.72%)	102798	102798 (100.0%)
neu01	vtrseoul	/24	364	229 (62.91%)	102800	49716 (48.36%)
neu01	vtrjohannesburg	/23	365	379 (103.83%)	102661	102661 (100.0%)
neu01	vtrjohannesburg	/24	364	268 (73.62%)	102683	57340 (55.84%)

Table 12: Impact of /23 or /24 prefix lengths for the original announcement with neu01 as the victim. Hijack announcement uses a /24 prefix.

When the original announcement is disaggregated into /24 prefixes, the impact is reduced, similar to what was observed with prefixes without prepending (see Section 6.1). However, in this scenario, no viable mitigation would exist, since most ASes along the path filter /25 prefixes. Therefore, the connectivity of the *mux* remains a critical factor.

For attackers, /23 or shorter prefixes are prime targets for prefix hijacking. While a hijack of a /22 prefix using a /23, for example, would have a severe impact, it would still allow the victim to mitigate using a /24. However, if the attacker uses a /24, this limits the

mitigation options for the victim to recover traffic, depending on the victim's connectivity. As such, ASes that announce prefixes shorter than /24 and have weaker connectivity are particularly at risk.

6.2.3 Ufmg01 as Victim

When *ufmg01* is the victim, we observe similar results: more specific prefixes have greater impact during a hijack. In the case of *ufmg01*, the impact when both the attacker and the victim use the same prefix length is less pronounced than in *neu01*. Table 13 shows the results for the control and data plane; we can observe that although the impact is lower than for *neu01*, it still affects, in some scenarios, the majority of control- and data-plane targets, especially against *amsterdam01*.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Victim Prefix Length	Total	Hijacked	Total	Hijacked
ufmg01	neu01	/23	366	364 (99.45%)	102485	102473 (99.98%)
ufmg01	neu01	/24	367	169 (46.04%)	102893	42988 (41.77%)
ufmg01	amsterdam01	/23	369	391 (105.96%)	102766	102754 (99.98%)
ufmg01	amsterdam01	/24	366	313 (85.51%)	102916	62263 (60.49%)
ufmg01	vtrseoul	/23	365	361 (98.90%)	102667	102655 (99.98%)
ufmg01	vtrseoul	/24	365	148 (40.54%)	102784	33742 (32.82%)
ufmg01	vtrjohannesburg	/23	367	382 (104.08%)	102500	102488 (99.98%)
ufmg01	vtrjohannesburg	/24	365	220 (60.27%)	102603	43723 (42.61%)

Table 13: Impact of /23 or /24 prefix length for original announcement on ufmg01 as victim. Hijack announcement uses a /24 prefix.

In these scenarios, *ufmg01* would remain vulnerable to prefix hijacks even when not using prepends and announcing a /24 prefix. As such, any use of ITE by *ufmg01* may increase its vulnerability.

6.2.4 VtrJohannesburg as Victim

In this scenario, the mux located in Johannesburg results in only 26.5% of monitors being hijacked when the attacker is *neu01*. For other configurations where the victim announces a /24 prefix, the impact is lower than that observed for *ufmg01*. However, it does not reach the same level of resilience as *amsterdam01*, which still allows 83.64% of control-plane monitors to be hijacked, although only 57.66% of data-plane targets.

6.2.5 VtrSeoul as Victim

The mux located in Seoul shows results similar to *ufmg01*, as presented in Table 15[2]. In almost every configuration, more than 50% of control-plane monitors were hijacked, although against *neu01* the results were better than those of *ufmg01*.

[2]Experiment configuration vtrseoul vs. amsterdam01 with the victim propagating a /24 prefix failed to gather results for the data plane.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Victim Prefix Length	Total	Hijacked	Total	Hijacked
vtrjohannesburg	neu01	/23	381	365 (95.80%)	102773	102772 (99.99%)
vtrjohannesburg	neu01	/24	381	101 (26.50%)	102872	42327 (41.14%)
vtrjohannesburg	ufmg01	/23	381	365 (95.80%)	103034	103033 (99.99%)
vtrjohannesburg	ufmg01	/24	380	153 (40.26%)	102990	60638 (58.87%)
vtrjohannesburg	vtrseoul	/23	381	363 (95.27%)	102927	102926 (99.99%)
vtrjohannesburg	vtrseoul	/24	380	119 (31.31%)	103039	34081 (33.07%)
vtrjohannesburg	amsterdam01	/23	379	384 (101.31%)	102942	102941 (99.99%)
vtrjohannesburg	amsterdam01	/24	379	317 (83.64%)	102936	59361 (57.66%)

Table 14: Impact of /23 or /24 prefix length for original announcement on vtrjohannesburg as victim. Hijack announcement uses a /24 prefix.

Data-plane targets exhibited behavior similar to control-plane monitors. The only exception occurred during measurements with a /24 prefix and *amsterdam01* as the attacker, where a failure prevented complete results.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Victim Prefix Length	Total	Hijacked	Total	Hijacked
vtrseoul	neu01	/23	363	364 (100.27%)	102657	102657 (100.0%)
vtrseoul	neu01	/24	363	129 (35.53%)	102857	42997 (41.80%)
vtrseoul	ufmg01	/23	362	366 (101.10%)	102794	102794 (100.0%)
vtrseoul	ufmg01	/24	363	219 (60.33%)	102748	66743 (64.95%)
vtrseoul	amsterdam01	/23	364	392 (107.69%)	102874	102874 (100.0%)
vtrseoul	amsterdam01	/24	363	328 (90.35%)	*	* (*%)
vtrseoul	vtrjohannesburg	/23	362	380 (104.97%)	102491	102491 (100.0%)
vtrseoul	vtrjohannesburg	/24	363	244 (67.21%)	102430	61762 (60.29%)

Table 15: Impact of /23 or /24 prefix length for original announcement on vtrseoul as victim. Hijack announcement uses a /24 prefix.

Although both the Seoul and Johannesburg muxes are within Vultr, we observe different results in terms of prefix hijack impact. This may be due to geographic and topological differences between the two sites. Seoul is another example of a mux that remains vulnerable to hijacks even when not using ITE and announcing a /24 prefix. A better-connected AS, such as *amsterdam01*, can still cause severe impacts even if *vtrseoul* takes no action that negatively affects its security.

6.2.6 Theoretical Impact

Table 16 summarizes the results for *amsterdam01*, the best-performing mux, when it announces a /23 prefix and is the victim of hijacks with /24 prefixes. Considering the case of *amsterdam01* versus *neu01*, we observed 13 targets that were not affected by the hijack. Of these, 6 belong to the intersection set, while 7 are included in the affected ASes set. This suggests that the ASes containing those 7 targets may have been misclassified and should instead fall within either the unaffected set or the intersection set.

This discrepancy could be due to the number of RIS Live monitors being insufficient

Experiment Configuration			Control Plane			
Origin	Hijacker	Victim Prefix Length	Total	Unaffected	Intersection	Affected
amsterdam01	neu01	/23	73218	358	9301	63559
amsterdam01	ufmg01	/23	74297	339	9322	64636
amsterdam01	vtrseoul	/23	73133	358	9249	63526
amsterdam01	vtrjohannesburg	/23	74297	333	8044	65920

Table 16: Control plane cone impact, amsterdam01 as the victim announces a /23 prefix. Hijack announcement uses a /24 prefix.

to provide full visibility of the Internet topology regarding AS cones. When we observe the data plane targets, we also see another discrepancy: 147 targets that were hijacked are only seen in the cone of RIS Live monitors that were not hijacked.

These results again exemplify what was seen in the theoretical impact of the prepend experiments. There is either insufficient RIS Live coverage or relationships between ASes that are not publicly known or observable.

Experiment Configuration			Data Plane Safe			Data Plane Hijacked		
Origin	Hijacker	Victim Prefix Length	Unaffected	Intersection	Affected	Unaffected	Intersection	Affected
amsterdam01	neu01	/23	0	6	7	147	5357	33266
amsterdam01	ufmg01	/23	0	7	5	141	5431	33694
amsterdam01	vtrseoul	/23	0	1	5	147	5392	33268
amsterdam01	vtrjohannesburg	/23	0	1	2	140	4722	34291

Table 17: Data plane cone impact, amsterdam01 as the victim announces a /23 prefix. Hijack announcement uses a /24 prefix.

6.2.7 Key Findings on Prefix Length

The impact of prefix hijacking is strongly influenced by prefix length. When the attacker uses a prefix of equal length to the victim's announcement, the outcome is largely determined by connectivity. While announcing only `/24` prefixes may provide greater resilience against more specific hijacks, this practice increases the size of routing tables and imposes additional workload on network operators, making large-scale adoption impractical. Moreover, prefixes shorter than `/24` remain particularly vulnerable to hijacks performed with longer, more specific prefixes.

It is also important to note that even announcing a prefix as a `/24` does not guarantee protection. The results observed for the prepend technique with a prepend count of zero also apply in scenarios where the attacker and victim use prefixes of equal prefix length.

6.3 Selective Announcements and Connectivity

We conducted measurements using the selective announcement technique, as described in Subsection 4.2.3. Due to limitations in PEERING's connectivity, each *mux* offers different peering options. Among them, *amsterdam01* has the largest number of

peers and was therefore chosen as the single victim in these experiments. Other *muxes*, with fewer selective announcement options, were not used as victims but still participated as attackers.

In this configuration, both victim and attacker used prefixes of equal length, with no prepending applied. Selective announcements were directed to specific peers: AMS-IX (Amsterdam Internet Exchange), Bit BV, and Coloclue. Experiments were conducted with different combinations of these peers, for instance, announcing to AMS-IX and Bit BV simultaneously.

The results, summarized in Table 18, reveal that limiting announcements exclusively to IXPs significantly reduces the visibility of the victim's prefix. In one case, only 18 monitors observed the announcement. This reduced visibility amplifies the impact of a prefix hijack, since for most monitors the attacker's announcement is the only one visible.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Peers/IX	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	AMS-IX	18	345 (N/A)	1162	619 (53.27%)
amsterdam01	neu01	Bit BV	371	65 (17.52%)	98872	21726 (21.97%)
amsterdam01	neu01	AMS-IX, Bit BV	371	65 (17.52%)	98939	21853 (22.08%)
amsterdam01	neu01	Coloclue, Bit BV	386	63 (16.32%)	98794	21681 (21.94%)
amsterdam01	neu01	AMS-IX, Coloclue, Bit BV	386	65 (16.83%)	98779	21818 (22.08%))
amsterdam01	neu01	Coloclue	391	39 (9.97%)	98884	20967 (21.20%)
amsterdam01	neu01	AMS-IX, Coloclue	391	39 (9.97%)	98716	20980 (21.25%)

Table 18: Selective announcement results with amsterdam01 as victim.

Considering *neu01* as the attacker, we observe that announcing only to Coloclue yields better results than announcing only to Bit BV. Announcing only to Coloclue, 391 monitors responded to the original announcement, and only 39 (9.97%) were hijacked, compared to 371 monitors and 65 hijacked (17.52%) when announcing to a subset that contains Bit BV and excludes Coloclue. Contrary to expectations, announcing to all three peers (AMS-IX, Bit BV, and Coloclue) did not improve security compared to excluding Bit BV from the set: only 386 monitors responded to the original announcement, while 65 (16.83%) were hijacked. These results show that increasing the number of neighbors does not necessarily reduce the impact of a hijack.

This behavior could be the result of local preference or routing policies along the AS path, demonstrating that the number of neighbors does not directly correlate with security. In fact,relying on a single neighbor provided better results than using two neighbors and an IXP. showing the characteristics of the neighbors can influence the hijack impact. Therefore, ASes must consider not only the number of neighbors they maintain but also the routing policies of those neighbors when evaluating potential benefits. While maintaining multiple peering agreements may increase operational costs, in some cases, it can also strengthen security.

6.3.1 Key Findings on Selective Announcements

Connectivity is a critical aspect of AS operations, and in the event of a prefix hijack, the set of connections a victim maintains influences the impact. Our results show that the number of connections does not directly correlate with security. Instead, other factors—such as the behavior of neighboring ASes and local preference policies along the AS path—play a decisive role.

For instance, when announcing to a specific subset of neighbors results in a longer AS path, the effect can resemble that of AS path prepending. As demonstrated in the prepend experiments, such changes may determine whether an announcement remains relatively safe or whether traffic is entirely diverted to the hijacker.

7 MITIGATION

In the event of a prefix hijack, the primary objective of the victim is to reduce or eliminate its impact. To this end, several mitigation strategies can be employed, often by modifying how the prefix is announced to the rest of the Internet. In this chapter, we discuss mitigation options in the context of the traffic engineering techniques analyzed in the previous chapters.

7.1 Techniques

The impact of a prefix hijack depends on the number of ASes that accept the hijacker's announcement, which occurs when the announcement is selected as the best route according to BGP decision criteria. As shown in Chapter 6, attackers can exploit victims that rely on Internet Traffic Engineering (ITE), since such techniques may influence the severity of a hijack.

Victims, in turn, can attempt to mitigate the event by removing the use of ITE or by adopting alternative techniques to recover traffic. To evaluate this, we extend our experiments with an additional step:

Mitigation Announcement: The victim initiates a mitigation attempt, referred to as the *mitigation announcement*. In this stage, the victim announces more specific prefixes than in the *original announcement*, aiming to reduce the impact of the hijack. Measurements and data collection are again performed in both the control plane and the data plane.

The most effective mitigation strategy for a victim mirrors that of an attacker: announcing a longer prefix than the other party, provided the attacker is not already using a /24 prefix. However, since prefixes more specific than /24 are often filtered, mitigation using a prefix of equal length to the hijack announcement will rely on local preference and AS path length to influence route selection. While this approach can reduce the impact—as shown in Table 19—it does not guarantee full traffic recovery.

Another possible mitigation step is to remove prepends. In the case of *amsterdam01* vs. *neu01*, the impact of using three prepends compared to zero prepends was 40.09% of the monitors, as shown in Table 20. Removing prepends is therefore only partially

Experiment Configuration			Control Plane Monitors			Data Plane Targets		
Origin	Hijacker	Victim Prefix Length	Total	Hijacked	Recovered	Total	Hijacked	Recovered
amsterdam01	neu01	/23	391	364 (93.09%)	295 (81.04%)	102631	102600 (99.96%)	73270 (71.39%)
amsterdam01	ufmg01	/23	390	365 (93.58%)	264 (72.32%)	103050	103019 (99.96%)	62746 (60.88%)
amsterdam01	vtrseoul	/23	390	362 (92.82%)	286 (79.00%)	102983	102963 (99.98%)	78777 (76.49%)
amsterdam01	vtrjohannesburg	/23	389	380 (97.68%)	250 (67.78%)	102801	102785 (99.98%)	59558 (57.93%)

Table 19: Results for amsterdam01 as the victim while it uses a /23 prefix and mitigates with a /24. The attacker in this scenario using a /24 prefix.

effective: it is unlikely to recover all traffic but can help reduce the damage.

If announcing a longer prefix than the attacker is not feasible, the victim may resort to announcing a prefix of equal length and removing all prepends. In the best case observed in our experiments, *amsterdam01* vs. *neu01*, 14.57% of the monitors would still be hijacked even after this mitigation attempt.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	0	391	57 (14.57%)	100171	19609 (19.57%)
amsterdam01	neu01	3	386	211 (54.66%)	99916	78831 (78.89%)
amsterdam01	ufmg01	0	391	92 (23.52%)	100180	32642 (32.58%)
amsterdam01	ufmg01	3	389	240 (61.69%)	100161	84170 (84.03%)
amsterdam01	vtrseoul	0	392	69 (17.60%)	99904	20379 (20.39%)
amsterdam01	vtrseoul	3	387	261 (67.44%)	100116	88473 (88.37%)
amsterdam01	vtrjohannesburg	0	392	117 (29.84%)	100175	34229 (34.16%)
amsterdam01	vtrjohannesburg	3	392	336 (85.71%)	99904	93529 (93.61%)

Table 20: Prepend results for amsterdam01 as the victim while it uses 0, 1, 2 or 3 prepends.

The last ITE technique that can be employed — or removed as a countermeasure — is selective announcements. As shown in Chapter 6, the choice of which neighbors receive the prefix announcement directly influences the impact of a prefix hijack.

In the case of *amsterdam01*, announcing to both ASes and the IXP resulted in a worse outcome than announcing to a single AS, as shown in Table 21. This indicates that, in some scenarios, an AS can use selective announcements to reduce the impact of a hijack. However, this approach may also increase operational costs, depending on the agreements established with each peer.

Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Peers/IX	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	Coloclue	391	39 (9.97%)	98884	20967 (21.20%)
amsterdam01	neu01	Bit BV	371	65 (17.52%)	98872	21726 (21.97%)
amsterdam01	neu01	Coloclue, Bit BV	386	63 (16.32%)	98794	21681 (21.94%)

Table 21: Result of selective announcement experiments where amsterdam01 is the victim, considering only Bit BV and Coloclue.

