ARTEMIS: Neutralizing BGP Hijacking within a Minute

Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti

Abstract—BGP prefix hijacking is a critical threat to Internet organizations and users. Despite the availability of several defense approaches (ranging from RPKI to popular third-party services), none of them solves the problem adequately in practice. In fact, they suffer from: (i) lack of detection comprehensiveness, allowing sophisticated attackers to evade detection, (ii) limited accuracy, especially in the case of third-party detection, (iii) delayed verification and mitigation of incidents, reaching up to days, and (iv) lack of privacy and of flexibility in post-hijack counteractions, on the side of network operators. In this work, we propose ARTEMIS (Automatic and Real-Time dEtection and MItigation System), a defense approach (a) based on *accurate and fast detection operated by the AS itself*, leveraging the pervasiveness of publicly available BGP monitoring services and their recent shift towards real-time streaming, thus (b) enabling *flexible and fast mitigation* of hijacking events. Compared to previous work, our approach combines characteristics desirable to network operators such as comprehensiveness, accuracy, speed, privacy, and flexibility. Finally, we show through real-world experiments that, with the ARTEMIS approach, prefix hijacking can be neutralized within a minute.

1 Introduction

Autonomous Systems (ASes) use the Border Gateway Protocol (BGP) [35] to advertise their IP prefixes and establish inter-domain routes in the Internet. BGP is a distributed protocol, lacking authentication of routes. As a result, an AS is able to advertise illegitimate routes for IP prefixes it does not own. These illegitimate advertisements propagate and "pollute" many ASes, or even the entire Internet, affecting service availability, integrity, and confidentiality of communications. This phenomenon, called *BGP prefix hijacking* can be caused by router misconfiguration [1], [2] or malicious attacks [3], [56], [65]. Events with significant impact are frequently observed [4], [5], [47], [65], highlighting – despite the severity of such Internet infrastructural vulnerability – the ineffectiveness of existing countermeasures.

Currently, networks rely on *practical reactive mechanisms* to try to defend against prefix hijacking, since proposed *proactive* mechanisms [38], [39], [42], [43], [64] (*e.g.*, RPKI) are fully efficient only when globally deployed, and operators are reluctant to deploy them due to associated technical and financial costs [27], [45], [49]. Defending against hijacking reactively consists of two steps: *detection* and *mitigation*. Detection is mainly provided by third-party services, *e.g.*, [8], that notify networks about suspicious events involving their prefixes, based on routing information (such as traceroutes [69] or BGP updates [8]). The affected networks then proceed to mitigate the event, *e.g.*, by announcing more specific prefixes, or contacting other ASes to filter announcements.

However, due to a mix of technological and practical deployability issues, current reactive approaches are largely inadequate. In this paper, we address these issues by proposing ARTEMIS (Automatic and Real-Time dEtection and MItigation System), a *self-operated* and *unified* detection and mitigation approach based on *control-plane monitoring*. Specifically, the state of the art suffers from 4 main problems:

- **Evasion**. None of the detection approaches in literature is capable of detecting all attack configurations (nor can they be easily combined), thus allowing sophisticated attackers to evade them. We propose a modular taxonomy describing all variations of attack scenarios and we use it to carefully analyze detection comprehensiveness of related work. *ARTEMIS significantly overcomes limitations of the state of the art by covering all attack configurations in the context of a common threat model*.
- **Accuracy**. Legitimate changes in the routing policies of a network (*e.g.*, announcing a sub-prefix for traffic engineering or establishing a new peering connection), could be considered suspicious events by the majority of third-party detection systems [25], [36], [40], [63], [69]. To avoid this, operators would need to timely inform third parties about every routing decision they make and share private information. On the other hand, adopting a less strict policy to compensate for the lack of updated information and reduce false positives (FP), incurs the danger of neglecting real hijacking events (false negatives – FN). *We designed ARTEMIS detection to be run directly by the network operator without relying on a third party, thus leveraging fully and constantly (and potentially automatically) updated information that enables 0% FP and FN for most of the attack scenarios and a configurable FP–FN trade-off otherwise*.
- **Speed**. A side effect of the inaccuracy of third-party approaches is the need for manual verification of alerts, which inevitably causes slow mitigation of malicious events (*e.g.*, hours or days). Few minutes of diverted traffic can cause large financial losses due to service unavailability or security breaches. On the contrary, *ARTEMIS is a*

- P. Sermpezis, V. Kotronis, and P. Gigis are with ICS-FORTH, Greece; X. Dimitropoulos is with ICS-FORTH and University of Crete, Greece; A. King, and A. Dainotti are with CAIDA, UC San Diego, USA; D. Cicalese is with CAIDA, UC San Diego, CA, USA, and Telecom ParisTech, France.

fully automated solution integrating detection and mitigation, allowing an AS to quickly neutralize attacks. We conduct real hijacking experiments in the Internet demonstrating that ARTEMIS can detect attacks *within seconds* and *neutralize them within a minute, i.e., orders of magnitude faster than current practices*.