Takeaway: Although ITE provides viable mitigation options, our experiments show that when the attacker and victim use the same prefix length, mitigation does not fully

eliminate the impact. Thus, ITE techniques — such as removing prepends or adjusting selective announcements — can help reduce the severity of a hijack but cannot guarantee full recovery. The most effective scenario occurs when the victim can announce a prefix longer than that of the attacker. As such, a network operator should:

- Announce a longer prefix than the attacker, or at least the same length;
- Remove prepends from announcements.

8 IMPACTED ASES

In this chapter, we focus on ASes that accepted the hijack announcements. We define any AS that is neither the victim nor the attacker but selects the hijacker's route as an *impacted AS*. Section 8.1 discusses the propagation time of a hijack within impacted ASes, while Section 8.2 examines how topological distance to the victim and attacker influences the hijack's outcome.

8.1 Attack Propagation

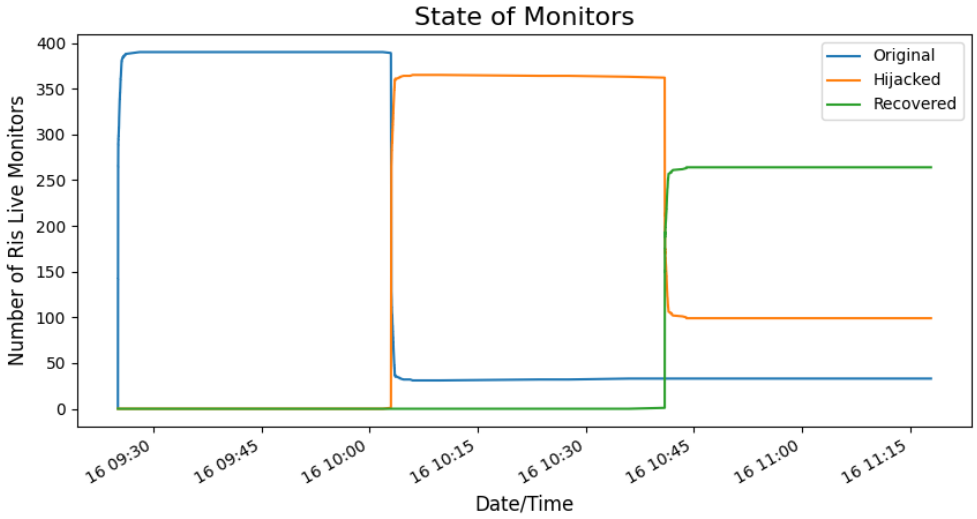
When a BGP announcement occurs, it takes time to propagate across other ASes. This delay not only shows how long a prefix takes to become visible but also defines how quickly a hijack takes effect and how quickly operators must apply mitigation techniques. Understanding this process is essential to assessing how fast impacted ASes divert their traffic from the victim to the attacker.

In our experiments, we allowed announcements to propagate for 15 minutes before starting the data-plane measurements. Most monitors received and responded to announcements within 5 minutes. For instance, in the scenario of *amsterdam01* versus *ufmg01*, shown in Figure 16, both the victim and the attacker announcements reached all RIS Live monitors in around 3 minutes after the annoucement was first made.

Considering that RIS Live includes approximately 400 monitors, it does not capture the behavior of all ASes on the Internet. Nevertheless, it provides valuable insight into how announcements propagate. In this case, any mitigation attempt that seeks to prevent the attacker from spreading the hijack announcement must occur as quickly as possible—ideally within 3 minutes.

Takeaway: Extrapolating this result as the time it takes for an announcement to propagate to all ASes shows that real-time mitigation tactics are required. Mitigation attempts under 5 minutes since the beginning of the attack can also prevent the attacker from acquiring all possible targets.

Figure 16: amsterdam01 vs. ufmg01. In this scenario, the announcements propagate to all monitors in around 3 minutes.



8.2 Distance and Local Preference

An impacted AS may accept a hijack announcement due to its distance from the victim, which can be measured either topologically (number of hops in the AS path) or geographically. In BGP route selection, shorter AS paths are generally preferred. Consequently, an AS that is topologically closer to the victim is less likely to accept the hijacker's announcement than one located farther away.

We observed this behavior in the prepend experiments, where increasing the prepend size caused more monitors to be hijacked. To analyze this effect further, we examine the path lengths of both the victim and the attacker for the monitors that accepted the hijack, in order to determine how many cases can be explained by differences in AS path length.

Experiment Configuration			Hijacker Path Size		
Origin	Hijacker	Prepend Size	Shorter	Equal	Longer
amsterdam01	neu01	0	21	18	18
amsterdam01	neu01	3	200	6	5
amsterdam01	ufmg01	0	35	44	13
amsterdam01	ufmg01	3	233	3	4
amsterdam01	vtrseoul	0	12	41	16
amsterdam01	vtrseoul	3	246	1	14
amsterdam01	vtrjohannesburg	0	42	44	31
amsterdam01	vtrjohannesburg	3	326	0	10
neu01	amsterdam01	0	254	49	45
neu01	amsterdam01	3	340	0	32

Table 22: Comparison between victim announcement AS path size to the hijacker AS path size in the event of a successful hijack during prepend experiments.

In Table 22, we observe that several monitors were hijacked because the attacker offered a shorter AS path than the victim. In the scenario where *neu01* was the victim, most hijacked monitors selected the attacker even without the use of prepending, indicating that the longer AS path of the victim played a decisive role.

It is important to note, however, that some monitors still selected the hijacker's announcement even when the attacker's AS path was longer. This behavior can be explained either by local preference policies applied by ASes along the path. Looking at *neu01* again, we can see that, without using prepends, 45 monitors were affected by the hijack from *amsterdam01* due to local preference, since both the attacker and victim announced prefixes of equal lengths.

Takeaway: Although topological distance does impact the chance of an AS accepting a hijack announcement instead of keeping the route to the victim, we can also observe that local preference in the AS paths will also be a factor that can minimize or amplify the effects of a hijack.

9 CURRENT SCENARIO

Building on the results presented in Chapter 6, we now examine the current state of BGP announcements to assess their potential vulnerability to prefix hijacking. Section 9.1 details the methodology and criteria used in this analysis.

9.1 Methodology

To determine whether a part of the address space is vulnerable to a hijack, we must first define the characteristics to be evaluated.

For the prepend technique, we base our analysis on the best-case scenario observed in our experiments, with *amsterdam01* as the victim, and extrapolate the susceptibility of the address space to hijacks. We adopt the lowest prepend value seen in the RIB. For example, if an AS announces the same prefix with prepends to one neighbor but without prepends to another, we classify the prefix as having no prepends. We classify announcements as follows:

- 0 or 1 prepends are considered *safe*;
- 2 prepends are considered *at risk*;
- 3 or more prepends are considered *not safe*.

For prefix length, we classify a /24 prefix as *safe* and a /23 prefix as *at risk*, since a /24 attack is possible, but disaggregating a /23 requires less effort than attacking shorter prefixes. Prefixes shorter than /23 are classified as *not safe*.

For connectivity, we again use *amsterdam01* as a reference, since it consistently showed the strongest resilience both as a victim and as an attacker in our experiments. With two transit ASes, we consider two neighbors (either providers or peers) as the minimum threshold for being classified as *safe*.

We classify the safety of each announcement based on the worst outcome among the three criteria. For example, an AS that propagates a /24 prefix with strong connectivity will still be marked as *at risk* if it uses 2 prepends. Similarly, we classify a prefix as *not safe* if it is announced as a /20, even when no prepends are used.

To perform this analysis, we disaggregate all prefixes observed in the RIB into /24s. This approach allows us to evaluate what portion of the address space falls into the categories of *safe*, *at risk*, or *not safe*. We base our analysis on a snapshot collected on July 12, 2025, at 12:00:00 UTC, using the RouteViews collector *2nd SAOPAULO* [41].

9.2 Results

First, we analyze the results for each case separately: the use of prepends in the address space, the number of peers or providers, and the prefix length. We then combine these dimensions to present the intersection of results, providing an overview of the current state of the address space.

9.2.1 Prepend Usage

Considering prepend usage in the observed address space, we obtained positive results, as shown in Figure 17a. Considering the snapshot used, the majority of the address space is announced without prepends or with only a single prepend. Only 3.4% of the address space falls into the *at risk* category due to the use of two prepends, while 5.5% is classified as *not safe*.

9.2.2 Peers and Providers

When analyzing the number of peers and providers for each AS originating a prefix, the results indicate that most ASes are on par with, or better than, *amsterdam01*, as shown in Figure 17c. In total, 82.9% of the address space originates from ASes with at least two peers or providers, which we classify as *safe* in this regard.

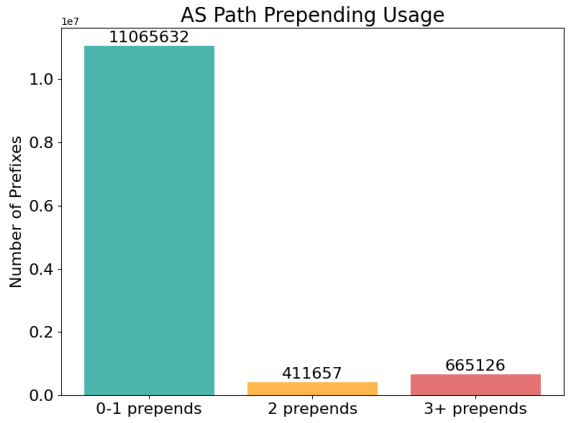
A small fraction of the address space (less than 0.1%) originates from ASes not listed in the ASRank API. We mark these as *No data* and classify them as *not safe*. Furthermore, 17.1% of the address space is announced by ASes with only a single peer or provider, which we also classify as *not safe*.

9.2.3 Prefix Length

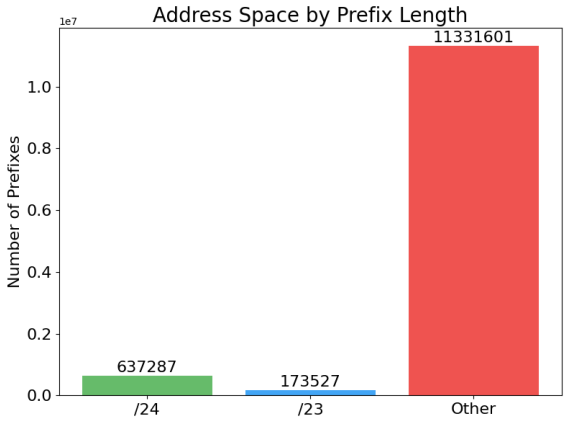
The results for prefix length are less favorable. While we must reiterate the operational challenges of disaggregating prefixes, the analysis shows that most of the address space is not announced as /24, as illustrated in Figure 17b.

Only 5.2% of the address space is covered by /24 announcements and 1.4% by /23. Consequently, the vast majority (93.3%) of the address space remains vulnerable to hijacks using longer, more specific prefixes.

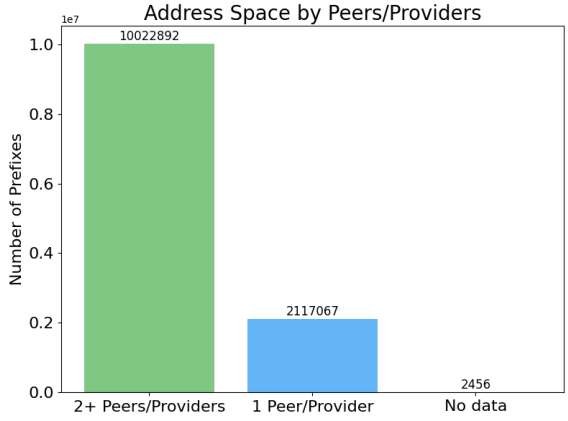
This behavior stands out compared to the other results and highlights a key vulnerability that adversaries could exploit. Although operational costs and practical challenges may render solutions to this issue difficult to implement, hijacks using more specific pre-



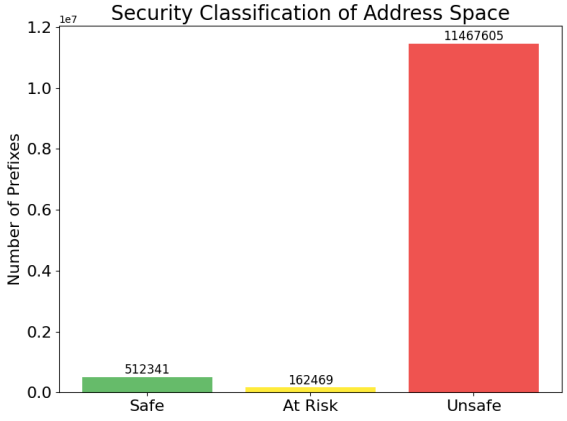
(a) Prepend usage in address space.



(b) Prefix Length in the address space.



(c) Number of Peers/Providers in the address space.



(d) Security of the address space.

Figure 17: amsterdam01 as victim while using selective announcement. neu01 as attacker.

fixes remain the most straightforward vector for malicious networks.

9.2.4 The Address Space Safety

When we intersect the characteristics of the address space, we can classify it into portions that are *safe*, *at risk*, or *not safe*, based on the use of prepends, the number of peers and providers, and the prefix length. Figure 17d shows that the majority of the address space remains vulnerable to prefix hijacks.

This metric is driven primarily by the prefix length results, with 94.4% of the address space classified as *not safe* by at least one criterion. Only 4.2% of the address space can be considered *safe*, given the thresholds defined and the collector used in this study.

10 FINAL REMARKS

In this chapter, we revisit the research questions, link them with the experimental results, state our final considerations, and outline possible directions for future research. We also highlight the contributions achieved during this work and list the related publications authored or co-authored by the researcher.

10.1 Revisiting Research Questions

(RQ1) How do different traffic engineering practices affect the impact of a prefix hijack?

Our experiments show that traffic engineering practices can significantly alter the impact of prefix hijacks. Prepending, in particular, increases the likelihood of hijacks by artificially lengthening AS paths, with more vulnerable ASes being heavily impacted even at small prepend sizes. The use of prefixes longer than the victim's is consistently effective for hijacking. Selective announcements, in turn, demonstrate that security depends not only on the number of neighbors but also on their routing policies. As such, prepend values above two should generally be avoided, and although disaggregation to /24 offers better protection, it introduces operational challenges such as larger routing tables and increased management workload.

(RQ2 and RQ3) Which characteristics of the victim and attacker influence the outcome of a hijack?

Connectivity emerges as the decisive factor. Well-connected ASes, such as *amsterdam01*, can mitigate the effects of prepending and selective announcements, while poorly connected ASes, such as *neu01*, remain vulnerable even without prepends. On the attacker's side, strong connectivity amplifies the impact of hijacks, allowing them to quickly dominate route selection. We also observed that restricting announcements to a carefully chosen subset of neighbors can sometimes improve security, indicating that peering quality and policies of connections matter more than their absolute number.

(RQ4) What leads an AS to accept a hijack announcement?

Most hijacked ASes preferred the attacker's route due to shorter AS paths, especially when the victim employed prepending. However, we also observed hijacks being accepted even when the attacker's path was longer. In these cases, BGP's longest-prefix matching does not apply (since prefix length was equal), which suggests that *local preference policies* drove route selection. This highlights how AS-level policies, beyond basic BGP rules, can both amplify or mitigate the effects of hijacks.

(RQ5) Based on these results, what is the current state of traffic engineering employment on the Internet, and its possible impact on security?

Traffic engineering remains widely used to optimize costs and performance, but it creates non-negligible risks. Our analysis indicates that 94.4% of the IPv4 address space is vulnerable to hijacks, primarily due to the prevalence of prefixes longer than /24 and the effects of prepending. Since hijacks can propagate in under five minutes, mitigation tactics must be applied rapidly to be effective. Operators should weigh the benefits of ITE against these security risks, perhaps applying stricter protection measures only to the most critical prefixes.

10.2 Future Research Directions

This work opens several avenues for further research. First, we plan to expand the experiments to **IPv6** in order to assess whether differences in protocol characteristics and network topology produce distinct results for each traffic-engineering technique. Particular attention must be given to the fact that PEERING muxes may present different connectivity properties for IPv4 and IPv6, which could influence the outcomes.

Second, we intend to enrich the **data-plane analysis** by capturing both incoming and outgoing packets, allowing us to compute round-trip times (RTT). This extension would enable us to evaluate whether hijack events conducted through PEERING introduce measurable latency differences. Moreover, it would allow us to examine correlations between RTT values and the likelihood of targets accepting hijack announcements.

A third line of investigation involves exploring the role of **RPKI adoption and validation**. By manipulating PEERING Route Origin Authorizations (ROAs), we can design announcements that should be protected by RPKI and evaluate to what extent current deployments reduce hijack impact. Such experiments would provide empirical evidence of RPKI's effectiveness in mitigating real-world attacks.

Finally, we aim to consolidate and extend the analyses presented here with the goal of submitting the results to major venues in 2025, such as the *Passive and Active Measurement Conference (PAM)* or the *Internet Measurement Conference (IMC)*.

Contributions

This research produced several contributions that advance the understanding of traffic engineering and hijack dynamics:

- **PEERING infrastructure improvements:** Our experiments helped identify and solve issues with the Vultr muxes, revealed potential difficulties in using Vultr BGP communities through PEERING, and uncovered an anomaly in the *ufmg01* mux. These findings enable future research to progress without facing the same limitations.

- **Knowledge transfer to the community:** We contributed to capacity building by delivering an online presentation hosted by the Brazilian Network Information Centre (NIC.br), demonstrating how to use PEERING and RIS Live to assess the impact of Internet Traffic Engineering (ITE) on hijack events [26].

- **Theoretical impact analysis:** By investigating the theoretical impact of prepend experiments, we observed networks that, according to RIS Live monitors, should not have been affected but nevertheless accepted the hijack. This result highlights the limitations of RIS Live's visibility and reinforces the importance of integrating additional data sources to achieve broader and more reliable coverage.

Publications by the Author

The following is a chronologically ordered list of papers published by the author, or with contributions from the author, throughout the course of the Master's Program:

- Barreto, R. P., Bertholdo, L. M., and Marcos, P. d. B. (2024b). Poster: Traffic engineering security implications. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 771–772, New York, NY, USA. Association for Computing Machinery

- Bertholdo, L. M., Barreto, R. P., and Marcos, P. d. B. (2024). Poster: Building comprehensive telecommunications datasets during a major climatic event. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 781–782, New York, NY, USA. Association for Computing Machinery

- Bertholdo, L. M., Paredes, R. B., de Lima Marin, G., Loureiro, C. A. H., Kashiwakura, M. K., and de Botelho Marcos, P. (2025). Analyzing the effect of an extreme weather event on telecommunications and information technology: Insights from 30 days of flooding. In *Passive and Active Measurement: 26th International Conference, PAM 2025, Virtual Event, March 10–12, 2025, Proceedings*, page 276–304, Berlin, Heidelberg. Springer-Verlag

- Barreto, R., Bertholdo, L., and Marcos, P. (2024a). Investigating the security implications of traffic engineering and connectivity in internet routing. In *Proceedings of the 21st Regional School of Computer Networks*, pages 35–41, Porto Alegre, RS, Brasil. SBC

REFERENCES

[1] Azimov, A., Bogomazov, E., Bush, R., Patel, K., Snijders, J., and Sriram, K. (2024). BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects. Internet-Draft draft-ietf-sidrops-aspa-verification-18, Internet Engineering Task Force. Work in Progress.

[2] Ballani, H., Francis, P., and Zhang, X. (2007). A study of prefix hijacking and interception in the internet. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, page 265–276, New York, NY, USA. Association for Computing Machinery.

[3] Barreto, R., Bertholdo, L., and Marcos, P. (2024a). Investigating the security implications of traffic engineering and connectivity in internet routing. In *Proceedings of the 21st Regional School of Computer Networks*, pages 35–41, Porto Alegre, RS, Brasil. SBC.

[4] Barreto, R. P., Bertholdo, L. M., and Marcos, P. d. B. (2024b). Poster: Traffic engineering security implications. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 771–772, New York, NY, USA. Association for Computing Machinery.

[5] Bertholdo, L. M., Barreto, R. P., and Marcos, P. d. B. (2024). Poster: Building comprehensive telecommunications datasets during a major climatic event. In *Proceedings of the 2024 ACM on Internet Measurement Conference*, IMC '24, page 781–782, New York, NY, USA. Association for Computing Machinery.

[6] Bertholdo, L. M., Paredes, R. B., de Lima Marin, G., Loureiro, C. A. H., Kashiwakura, M. K., and de Botelho Marcos, P. (2025). Analyzing the effect of an extreme weather event on telecommunications and information technology: Insights from 30 days of flooding. In *Passive and Active Measurement: 26th International Conference, PAM 2025, Virtual Event, March 10–12, 2025, Proceedings*, page 276–304, Berlin, Heidelberg. Springer-Verlag.

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

[8] Caida (2025). As rank. https://asrank.caida.org/.

[9] Chang, R. and Lo, M. (2005). Inbound traffic engineering for multihomed ass using as path prepending. *IEEE Network*, 19(2):18–25.

[10] Cho, S., Fontugne, R., Cho, K., Dainotti, A., and Gill, P. (2019). Bgp hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*, pages 25–32.

[11] Chung, T., Aben, E., Bruijnzeels, T., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., Rijswijk-Deij, R. v., Rula, J., and Sullivan, N. (2019). Rpki is coming of age: A longitudinal study of rpki deployment and invalid route origins. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 406–419, New York, NY, USA. Association for Computing Machinery.

[12] de Botelho Marcos, P., Prehn, L., Leal, L., Dainotti, A., Feldmann, A., and Barcellos, M. (2020). As-path prepending: There is no rose without a thorn. In *ACM IMC 2020*.

[13] Fan, X. and Heidemann, J. (2010). Selecting representative IP addresses for Internet topology studies. In *ACM IMC 2010*, IMC '10. ACM.

[14] Fanou, R., Huffaker, B., Mok, R., and Claffy, K. C. (2020). Unintended consequences: Effects of submarine cable deployment on internet routing. In Sperotto, A., Dainotti, A., and Stiller, B., editors, *Passive and Active Measurement*, pages 211–227, Cham. Springer International Publishing.

[15] Feamster, N., Borkenhagen, J., and Rexford, J. (2003). Guidelines for interdomain traffic engineering. *SIGCOMM Comput. Commun. Rev.*, 33(5):19–30.

[16] Gao, L. and Rexford, J. (2000). Stable internet routing without global coordination. *SIGMETRICS Perform. Eval. Rev.*, 28(1):307–317.

[17] Garcia, L. M. and Fyodor (2024). Nmap. https://nmap.org/.

[18] Gill, P., Schapira, M., and Goldberg, S. (2014). A survey of interdomain routing policies. *SIGCOMM Comput. Commun. Rev.*, 44(1):28–34.

[19] Group, T. T. (2024). Tcpdump and libpcap. https://www.tcpdump.org/.

[20] Hlavacek, T., Shulman, H., Vogel, N., and Waidner, M. (2023). Keep your friends close, but your routeservers closer: Insights into RPKI validation in the internet. In

32nd USENIX Security Symposium (USENIX Security 23), pages 4841–4858, Anaheim, CA. USENIX Association.

[21] Holterbach, T., Alfroy, T., Phokeer, A. D., Dainotti, A., and Pelsser, C. (2024). A system to detect forged-origin hijacks. In *21th USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. USENIX Association.

[22] Kastanakis, S., Giotsas, V., Livadariu, I., and Suri, N. (2023). Replication: 20 years of inferring interdomain routing policies. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, IMC '23, page 16–29, New York, NY, USA. Association for Computing Machinery.

[23] Koch, T., Yu, S., Agarwal, S., Katz-Bassett, E., and Beckett, R. (2023). Painter: Ingress traffic engineering and routing for enterprise cloud networks. In *Proceedings of the ACM SIGCOMM 2023 Conference*, ACM SIGCOMM '23, page 360–377, New York, NY, USA. Association for Computing Machinery.

[24] Kuhn, D., Sriram, K., and Montgomery, D. (2007). Sp 800-54. border gateway protocol security.

[25] Lepinski, M. and Sriram, K. (2017). BGPsec Protocol Specification. RFC 8205.

[26] Marcos, P. and Barreto, R. (2025). [capweek 10] prefix hijacks: How to defend and how traffic engineering can help? https://www.youtube.com/watch?v=BgJ3ubxMxZE. [Online; accessed 25-July-2025].

[27] Milolidakis, A., Bühler, T., Wang, K., Chiesa, M., Vanbever, L., and Vissicchio, S. (2023). On the effectiveness of bgp hijackers that evade public route collectors. *IEEE Access*, 11:31092–31124.

[28] NCC, R. (2022). Route collectors. https://ris.ripe.net/docs/10_routecollectors.html. [Online; accessed 12-June-2022].

[29] NCC, R. (2024). Ris live. https://ris-live.ripe.net/.

[30] Oesterle, J. (2021). Challenges with BGPSec. In Carle, G., Günther, S., and Jaeger, B., editors, *Proceedings of the Seminar Innovative Internet Technologies and Mobile Communications (IITM), Summer Semester 2021*, volume NET-2022-01-1 of *Network Architectures and Services (NET)*, pages 5–9, Munich, Germany. Chair of Network Architectures and Services, Department of Computer Science, Technical University of Munich.

[31] Oliver, L., Akiwate, G., Luckie, M., Du, B., and claffy, k. (2022). Stop, drop, and roa: Effectiveness of defenses through the lens of drop. In *Proceedings of the*

22nd ACM Internet Measurement Conference, IMC '22, page 730–737, New York, NY, USA. Association for Computing Machinery.

[32] Rekhter, Y., Hares, S., and Li, T. (2006a). A Border Gateway Protocol 4 (BGP-4). RFC 4271.

[33] Rekhter, Y., Hares, S., and Li, T. (2006b). A Border Gateway Protocol 4 (BGP-4). RFC 4271.

[34] Rizvi, A., Bertholdo, L., Ceron, J., and Heidemann, J. (2022). Anycast agility: Network playbooks to fight {DDoS}. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4201–4218.

[35] Schlinker, B., Arnold, T., Cunha, I., and Katz-Bassett, E. (2019). Peering: virtualizing bgp at the edge for research. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, page 51–67, New York, NY, USA. Association for Computing Machinery.

[36] Sermpezis, P., Kotronis, V., Dainotti, A., and Dimitropoulos, X. (2018). A survey among network operators on bgp prefix hijacking. *SIGCOMM Comput. Commun. Rev.*, 48(1):64–69.

[37] Siddiqui, A. (2022). Klayswap – another bgp hijack targeting crypto wallets. https://manrs.org/2022/02/klayswap-another-bgp-hijack-targeting-crypto-wallets/.

[38] Testart, C., Richter, P., King, A., Dainotti, A., and Clark, D. (2019). Profiling bgp serial hijackers: Capturing persistent misbehavior in the global routing table. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 420–434, New York, NY, USA. Association for Computing Machinery.

[39] Testart, C., Richter, P., King, A., Dainotti, A., and Clark, D. (2020). To Filter or not to Filter: Measuring the Benefits of Registering in the RPKI Today. In *Passive and Active Measurement Conference (PAM)*.

[40] Umeda, N., Kimura, T., and Yanai, N. (2023). The juice is worth the squeeze: Analysis of autonomous system provider authorization in partial deployment. *IEEE Open Journal of the Communications Society*, 4:269–306.

[41] University of Oregon (2022). University of oregon route views archive project. http://archive.routeviews.org/. [Online; accessed 12-June-2022].

[42] van Hove, K., van der Ham-de Vos, J., and van Rijswijk-Deij, R. (2023). Rpkiller: Threat analysis of the bgp resource public key infrastructure. *Digital Threats*, 4(4).

A PREPEND TABLES

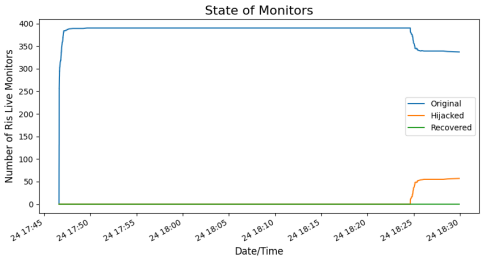
Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	0	391	57 (14.57%)	100171	19609 (19.57%)
amsterdam01	neu01	1	390	128 (32.82%)	99878	50246 (50.30%)
amsterdam01	neu01	2	393	186 (47.32%)	99918	71538 (71.59%)
amsterdam01	neu01	3	386	211 (54.66%)	99916	78831 (78.89%)
amsterdam01	ufmg01	0	391	92 (23.52%)	100180	32642 (32.58%)
amsterdam01	ufmg01	1	390	161 (41.28%)	100007	60404 (60.39%)
amsterdam01	ufmg01	2	388	219 (56.44%)	99752	80739 (80.93%)
amsterdam01	ufmg01	3	389	240 (61.69%)	100161	84170 (84.03%)
amsterdam01	vtrseoul	0	392	69 (17.60%)	99904	20379 (20.39%)
amsterdam01	vtrseoul	1	389	166 (42.67%)	99875	57586 (57.65%)
amsterdam01	vtrseoul	2	390	243 (62.30%)	100036	85256 (85.22%)
amsterdam01	vtrseoul	3	387	261 (67.44%)	100116	88473 (88.37%)
amsterdam01	vtrjohannesburg	0	392	117 (29.84%)	100175	34229 (34.16%)
amsterdam01	vtrjohannesburg	1	390	240 (61.53%)	99827	64593 (64.70%)
amsterdam01	vtrjohannesburg	2	378	318 (84.12%)	99876	89523 (89.63%))
amsterdam01	vtrjohannesburg	3	392	336 (85.71%)	99904	93529 (93.61%)
neu01	amsterdam01	0	363	348 (95.86%)	99786	72218 (72.37%)
neu01	amsterdam01	1	363	374 (103.03%)	99573	87131 (87.50%))
neu01	amsterdam01	2	361	371 (102.77%)	99536	88874 (89.28%)
neu01	amsterdam01	3	360	372 (103.33%)	99629	89047 (89.37%)
neu01	ufmg01	0	363	187 (51.51%)	99690	52711 (52.87%)
neu01	ufmg01	1	361	350 (96.95%)	99507	94490 (94.95%)
neu01	ufmg01	2	361	349 (96.67%)	99698	94723 (95.00%)
neu01	ufmg01	3	358	349 (97.48%)	99583	94609 (95.00%)
neu01	vtrseoul	0	363	237 (65.28%)	99640	48552 (48.72%)
neu01	vtrseoul	1	361	314 (86.98%)	99656	81614 (81.89%)
neu01	vtrseoul	2	360	348 (96.66%)	99599	88606 (88.96%)
neu01	vtrseoul	3	361	351 (97.22%)	99636	89293 (89.61%)
neu01	vtrjohannesburg	0	362	273 (75.41%)	99475	55170 (55.46%)
neu01	vtrjohannesburg	1	361	366 (101.38%)	99487	94468 (94.95%)
neu01	vtrjohannesburg	2	361	366 (101.38%)	99384	94397 (94.98%)
neu01	vtrjohannesburg	3	360	367 (101.94%)	83721	78716 (94.02%)
ufmg01	neu01	0	362	157 (43.37%)	99733	39768 (39.87%)
ufmg01	neu01	1	364	285 (78.29%)	99772	79586 (79.76%)
ufmg01	neu01	2	363	315 (86.77%)	99793	86669 (86.84%)
ufmg01	neu01	3	364	319 (87.63%)	99816	88291 (88.45%)
ufmg01	amsterdam01	0	362	310 (85.63%)	99835	60619 (60.71%)
ufmg01	amsterdam01	1	363	349 (96.14%)	99752	81572 (81.77%)
ufmg01	amsterdam01	2	364	361 (99.17%)	99782	83813 (83.99%)
ufmg01	amsterdam01	3	363	361 (99.44%)	99767	84605 (84.80%)
ufmg01	vtrseoul	0	362	152 (41.98%)	99834	32995 (33.04%)
ufmg01	vtrseoul	1	364	291 (79.94%)	99859	55684 (55.76%)
ufmg01	vtrseoul	2	363	326 (89.80%)	99736	86501 (86.72%)
ufmg01	vtrseoul	3	363	327 (90.08%)	99791	82174 (82.34%)
ufmg01	vtrjohannesburg	0	362	212 (58.56%)	99580	26832 (26.94%)
ufmg01	vtrjohannesburg	1	364	338 (92.85%)	99687	62121 (62.31%)
ufmg01	vtrjohannesburg	2	363	346 (95.31%)	99587	88632 (88.99%)
ufmg01	vtrjohannesburg	3	362	351 (96.96%)	99665	89611 (89.91%)

Table 23: Results in the data and control plane using 0, 1, 2 or 3 prepends

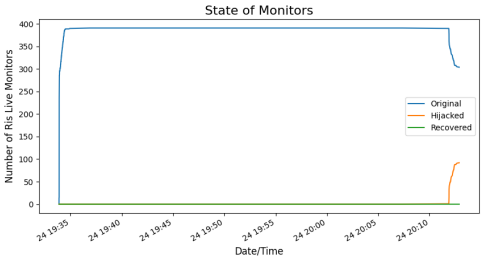
Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Prepend Size	Total	Hijacked	Total	Hijacked
vtrseoul	neu01	0	364	117 (32.14%)	99993	37739 (37.74%)
vtrseoul	neu01	1	364	222 (60.98%)	96658	69131 (71.52%)
vtrseoul	neu01	2	365	270 (73.97%)	99973	83221 (83.24%)
vtrseoul	neu01	3	363	271 (74.65%)	99887	83978 (84.07%)
vtrseoul	ufmg01	0	363	209 (57.57%)	100102	61962 (61.89%)
vtrseoul	ufmg01	1	364	269 (73.90%)	99914	82363 (82.43%)
vtrseoul	ufmg01	2	365	288 (78.90%)	99913	85316 (85.39%)
vtrseoul	ufmg01	3	363	284 (78.23%)	99929	85521 (85.58%)
vtrseoul	amsterdam01	0	364	335 (92.03%)	100173	74981 (74.85%)
vtrseoul	amsterdam01	1	363	353 (97.24%)	100043	89601 (89.56%)
vtrseoul	amsterdam01	2	365	363 (99.45%)	99967	90143 (90.17%)
vtrseoul	amsterdam01	3	362	359 (99.17%)	99813	90074 (90.24%)
vtrseoul	vtrjohannesburg	0	363	241 (66.39%)	99608	56957 (57.18%)
vtrseoul	vtrjohannesburg	1	364	338 (92.85%)	99777	89763 (89.96%)
vtrseoul	vtrjohannesburg	2	365	350 (95.89%)	99831	92276 (92.43%)
vtrseoul	vtrjohannesburg	3	363	348 (95.86%)	99624	92318 (92.66%)
vtrjohannesburg	neu01	0	377	107 (28.38%)	99941	41520 (41.54%)
vtrjohannesburg	neu01	1	379	173 (45.64%)	99840	63519 (63.62%)
vtrjohannesburg	neu01	2	378	200 (52.91%)	99885	69876 (69.95%)
vtrjohannesburg	neu01	3	379	204 (53.82%)	99604	70769 (71.05%)
vtrjohannesburg	ufmg01	0	378	153 (40.47%)	99985	54377 (54.38%)
vtrjohannesburg	ufmg01	1	379	217 (57.25%)	99806	71531 (71.67%)
vtrjohannesburg	ufmg01	2	380	236 (62.10%)	99894	74927 (75.00%)
vtrjohannesburg	ufmg01	3	380	241 (63.42%)	99834	84834 (84.97%)
vtrjohannesburg	vtrseoul	0	379	118 (31.13%)	99996	33581 (33.58%)
vtrjohannesburg	vtrseoul	1	379	196 (51.71%)	99912	60838 (60.89%)
vtrjohannesburg	vtrseoul	2	378	249 (65.87%)	99884	68802 (68.88%)
vtrjohannesburg	vtrseoul	3	378	252 (66.66%)	99879	71159 (71.24%)
vtrjohannesburg	amsterdam01	0	378	312 (82.53%)	99974	59130 (59.14%)
vtrjohannesburg	amsterdam01	1	379	347 (91.55%)	99803	72787 (72.93%)
vtrjohannesburg	amsterdam01	2	379	349 (92.08%)	99815	74598 (74.73%)
vtrjohannesburg	amsterdam01	3	379	351 (92.61%)	99205	74711 (75.30%)

Table 24: Results in the data and control plane using 0, 1, 2 or 3 prepends.