- **Privacy and Flexibility**. One of the issues that impedes the adoption of third-party detection is privacy, *e.g.*, ISPs usually do not disclose their peering policies. Similarly, operators are sometimes reluctant to adopt mitigation services requiring other organizations to announce their prefixes or tunnel their traffic. *ARTEMIS offers full privacy for detection and the option to achieve self-operated mitigation*. Another factor affecting willingness to externalize mitigation is cost. Trade-offs between cost, privacy, and risk may be evaluated differently by the same organization for distinct prefixes they own. *Leveraging the availability of local private information and its fully automated approach, ARTEMIS offers the flexibility to customize mitigation (*e.g., *self-operated or third-party-assisted) per prefix and per attack class*.

The ARTEMIS approach relies on two key observations: *(i)* today's public BGP monitoring infrastructure (such as RouteViews [13] and RIPE RIS [12]) is much more advanced than when previous solutions for BGP hijacking detection were proposed, making it a valuable resource for comprehensive, live monitoring of the Internet control plane that is available to anybody; *(ii)* shifting from a third-party perspective to a self-operated approach enables us to effectively address the long-standing and persistent issues undermining the state of the art in BGP hijacking defense approaches.

In this work, we first define our threat model and propose a novel attack taxonomy used throughout the paper (§ 2). We investigate the visibility (from the public monitoring infrastructure § 3) and impact of different hijacking types in § 4, and then design the ARTEMIS detection (§ 5) and mitigation (§ 6) approach. We evaluate our design decisions through simulations and analysis of real-world Internet control-plane measurements (§ 3,§ 4,§ 5,§ 6). Furthermore, the ARTEMIS approach is immediately deployable *today*: we build a prototype system implementing our approach, and we show its effectiveness through experiments on the real Internet (§ 7). Finally, we provide an extensive background on the state of the art, both in terms of practical experience (by conducting a survey among operators and referring to reported events; § 8.1) and related literature (§ 8.2).

2 Threat Model and Attack Taxonomy

We consider a common and general hijacking threat model (*e.g.*, similarly to [59]), where a hijacker controls a single AS and its edge routers, and has full control of the control plane and data plane within its own AS. The hijacker can arbitrarily manipulate the BGP messages that it sends to its neighboring ASes (control plane) and the traffic that crosses its network (data plane), but has otherwise no control over BGP messages and traffic exchanged between two other ASes.

In this threat model, there are three dimensions that characterize how a hijacking attack can be carried out: *(i)* the affected prefix, *(ii)* the manipulation of the AS-PATH in the BGP messages, and *(iii)* how the (hijacked) data-plane traffic is treated. Any attack can be represented by a "point" in this three-dimensional "space". Table 1 presents all possible attack combinations (three leftmost columns); "*" denote wildcarded fields.

In the following, we provide a taxonomy of hijacking attacks, based on these 3 properties. For the sake of demonstration, we assume that *AS1* owns and legitimately announces the prefix *10.0.0.0/23*, and *AS2* is the hijacking AS. We denote a BGP message with two fields: its AS-PATH and announced prefix. For example, {*ASx, ASy, AS1 – 10.0.0.0/23*} is a BGP announcement for prefix *10.0.0.0/23*, with AS-PATH {*ASx, ASy, AS1*}, originated by the legitimate AS (*AS1*).

2.1 Classification by Announced AS-Path

Origin-AS (or Type-0) hijacking: The hijacker *AS2* announces – as its own – a prefix that it is not authorized to originate, *e.g.*, {*AS2 – 10.0.0.0/23*}. This is the most commonly observed hijack type, and might occur either due to an attack or a misconfiguration.