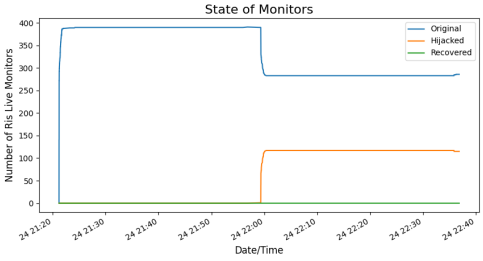
B PREPEND GRAPHS



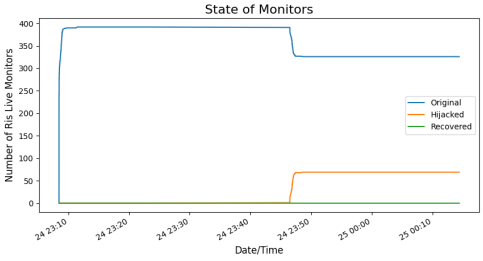
(a) vs. neu01 - prepend 0



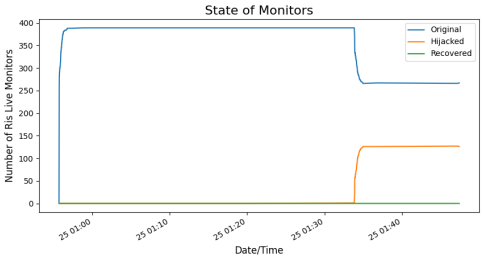
(b) vs. ufmg01 - prepend 0



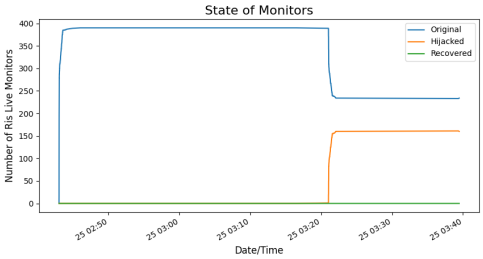
(c) vs. vtrjohannesburg - prepend 0



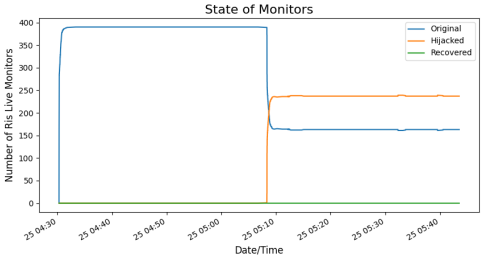
(d) vs. vtrseoul - prepend 0



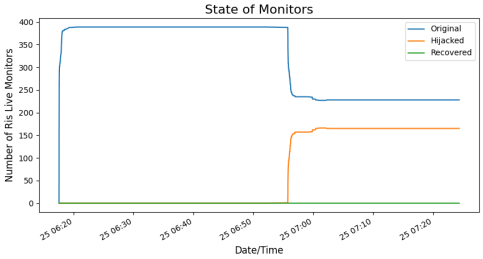
(e) vs. neu01 - prepend 1



(f) vs. ufmg01 - prepend 1

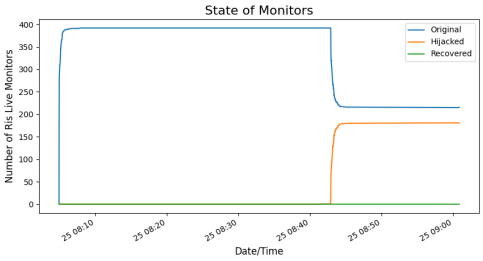


(g) vs. vtrjohannesburg - prepend 1

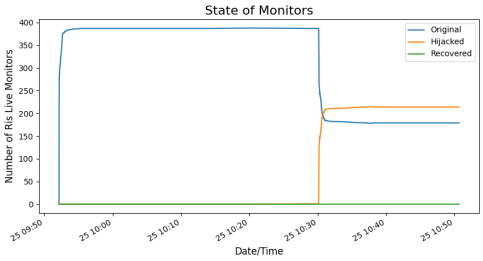


(h) vs. vtrseoul - prepend 1

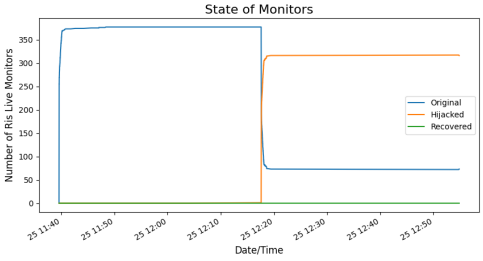
Figure 18: amsterdam01 as victim with 0 and 1 prepends.



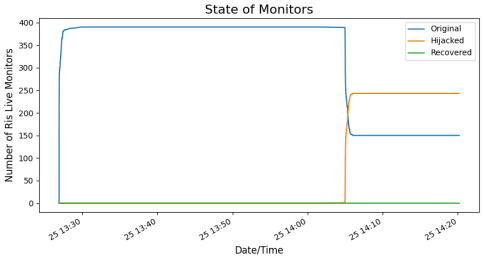
(a) vs. neu01 - prepend 2



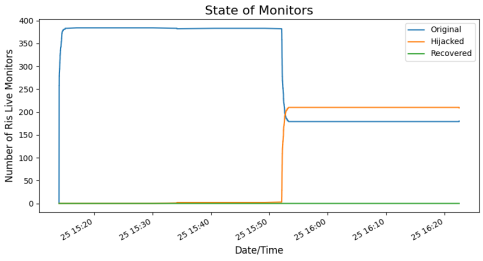
(b) vs. ufmg01 - prepend 2



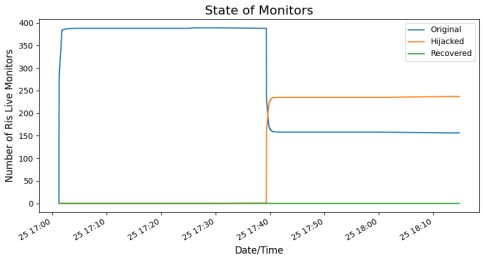
(c) vs. vtrjohannesburg - prepend 2



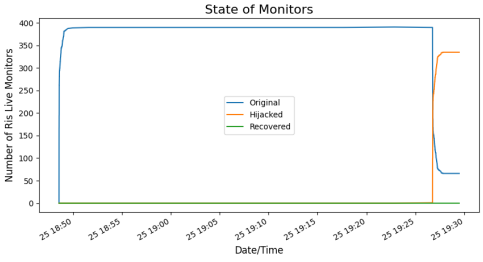
(d) vs. vtrseoul - prepend 2



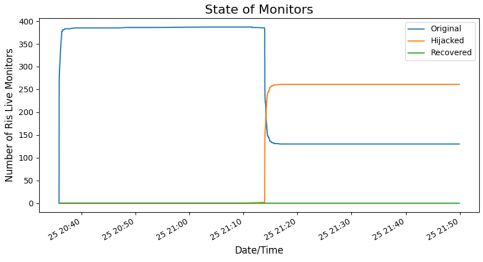
(e) vs. neu01 - prepend 3



(f) vs. ufmg01 - prepend 3

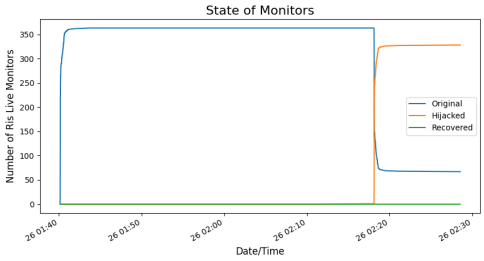


(g) vs. vtrjohannesburg - prepend 3

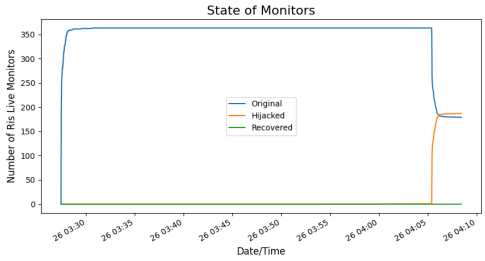


(h) vs. vtrseoul - prepend 3

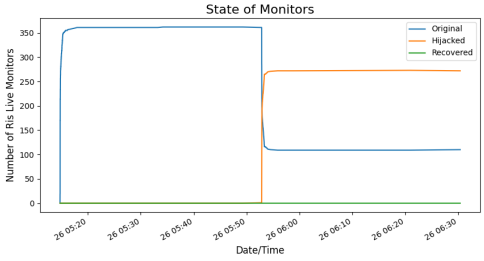
Figure 19: amsterdam01 as victim with 2 and 3 prepends.



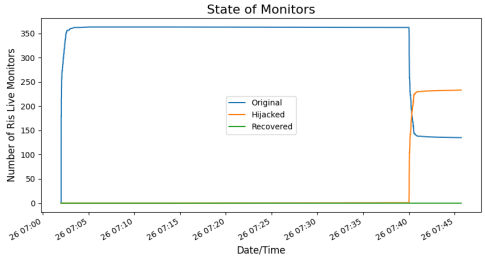
(a) vs. amsterdam01 - prepend 0



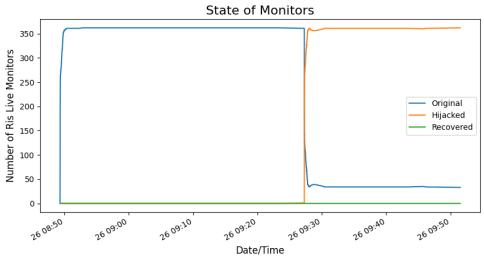
(b) vs. ufmg01 - prepend 0



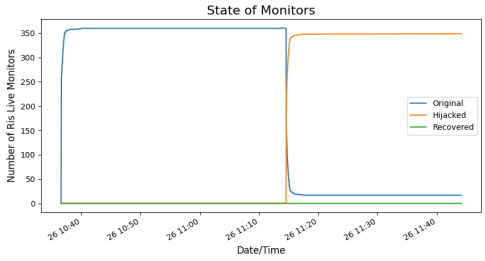
(c) vs. vtrjohannesburg - prepend 0



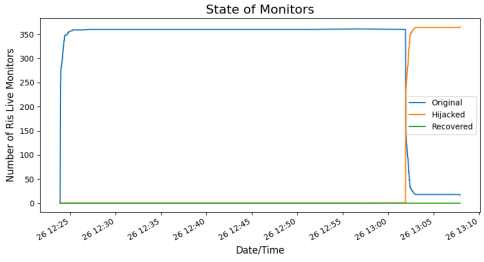
(d) vs. vtrseoul - prepend 0



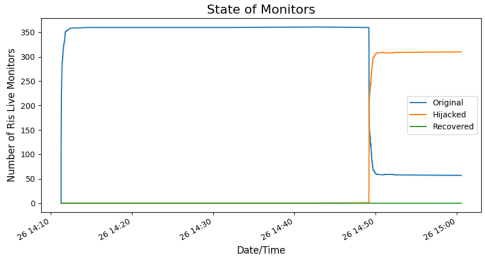
(e) vs. amsterdam01 - prepend 1



(f) vs. ufmg01 - prepend 1

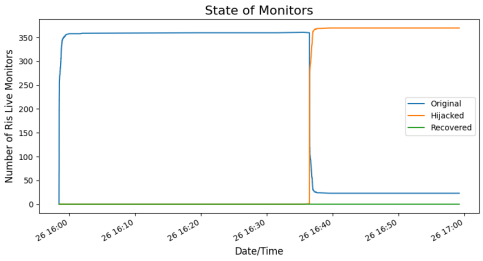


(g) vs. vtrjohannesburg - prepend 1

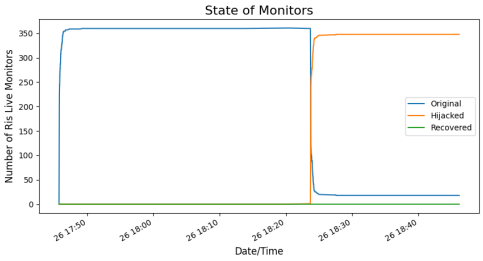


(h) vs. vtrseoul - prepend 1

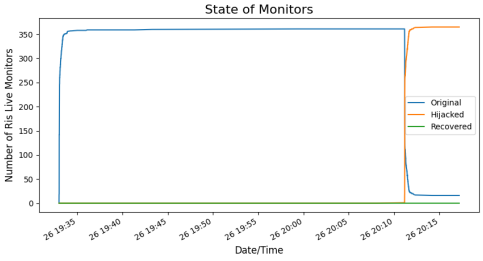
Figure 20: neu01 as victim with 0 and 1 prepends.



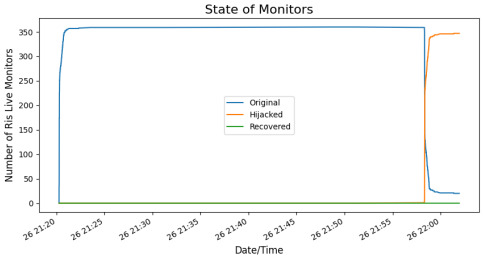
(a) vs. amsterdam01 - prepend 2



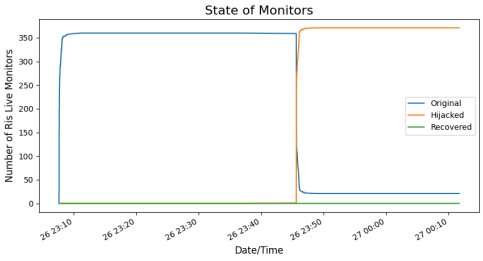
(b) vs. ufmg01 - prepend 2



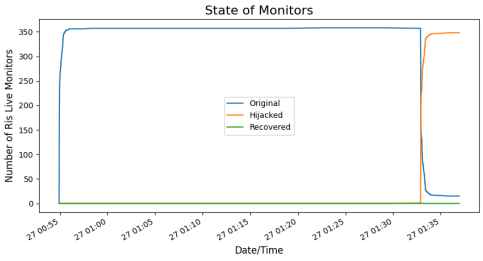
(c) vs. vtrjohannesburg - prepend 2



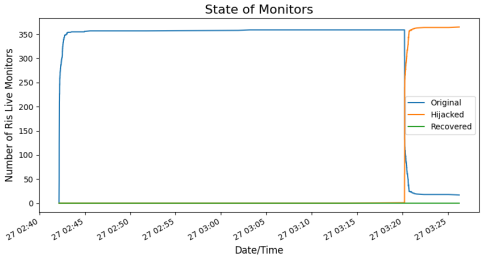
(d) vs. vtrseoul - prepend 2



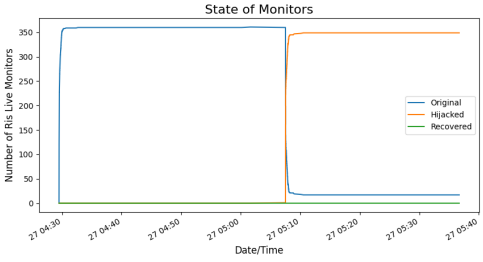
(e) vs. amsterdam01 - prepend 3



(f) vs. ufmg01 - prepend 3

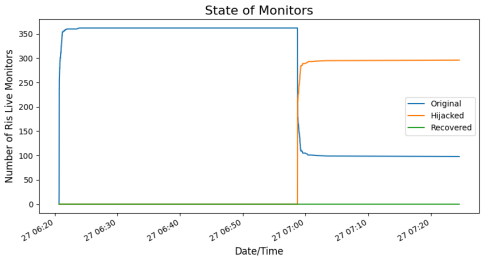


(g) vs. vtrjohannesburg - prepend 3

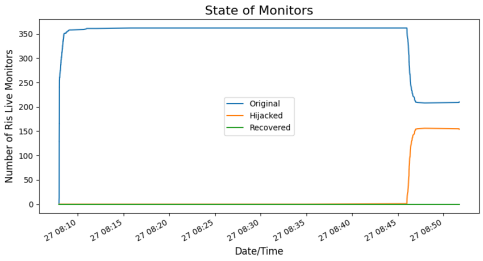


(h) vs. vtrseoul - prepend 3

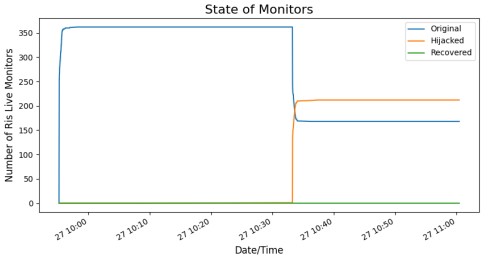
Figure 21: neu01 as victim with 2 and 3 prepends.



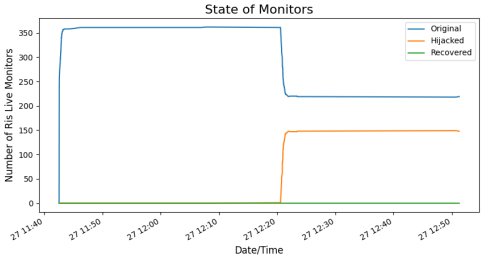
(a) vs. amsterdam01 - prepend 0



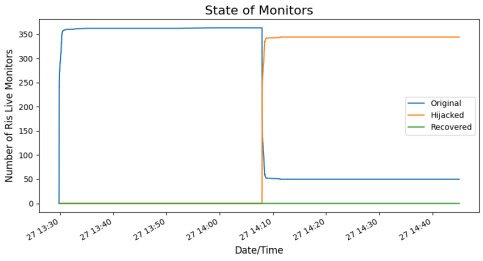
(b) vs. neu01 - prepend 0



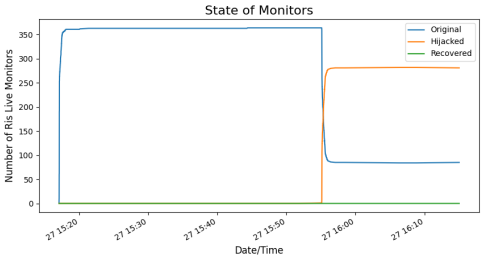
(c) vs. vtrjohannesburg - prepend 0



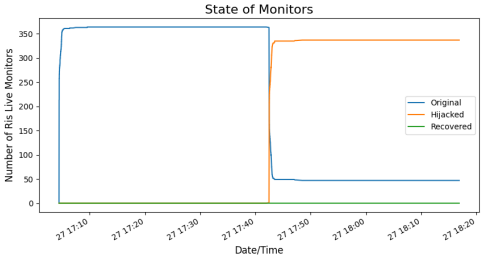
(d) vs. vtrseoul - prepend 0



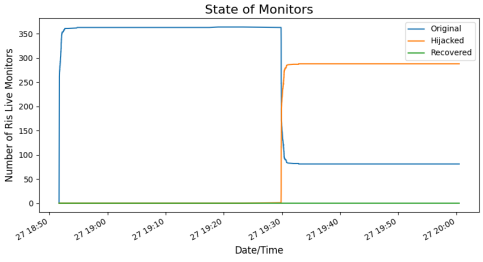
(e) vs. amsterdam01 - prepend 1



(f) vs. neu01 - prepend 1

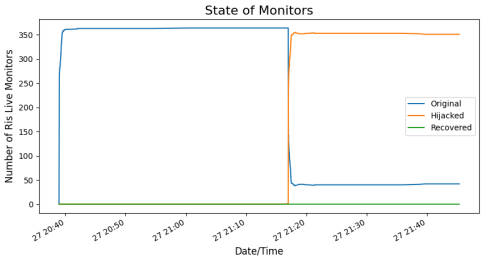


(g) vs. vtrjohannesburg - prepend 1

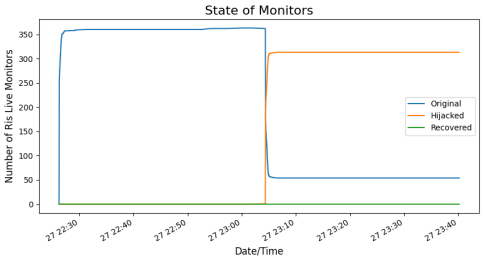


(h) vs. vtrseoul - prepend 1

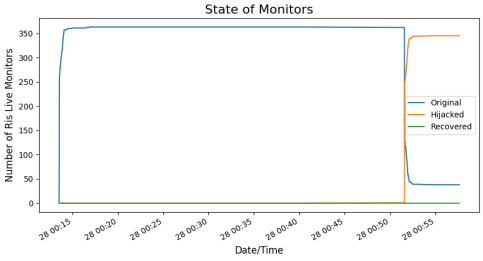
Figure 22: ufmg01 as victim with 0 and 1 prepends.



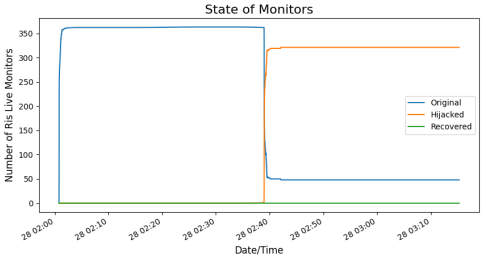
(a) vs. amsterdam01 - prepend 3



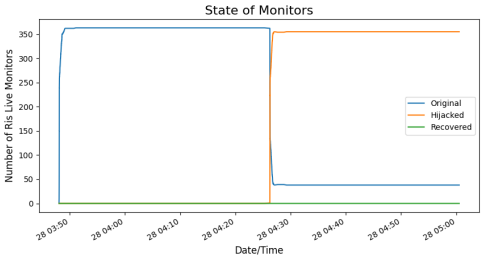
(b) vs. neu01 - prepend 3



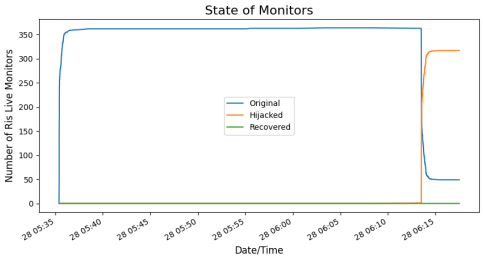
(c) vs. vtrjohannesburg - prepend 3



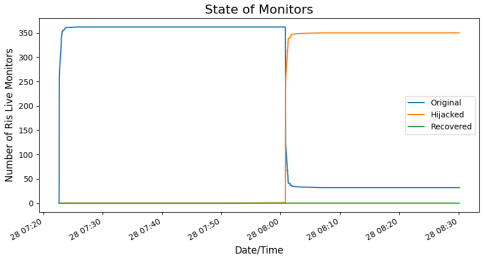
(d) vs. vtrseoul - prepend 3



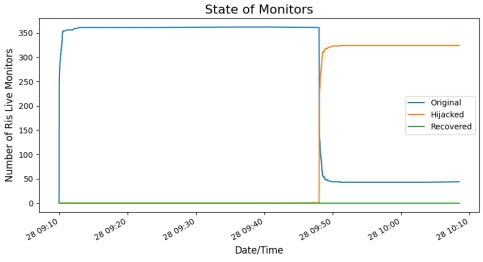
(e) vs. amsterdam01 - prepend 4



(f) vs. neu01 - prepend 4

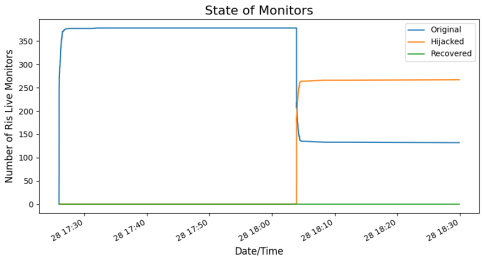


(g) vs. vtrjohannesburg - prepend 4

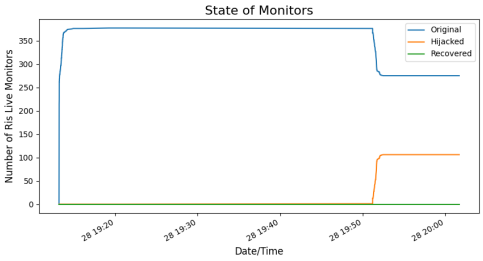


(h) vs. vtrseoul - prepend 4

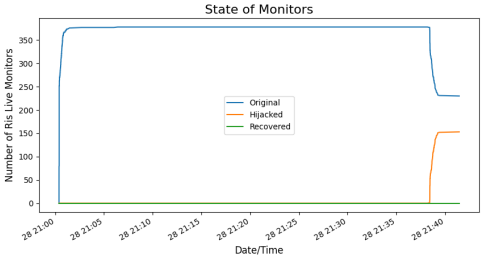
Figure 23: ufgm01 as victim with 2 and 3 prepends.



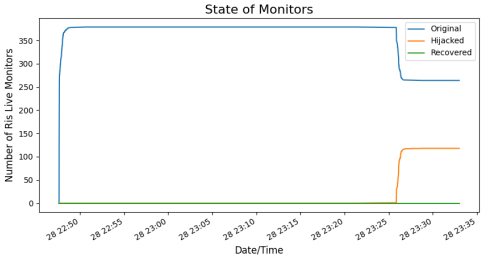
(a) vs. amsterdam01 - prepend 0



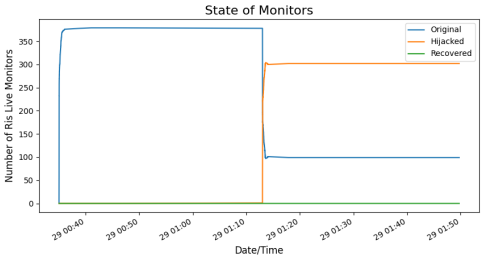
(b) vs. neu01 - prepend 0



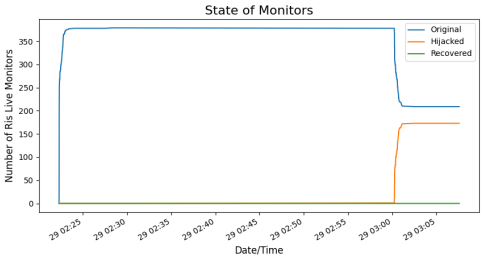
(c) vs. ufmg01 - prepend 0



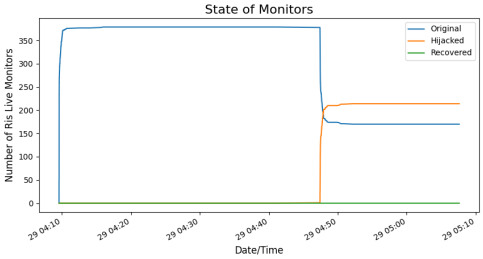
(d) vs. vtrseoul - prepend 0



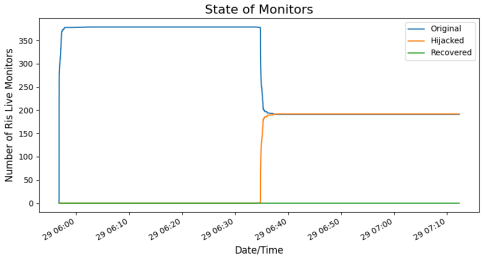
(e) vs. amsterdam01 - prepend 1



(f) vs. neu01 - prepend 1

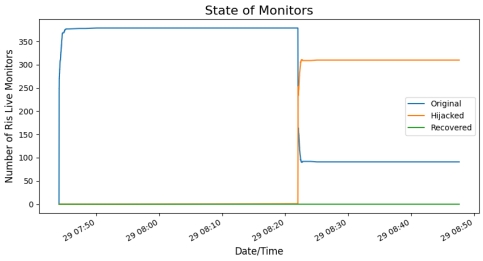


(g) vs. ufmg01 - prepend 1

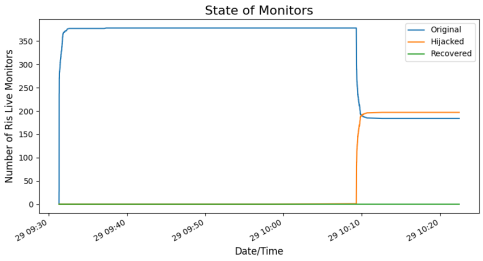


(h) vs. vtrseoul - prepend 1

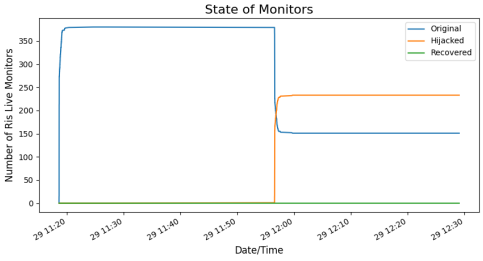
Figure 24: vtrjohannesburg as victim with 0 and 1 prepends.



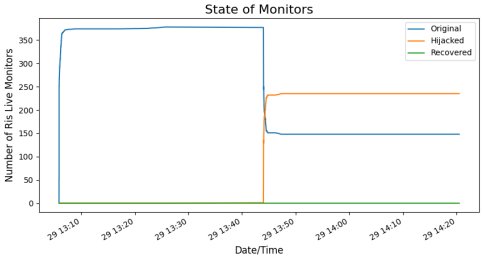
(a) vs. amsterdam01 - prepend 2



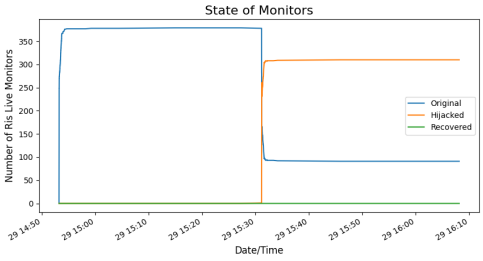
(b) vs. neu01 - prepend 2



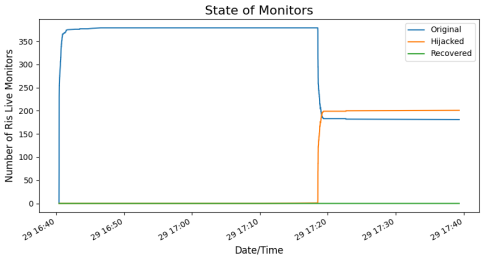
(c) vs. ufmg01 - prepend 2



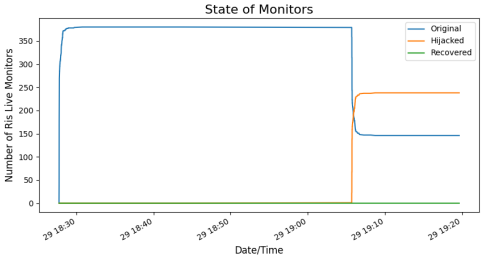
(d) vs. vtrseoul - prepend 2



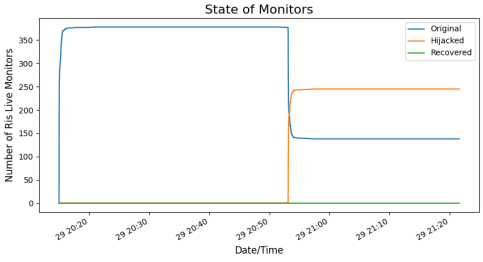
(e) vs. amsterdam01 - prepend 3



(f) vs. neu01 - prepend 3

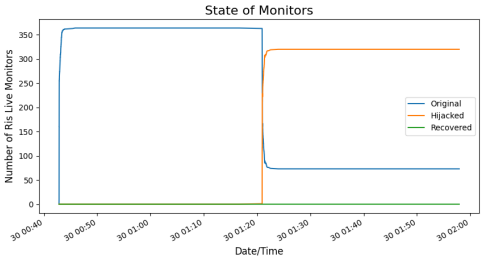


(g) vs. ufmg01 - prepend 3

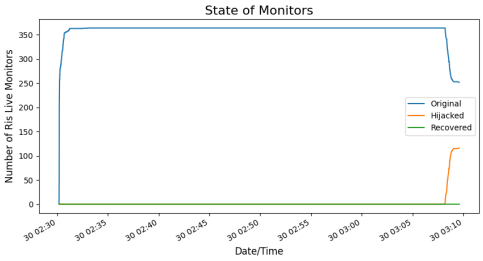


(h) vs. vtrseoul - prepend 3

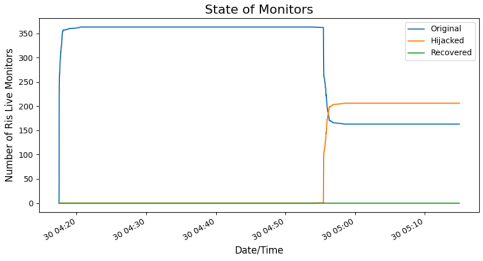
Figure 25: vtrjohannesburg as victim with 2 and 3 prepends.