Type-N hijacking ($N \geq 1$**):** The hijacker *AS2* deliberately announces an illegitimate path for a prefix it does not own. The announced path contains the ASN of the victim (first AS in the path) and hijacker (last AS in the path), *e.g.*, {*AS2, ASx, ASy, AS1 – 10.0.0.0/23*}, while the sequence of ASes in the path is not a valid route, *e.g.*, AS2 is not an actual neighbor of ASx. In our taxonomy, the position of the *rightmost fake link* in the forged announcement determines the *type*. E.g., {*AS2, AS1 – 10.0.0.0/23*} is a *Type-1* hijacking, {*AS2, ASy, AS1 – 10.0.0.0/23*} is a *Type-2* hijacking, etc.

Type-U: The hijacker leaves the legitimate AS-PATH unaltered (but may alter the announced prefix [52]) [1] .

2.2 Classification by Affected Prefix

Exact prefix hijacking: The hijacker announces a path for exactly the same prefix announced by the legitimate AS. Since shortest AS-paths are typically preferred, only a part of the Internet that is close to the hijacker (*e.g.*, in terms of AS hops) switches to routes towards the hijacker. The examples presented above (§ 2.1) are exact prefix hijacks.

Sub-prefix hijacking: The hijacker *AS2* announces a more specific prefix, *i.e.*, a sub-prefix of the prefix of the legitimate AS. For example, *AS2* announces a path {*AS2 – 10.0.0.0/24*} or {*AS2, ASx, ASy, AS1 – 10.0.0.0/24*}. Since in BGP more specific prefixes are preferred, *the entire Internet* routes traffic towards the hijacker to reach the announced sub-prefix.

Squatting: The hijacker AS announces a prefix owned but not (currently) announced by the owner AS [46].

2.3 Classification by Data-Plane Traffic Manipulation

The effect of a hijack is to redirect traffic for the affected prefix to/through the network of the hijacker AS. This attracted traffic can be *(i)* dropped (*blackholing, BH*), *(ii)* manipulated or eavesdropped and then sent on to the victim *AS1* (*man-in-the-middle, MM*), or *(iii)* used in an impersonation of the

1. If the announced prefix is also left unaltered (*i.e.*, no path or prefix manipulation; see § 2.2), then the event is not a hijack (no misuse of BGP) but a traffic manipulation attempt, out of the scope of this paper.

TABLE 1: Comparison of BGP prefix hijacking detection systems/services w.r.t. ability to detect different classes of attacks.

Class of Hijacking Attack			Control-plane System/Service			Data-plane System/Service		Hybrid System/Service		
Affected prefix	AS-PATH (Type)	Data plane	ARTEMIS	Cyclops (2008) [25]	PHAS (2006) [40]	iSpy (2008) [68]	Zheng *et al.* (2007) [69]	HEAP (2016) [59]	Argus (2012) [63]	Hu *et al.* (2007) [36]
Sub	U	*	✓	×	×	×	×	×	×	×
Sub	0/1	BH	✓	×	✓	×	×	✓	✓	✓
Sub	0/1	IM	✓	×	✓	×	×	✓	×	✓
Sub	0/1	MM	✓	×	✓	×	×	×	×	×
Sub	≥ 2	BH	✓	×	×	×	×	✓	✓	✓
Sub	≥ 2	IM	✓	×	×	×	×	✓	×	✓
Sub	≥ 2	MM	✓	×	×	×	×	×	×	×
Exact	0/1	BH	✓	✓	✓	✓	×	×	✓	✓
Exact	0/1	IM	✓	✓	✓	×	✓	×	×	✓
Exact	0/1	MM	✓	✓	✓	×	✓	×	×	×
Exact	≥ 2	BH	✓	×	×	✓	×	×	✓	✓
Exact	≥ 2	IM	✓	×	×	×	✓	×	×	✓
Exact	≥ 2	MM	✓	×	×	×	✓	×	×	×

victim's service by responding to the senders (*imposture, IM*). While BH attacks might be easily noticed in the data plane (since a service is interrupted), MM or IM attacks can be invisible to the victim AS or the other involved ASes.

2.4 Example Hijack Scenarios & Motivations

The following examples illustrate different hijack scenarios, their underlying motivation, and how they are classified according to the presented taxonomy.

Human Error. The hijack is the result of a routing misconfiguration; *e.g.,* the leakage of a full BGP table from China Telecom [2], led to an accidental large-scale Type-0 exact-prefix hijack, with blackholing on the data plane.

High Impact Attack. The hijack is intentional, with widespread impact; *e.g.,* Pakistan Telecom engaged in a Type-0 sub-prefix hijack, blackholing YouTube's services for approximately 2 hours worldwide [15].