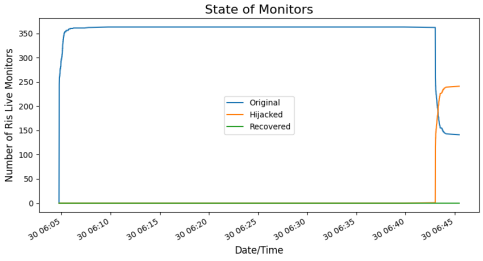
(a) vs. amsterdam01 - prepend 0



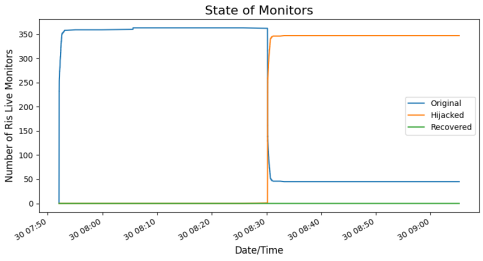
(b) vs. neu01 - prepend 0



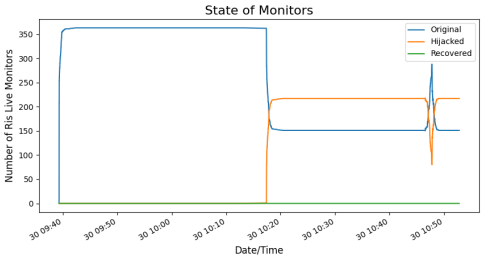
(c) vs. ufmg01 - prepend 0



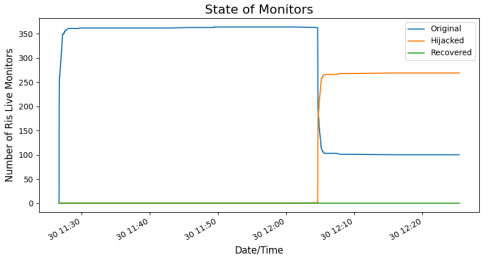
(d) vs. vtrjohannesburg - prepend 0



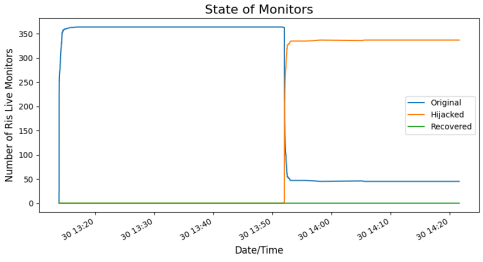
(e) vs. amsterdam01 - prepend 1



(f) vs. neu01 - prepend 1

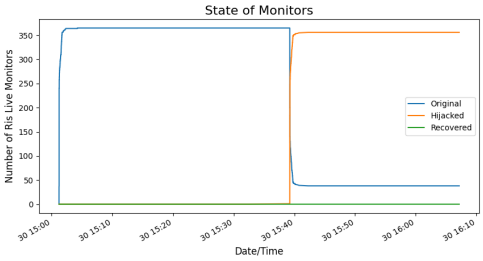


(g) vs. ufmg01 - prepend 1

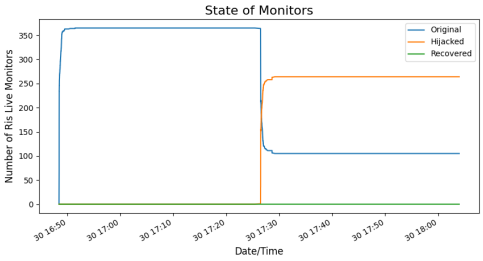


(h) vs. vtrjohannesburg - prepend 1

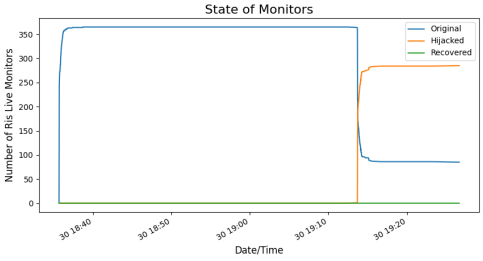
Figure 26: vtrseoul as victim with 0 and 1 prepends.



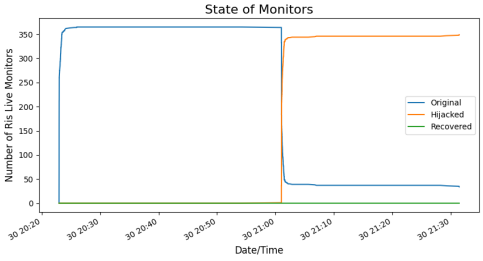
(a) vs. amsterdam01 - prepend 2



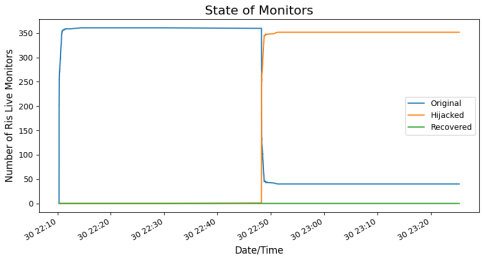
(b) vs. neu01 - prepend 2



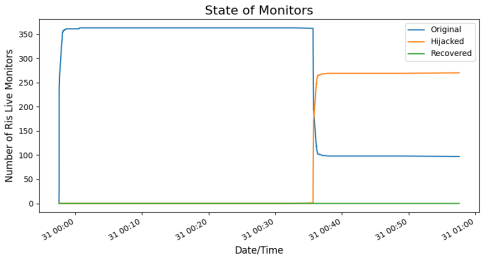
(c) vs. ufmg01 - prepend 2



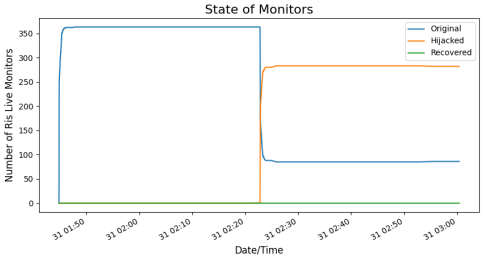
(d) vs. vtrjohannesburg - prepend 2



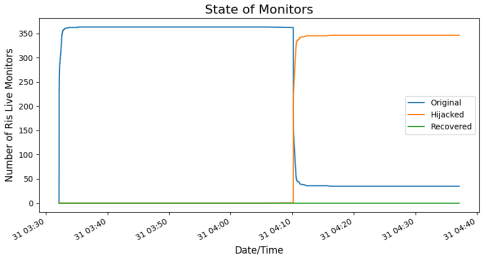
(e) vs. amsterdam01 - prepend 3



(f) vs. neu01 - prepend 3



(g) vs. ufmg01 - prepend 3



(h) vs. vtrjohannesburg - prepend 3

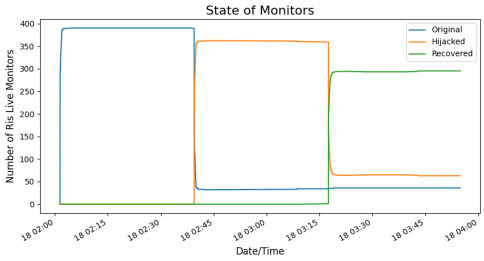
Figure 27: vtrseoul as victim with 2 and 3 prepends.

C PREFIX LENGTH TABLE

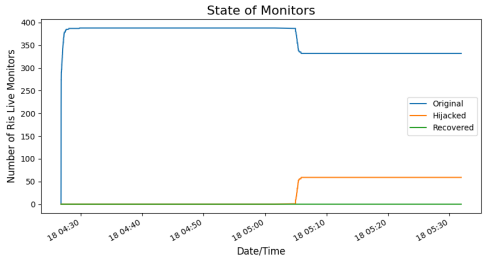
Experiment Configuration			Control Plane Monitors			Data Plane Targets		
Origin	Hijacker	victim Prefix Length	Total	Hijacked	Recovered	Total	Hijacked	Recovered
amsterdam01	neu01	/23	391	364 (93.09%)	295 (81.04%)	102631	102600 (99.96%)	73270 (71.39%)
amsterdam01	neu01	/24	388	60 (15.46%)	N/A	102591	22738 (22.16%)	N/A
amsterdam01	ufmg01	/23	390	365 (93.58%)	264 (72.32%)	103050	103019 (99.96%)	62746 (60.88%)
amsterdam01	ufmg01	/24	393	93 (23.66%)	N/A	103020	35394 (34.35%)	N/A
amsterdam01	vtrseoul	/23	390	362 (92.82%)	286 (79.00%)	102983	102963 (99.98%)	78777 (76.49%)
amsterdam01	vtrseoul	/24	N/A	73 (N/A%)	N/A	N/A	N/A (N/A%)	N/A
amsterdam01	vtrjohannesburg	/23	389	380 (97.68%)	250 (67.78%)	102801	102785 (99.98%)	59558 (57.93%)
amsterdam01	vtrjohannesburg	/24	389	119 (30.59%)	N/A	102745	36699 (35.71%)	N/A
neu01	amsterdam01	/23	363	389 (107.16%)	53 (13.62%)	102782	102782 (100.0%)	21540 (20.95%)
neu01	amsterdam01	/24	364	351 (96.42%)	N/A	102858	73383 (71.34%)	N/A
neu01	ufmg01	/23	364	366 (100.54%)	162 (44.26%)	102583	102583 (100.0%)	41532 (40.48%)
neu01	ufmg01	/24	365	185 (50.68%)	N/A	102935	55573 (53.98%)	N/A
neu01	vtrseoul	/23	364	363 (99.72%)	125 (34.43%)	102798	102798 (100.0%)	42976 (41.80%)
neu01	vtrseoul	/24	364	229 (62.91%)	N/A	102800	49716 (48.36%)	N/A
neu01	vtrjohannesburg	/23	365	379 (103.83%)	101 (26.64%)	102661	102661 (100.0%)	41922 (40.83%)
neu01	vtrjohannesburg	/24	364	268 (73.62%)	N/A	102683	57340 (55.84%)	N/A
ufmg01	neu01	/23	366	364 (99.45%)	178 (48.90%)	102485	102473 (99.98%)	53933 (52.62%)
ufmg01	neu01	/24	367	169 (46.04%)	N/A	102893	42988 (41.77%)	N/A
ufmg01	amsterdam01	/23	369	391 (105.96%)	86 (21.99%)	102766	102754 (99.98%)	35229 (34.28%)
ufmg01	amsterdam01	/24	366	313 (85.51%)	N/A	102916	62263 (60.49%)	N/A
ufmg01	vtrseoul	/23	365	361 (98.90%)	203 (56.23%)	102667	102655 (99.98%)	66293 (64.57%)
ufmg01	vtrseoul	/24	365	148 (40.54%)	N/A	102784	33742 (32.82%)	N/A
ufmg01	vtrjohannesburg	/23	367	382 (104.08%)	146 (38.21%)	102500	102488 (99.98%)	59390 (57.94%)
ufmg01	vtrjohannesburg	/24	365	220 (60.27%)	N/A	102603	43723 (42.61%)	N/A
vtrseoul	neu01	/23	363	364 (100.27%)	223 (61.26%)	102657	102657 (100.0%)	49668 (48.38%))
vtrseoul	neu01	/24	363	129 (35.53%)	N/A	102857	42997 (41.80%)	N/A
vtrseoul	ufmg01	/23	362	366 (101.10%)	144 (39.34%)	102794	102794 (100.0%)	33682 (32.76%)
vtrseoul	ufmg01	/24	363	219 (60.33%)	N/A	102748	66743 (64.95%)	N/A
vtrseoul	amsterdam01	/23	364	392 (107.69%)	71 (18.11%)	102874	102874 (100.0%)	22392 (21.76%)
vtrseoul	amsterdam01	/24	363	328 (90.35%)	N/A	*	* (*%)	N/A
vtrseoul	vtrjohannesburg	/23	362	380 (104.97%)	117 (30.78%)	102491	102491 (100.0%)	34148 (33.31%)
vtrseoul	vtrjohannesburg	/24	363	244 (67.21%)	N/A	102430	61762 (60.29%)	N/A
vtrjohannesburg	neu01	/23	381	365 (95.80%)	246 (67.39%)	102773	102772 (99.99%)	52733 (51.31%)
vtrjohannesburg	neu01	/24	381	101 (26.50%)	N/A	102872	42327 (41.14%)	N/A
vtrjohannesburg	ufmg01	/23	381	365 (95.80%)	200 (54.79%)	103034	103033 (99.99%)	44181 (42.88%)
vtrjohannesburg	ufmg01	/24	380	153 (40.26%)	N/A	102990	60638 (58.87%)	N/A
vtrjohannesburg	vtrseoul	/23	381	363 (95.27%)	224 (61.43%)	102927	102926 (99.99%)	62020 (60.25%)
vtrjohannesburg	vtrseoul	/24	380	119 (31.31%)	N/A	103039	34081 (33.07%)	N/A
vtrjohannesburg	amsterdam01	/23	379	384 (101.31%)	110 (28.64%)	102942	102941 (99.99%)	37161 (36.09%)
vtrjohannesburg	amsterdam01	/24	379	317 (83.64%)	N/A	102936	59361 (57.66%)	N/A

Table 25: Results for measurements in the control plane and data plane of experiments involving more specific prefixes without *prepend*, where the original announcement varies from /23 to /24, while the hijacks are carried out using /24.

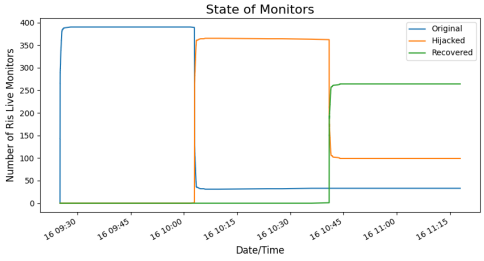
D PREFIX LENGTH GRAPHS



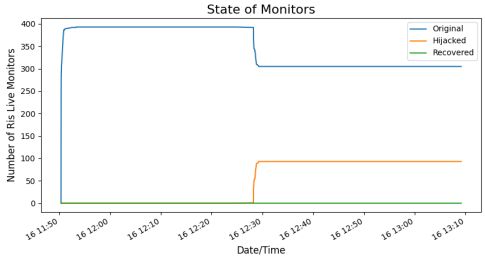
(a) vs. neu01 - /23 original prefix



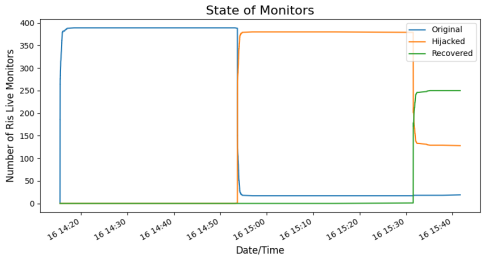
(b) vs. neu01 - /24 original prefix



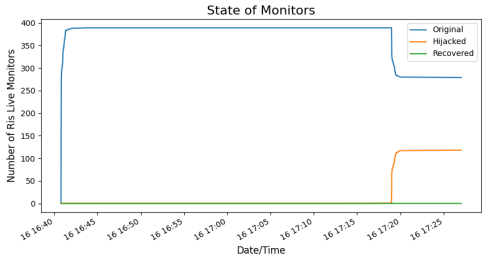
(c) vs. ufmg01 - /23 original prefix



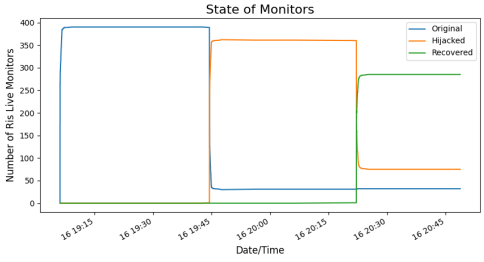
(d) vs. ufmg01 - /24 original prefix



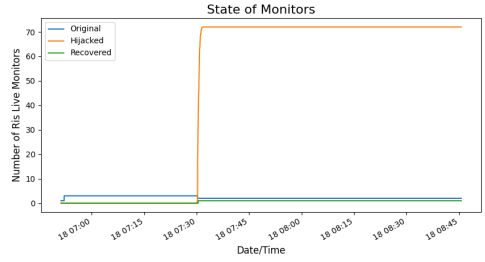
(e) vs. vtrjohannesburg - /23 original prefix



(f) vs. vtrjohannesburg - /24 original prefix

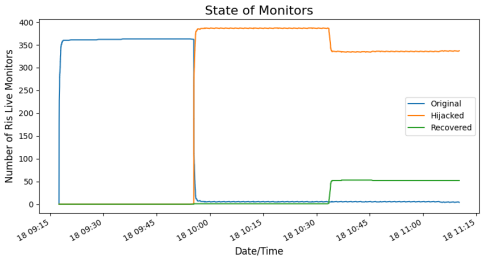


(g) vs. vtrseoul - /23 original prefix

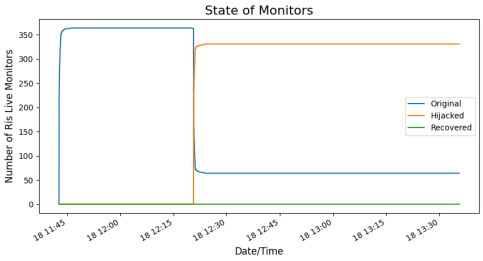


(h) vs. vtrseoul - /24 original prefix

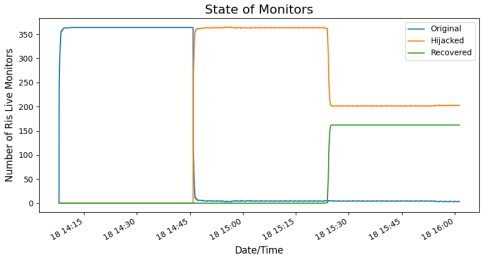
Figure 28: amsterdam01 as victim while announcing a /23 or a /24 prefix.



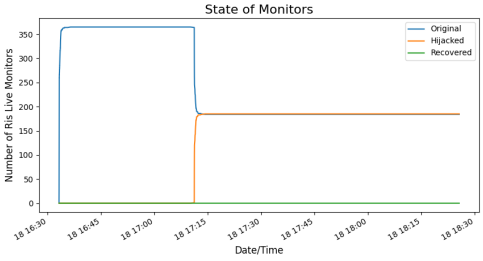
(a) vs. amsterdam01 - /23 original prefix



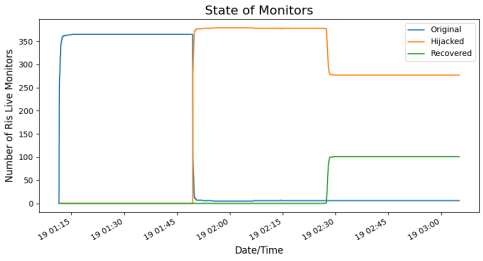
(b) vs. amsterdam01 - /24 original prefix



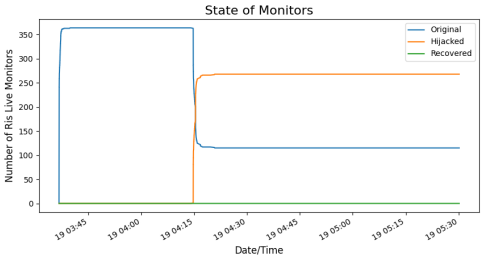
(c) vs. ufmg01 - /23 original prefix



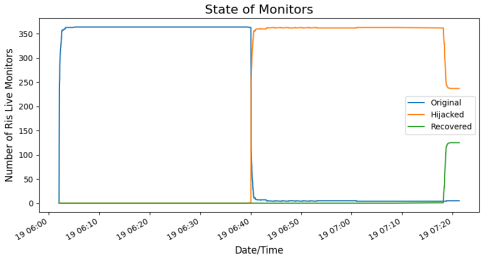
(d) vs. ufmg01 - /24 original prefix



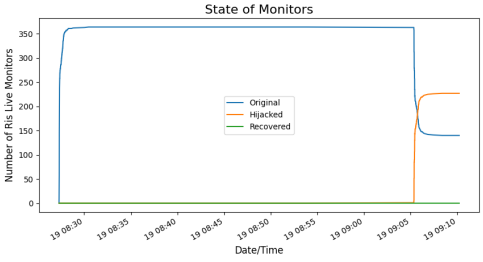
(e) vs. vtrjohannesburg - /23 original prefix



(f) vs. vtrjohannesburg - /24 original prefix

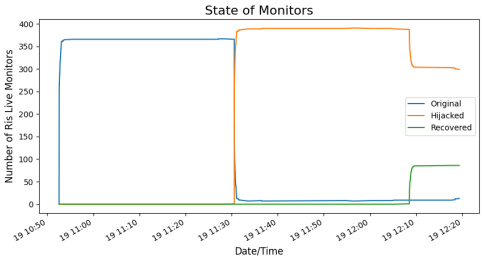


(g) vs. vtrseoul - /23 original prefix

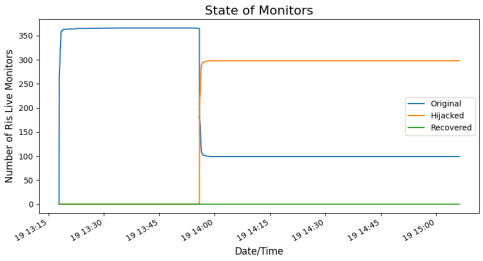


(h) vs. vtrseoul - /24 original prefix

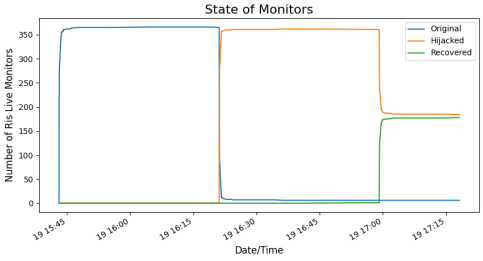
Figure 29: neu01 as victim while announcing a /23 or a /24 prefix.



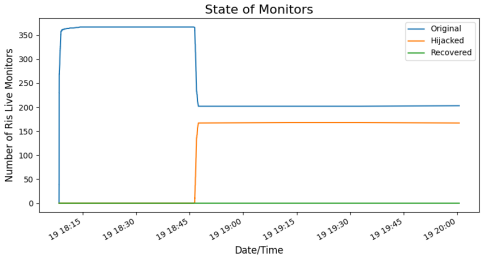
(a) vs. amsterdam01 - /23 original prefix



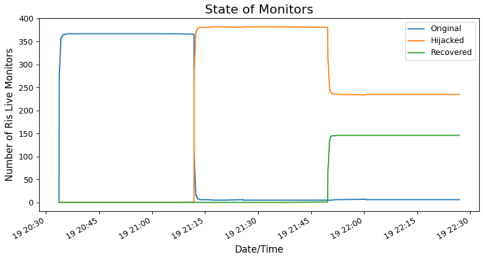
(b) vs. amsterdam01 - /24 original prefix



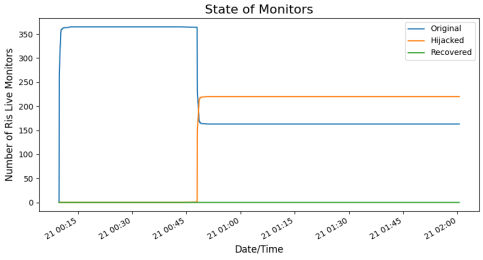
(c) vs. neu01 - /23 original prefix



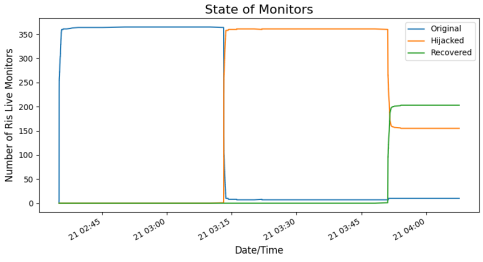
(d) vs. neu01 - /24 original prefix



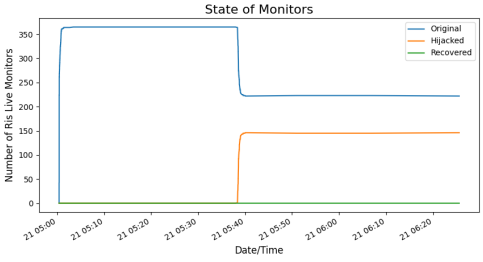
(e) vs. vtrjohannesburg - /23 original prefix



(f) vs. vtrjohannesburg - /24 original prefix

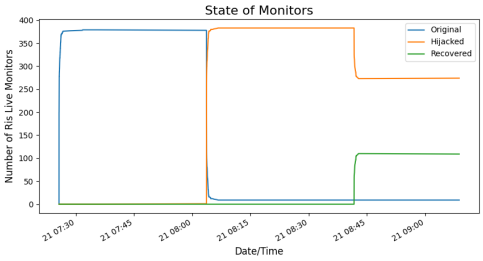


(g) vs. vtrseoul - /23 original prefix

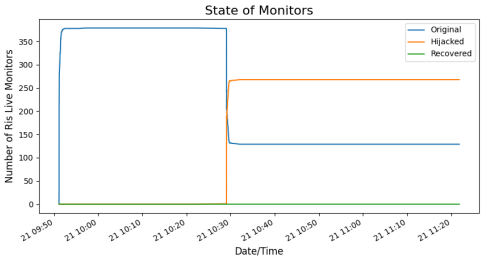


(h) vs. vtrseoul - /24 original prefix

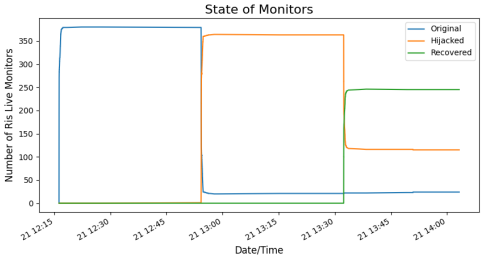
Figure 30: ufmg01 as victim while announcing a /23 or a /24 prefix.



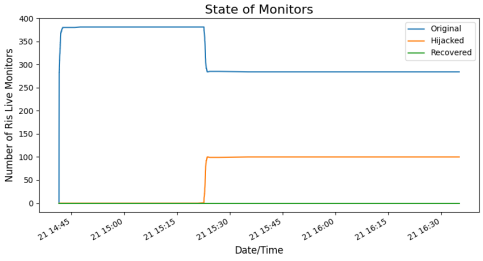
(a) vs. amsterdam01 - /23 original prefix



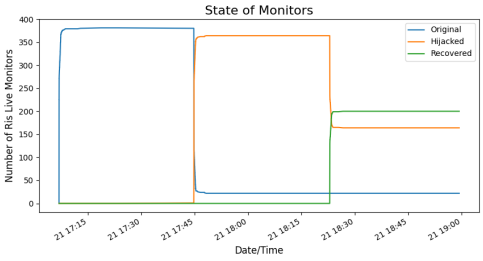
(b) vs. amsterdam01 - /24 original prefix



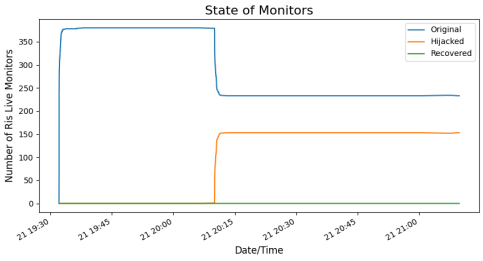
(c) vs. neu01 - /23 original prefix



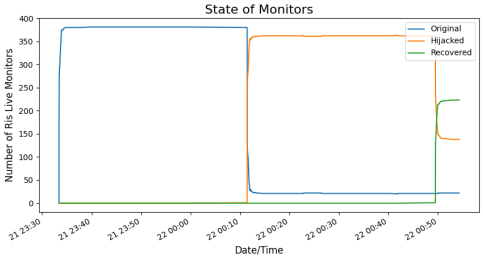
(d) vs. neu01 - /24 original prefix



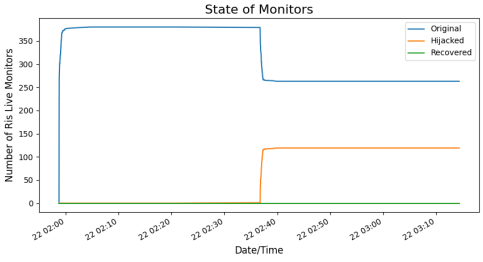
(e) vs. ufmg01 - /23 original prefix



(f) vs. ufmg01 - /24 original prefix

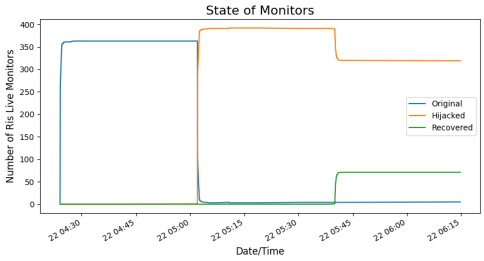


(g) vs. vtrseoul - /23 original prefix

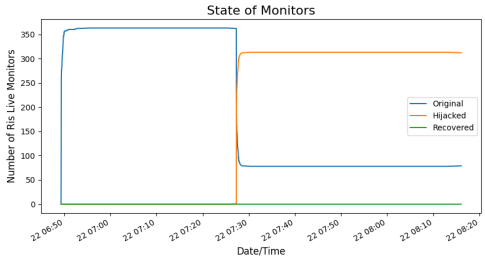


(h) vs. vtrseoul - /24 original prefix

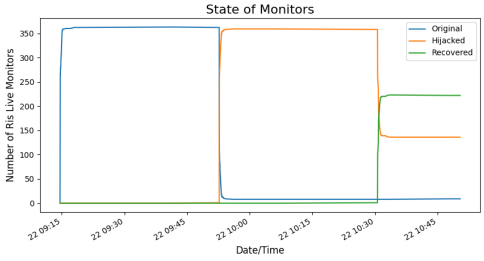
Figure 31: vtrjohannesburg as victim while announcing a /23 or a /24 prefix.



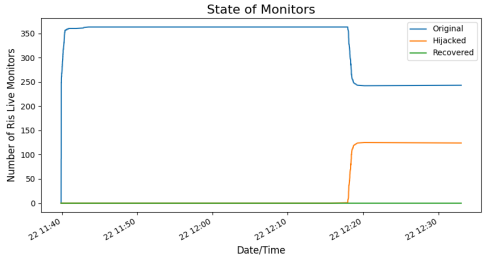
(a) vs. amsterdam01 - /23 original prefix



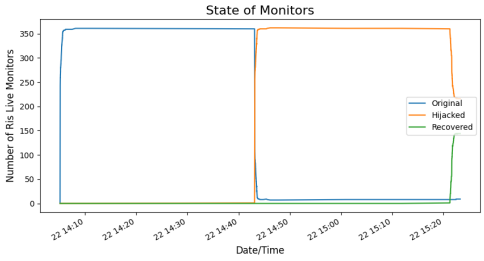
(b) vs. amsterdam01 - /24 original prefix



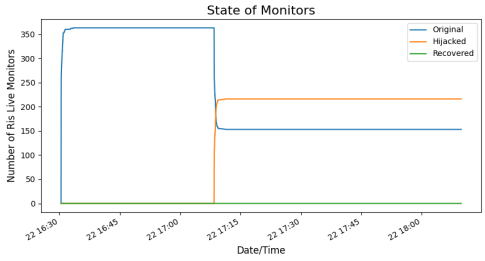
(c) vs. neu01 - /23 original prefix



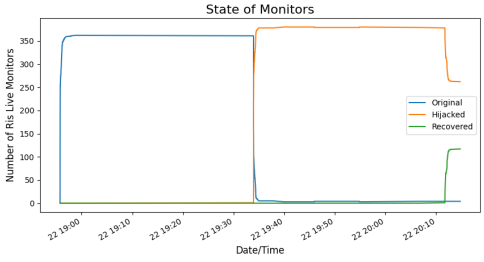
(d) vs. neu01 - /24 original prefix



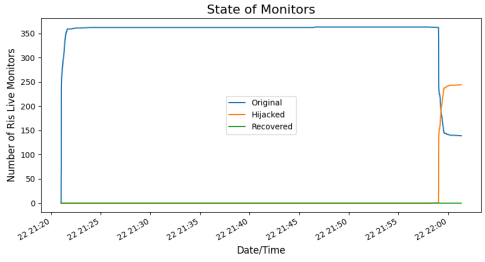
(e) vs. ufmg01 - /23 original prefix



(f) vs. ufmg01 - /24 original prefix



(g) vs. vtrjohannesburg - /23 original prefix



(h) vs. vtrjohannesburg - /24 original prefix

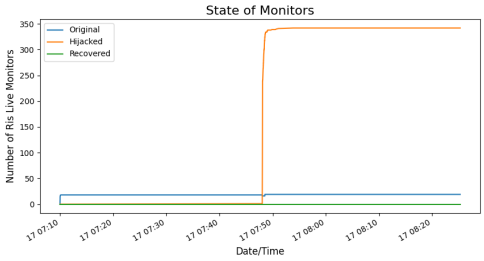
Figure 32: vtrseoul as victim while announcing a /23 or a /24 prefix.

E CONNECTIVITY TABLE

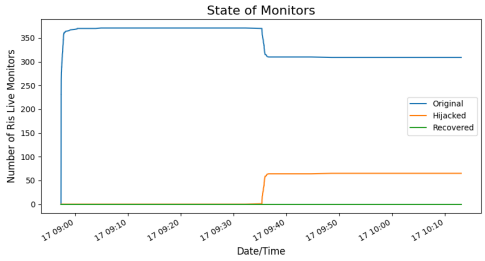
Experiment Configuration			Control Plane Monitors		Data Plane Targets	
Origin	Hijacker	Peers/IX	Total	Hijacked	Total	Hijacked
amsterdam01	neu01	AMS-IX	18	345 (N/A)	1162	619 (53.27%)
amsterdam01	neu01	Bit BV	371	65 (17.52%)	98872	21726 (21.97%)
amsterdam01	neu01	AMS-IX, Bit BV	371	65 (17.52%)	98939	21853 (22.08%)
amsterdam01	neu01	Coloclue, Bit BV	386	63 (16.32%)	98794	21681 (21.94%)
amsterdam01	neu01	AMS-IX, Coloclue, Bit BV	386	65 (16.83%)	98779	21818 (22.08%))
amsterdam01	neu01	Coloclue	391	39 (9.97%)	98884	20967 (21.20%)
amsterdam01	neu01	AMS-IX, Coloclue	391	39 (9.97%)	98716	20980 (21.25%)
amsterdam01	ufmg01	AMS-IX	18	352 (N/A)	1242	681 (54.83%)
amsterdam01	ufmg01	Bit BV	373	103 (27.61%)	99227	34642 (34.91%)
amsterdam01	ufmg01	AMS-IX, Bit BV	371	101 (27.22%)	99320	34644 (34.88%)
amsterdam01	ufmg01	Bit BV, Coloclue	386	101 (26.16%)	99210	34634 (34.90%)
amsterdam01	ufmg01	AMS-IX, Bit BV, Coloclue	390	103 (26.41%)	99262	35765 (36.03%)
amsterdam01	ufmg01	Coloclue	389	84 (21.59%)	99310	34012 (34.24%)
amsterdam01	ufmg01	AMS-IX, Coloclue	386	85 (22.02%)	99199	34274 (34.55%)
amsterdam01	vtrseoul	AMS-IX	18	361 (N/A)	1188	644 (54.20%)
amsterdam01	vtrseoul	Bit BV	367	121 (32.97%)	98721	38399 (38.89%)
amsterdam01	vtrseoul	AMS-IX, Bit BV	370	120 (32.43%)	98893	38407 (38.83%)
amsterdam01	vtrseoul	Bit BV, Coloclue	370	121 (32.70%)	98878	38110 (38.54%)
amsterdam01	vtrseoul	AMS-IX, Bit BV, Coloclue	387	119 (30.74%)	98783	38253 (38.72%)
amsterdam01	vtrseoul	Coloclue	390	88 (22.56%)	98741	38527 (39.01%)
amsterdam01	vtrseoul	AMS-IX, Coloclue	391	86 (21.99%)	98720	38608 (39.10%)
amsterdam01	vtrjohannesburg	AMS-IX	18	346 (N/A)	1146	591 (51.57%)
amsterdam01	vtrjohannesburg	Bit BV	371	76 (20.48%)	98761	22818 (23.10%)
amsterdam01	vtrjohannesburg	AMS-IX, Bit BV	370	77 (20.81%)	98771	23129 (23.41%)
amsterdam01	vtrjohannesburg	Bit BV, Coloclue	387	79 (20.41%)	98822	23583 (23.86%)
amsterdam01	vtrjohannesburg	AMS-IX, Bit BV, Coloclue	388	78 (20.10%)	98777	23517 (23.80%)
amsterdam01	vtrjohannesburg	Coloclue	390	69 (17.69%)	98774	23463 (23.75%)
amsterdam01	vtrjohannesburg	AMS-IX, Coloclue	390	70 (17.94%)	98798	23505 (23.79%)

Table 26: Result for selective announcement experiments were amsterdam01 is the victim.

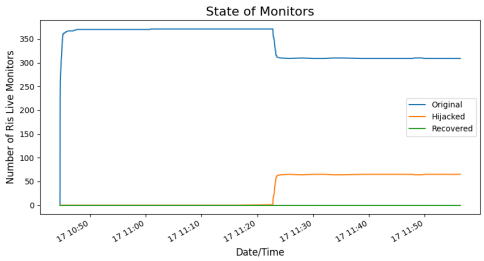
F CONNECTIVITY GRAPHS



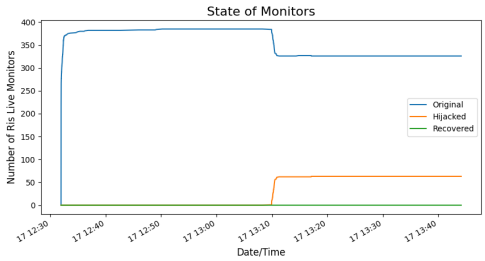
(a) Announcing to AMS-IX



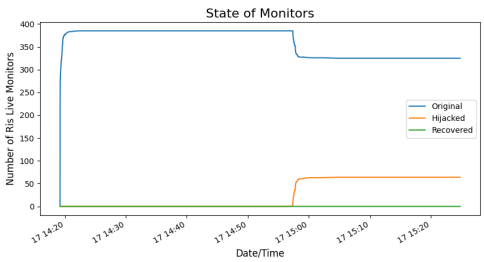
(b) Announcing to BitBV



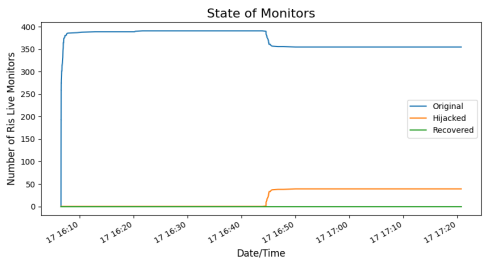
(c) Announcing to AMS-IX and BitBV



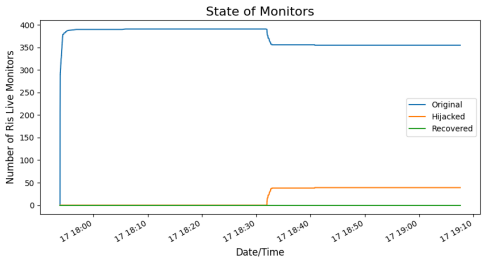
(d) Announcing to BitBV and Coloclue



(e) Announcing to AMS-IX, BitBV and Coloclue

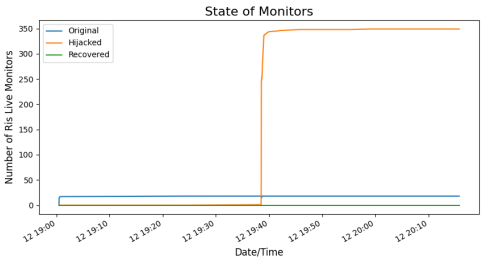


(f) Announcing to Coloclue

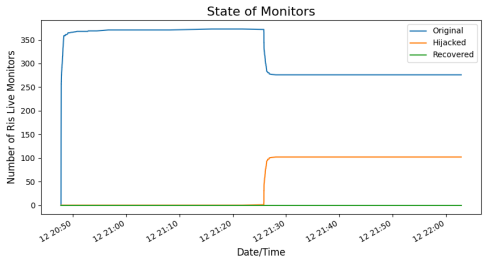


(g) Announcing to AMS-IX and Coloclue

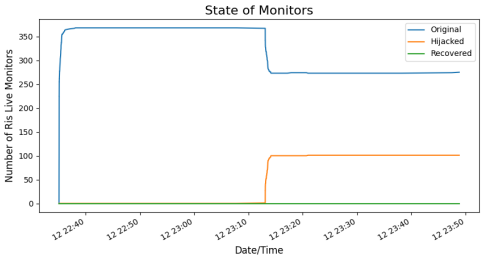
Figure 33: amsterdam01 as victim while using selective announcement. neu01 as attacker.



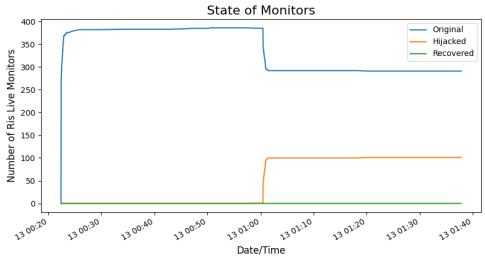
(a) Announcing to AMS-IX



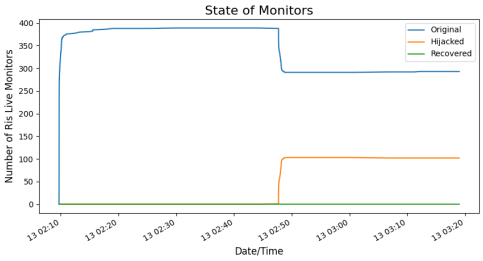
(b) Announcing to BitBV



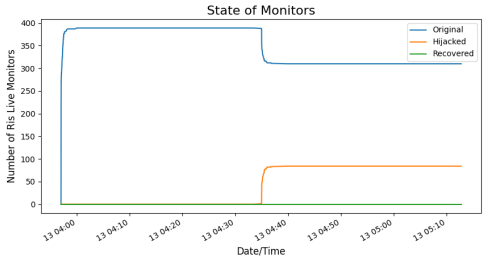
(c) Announcing to AMS-IX and BitBV



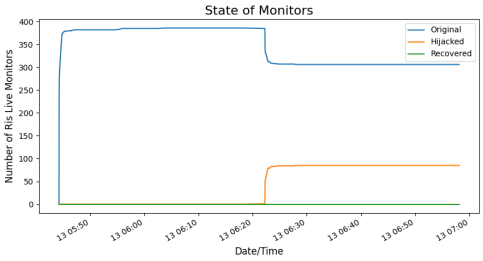
(d) Announcing to BitBV and Coloclue



(e) Announcing to AMS-IX, BitBV and Coloclue

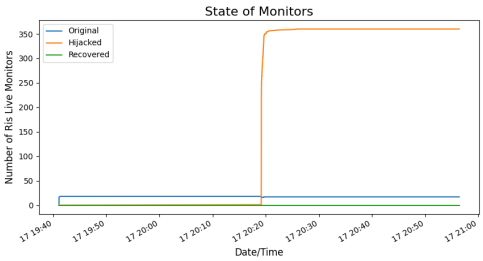


(f) Announcing to Coloclue

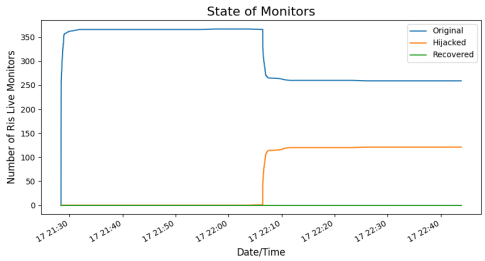


(g) Announcing to AMS-IX and Coloclue

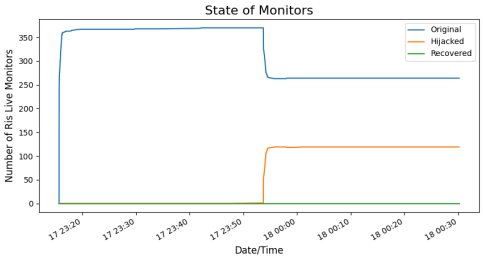
Figure 34: amsterdam01 as victim while using selective announcement. ufmg01 as attacker.



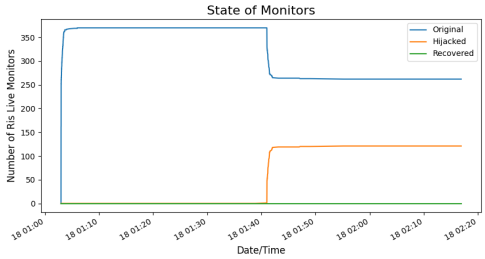
(a) Announcing to AMS-IX



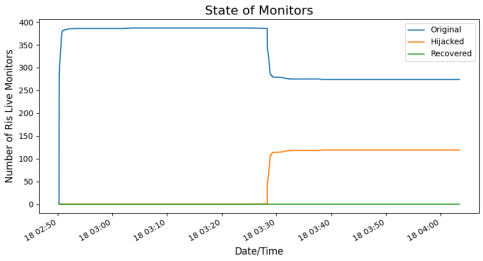
(b) Announcing to BitBV



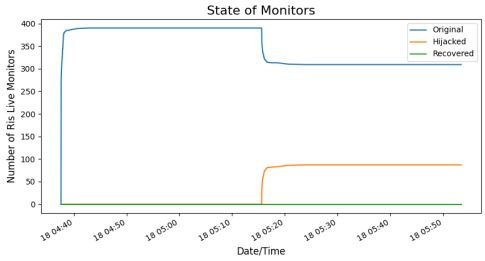
(c) Announcing to AMS-IX and BitBV



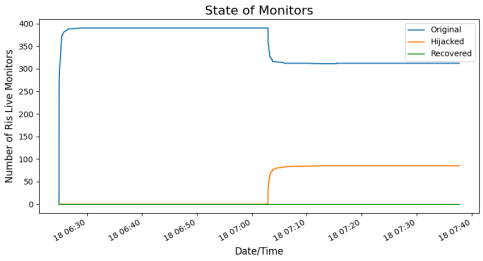
(d) Announcing to BitBV and Coloclue



(e) Announcing to AMS-IX, BitBV and Coloclue

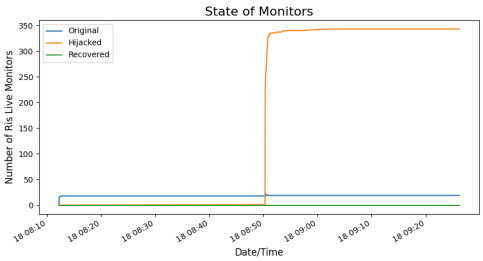


(f) Announcing to Coloclue

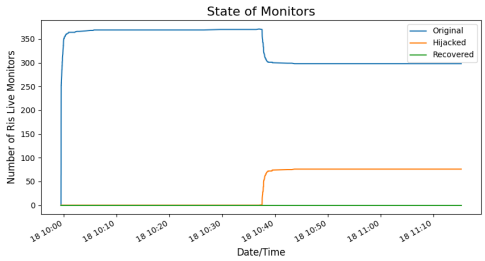


(g) Announcing to AMS-IX and Coloclue

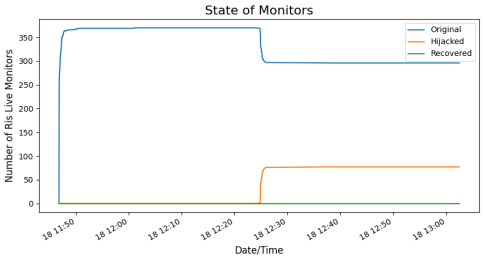
Figure 35: amsterdam01 as victim while using selective announcement. vtrjohannesburg as attacker.



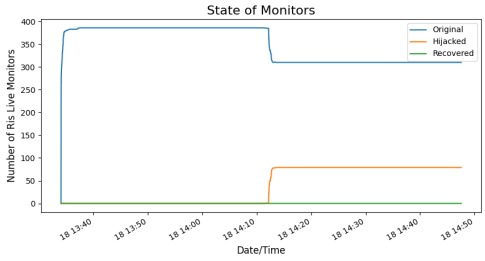
(a) Announcing to AMS-IX



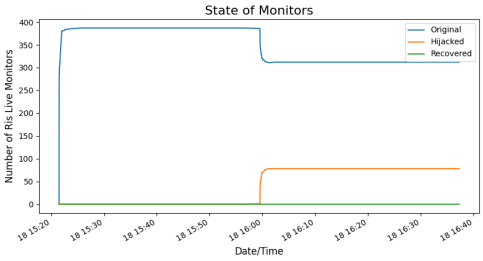
(b) Announcing to BitBV



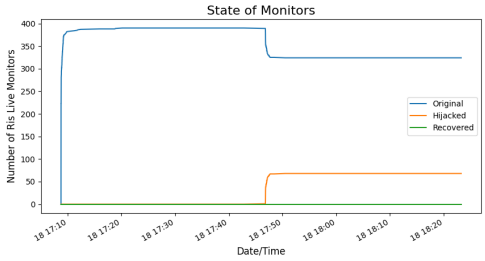
(c) Announcing to AMS-IX and BitBV



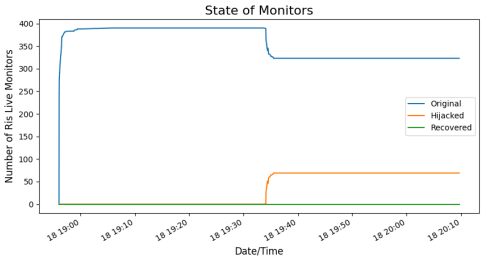
(d) Announcing to BitBV and Coloclue



(e) Announcing to AMS-IX, BitBV and Coloclue



(f) Announcing to Coloclue



(g) Announcing to AMS-IX and Coloclue

Figure 36: amsterdam01 as victim while using selective announcement. vtrseoul as attacker.