Targeted, Stealthy Attack. The hijacking AS launches a very targeted attack, attempting to intercept traffic (man-in-the-middle), while remaining under the radar on the control plane (Type-N or Type-U attack); *e.g.,* a Russian network hijacked traffic destined to Visa and Mastercard in 2017 [4].

The "Best" Attack. Motivations behind hijacks differ; there is no one "best" attack type that is always preferred. For example, an attacker may resort to a Type-N ($N > 0$) hijack to *(i)* evade simple detection systems currently used by operators or bypass RPKI ROV, or *(ii)* delay manual investigation and recovery from the malicious event; in contrast to origin AS validation, inferring that a link in an AS-path is fake is a hard challenge. Moreover, while a sub-prefix Type-U hijack can be very effective, it might be neither possible nor ideal in some cases; *e.g.,* the upstream providers of a hijacker might be configured to not accept routes for prefixes not owned by their customers.

3 Datasets and Tools

3.1 Control-Plane Monitoring

We study BGP prefix hijacking and evaluate ARTEMIS using publicly available services that offer control-plane monitoring from multiple *monitors* worldwide. We define as monitors the ASes that peer through their BGP routers with the infrastructure of the monitoring services, and provide BGP feeds (*i.e.,* BGP updates and RIBs). We consider the following monitoring services and tools.

BGPmon [7] (from Colorado State University[2]) provides *live* BGP feeds from several BGP routers of (a) the Route-Views [13] sites, and (b) a few dozens of peers worldwide.

RIPE RIS [11]. RIPE's Routing Information System (RIS) has 21 route collectors (RCs) distributed worldwide, collecting BGP updates from around 300 peering ASes. Currently, 4 RCs provide *live* BGP feeds (from approx. 60 monitors) [12], while data from all RCs can be accessed (with a delay of a few minutes) through RIPEstat [57] or the tools of CAIDA's BGPStream [9], [51] framework. However, RIPE RIS is in the process of upgrading all its RCs towards providing real-time BGP feeds [16].

RouteViews [13] provides control-plane information collected from 19 RCs that are connected to nearly 200 ASes worldwide. A subset of the RouteViews RCs provide *live* BGP feeds (through BGPmon), while all data can be accessed with a delay of approx. 20min (using tools from BGPStream). Several RouteViews monitors have started experimentally deploying live BMP [61] feeds [22] accessible through BGPStream; it is thus foreseeable that in the near future more live BGP feeds will be publicly available.

Our ARTEMIS prototype employs *live* BGP feeds such as the BGPmon and RIPE RIS *streaming services*. However, to understand the effect of adding more data sources, we perform additional simulations and real data analysis including BGP feeds from *all the monitors* of RIPE RIS and RouteViews services, which we access through the API of BGPStream[3]. A summary of the monitoring services that we use in this paper is given in Table 2.

3.2 Simulation Methodology

In this paper, through extensive simulations, we evaluate the impact of different types of hijacks, the performance of the monitoring services, and the efficiency of various mitigation methods. To simulate the Internet routing system, we use a largely adopted methodology [26], [30], [31], [34]:

2. BGPmon is also the name of a commercial network monitoring service. Throughout this paper, BGPmon refers to the (free) service provided by Colorado State University, unless stated otherwise.

3. In our simulations we consider only the full-feed monitors [51] of RIPE RIS and RouteViews that are more reliable: we include only full-feed monitors that consistently provided data during March 2017.

TABLE 2: Control-plane monitoring services

		#monitors	delay
Stream services	BGPmon [7]	8	$< 1s$
	RIPE RIS (stream) [12]	57	$< 1s$
	Total (unique)	65	
All services (BGPStream)	RouteViews [13]	128	$\sim 20min$
	RIPE RIS [11]	120	$\sim 5min$
	Total (unique)	218	

we build the Internet topology graph from a large experimentally collected dataset [17], use classic frameworks for inferring routing policies on existing links [29], and simulate BGP message exchanges between connected ASes.

Building the Internet Topology Graph. We use CAIDA's AS-relationship dataset [17], which is collected based on the methodology of [44] and enriched with many extra peering (p2p) links [32]. The dataset contains a list of AS pairs with a peering link, which is annotated based on their relationship as *c2p* (*customer to provider*) or *p2p* (*peer to peer*). In this topology, we represent the monitors of § 3.1 as AS nodes using their associated ASNs.

Simulating Routing Policies. When an AS learns a new route for a prefix (or, announces a new prefix), it updates its routing table and, if required, sends BGP updates to its neighbors. The update and export processes are determined by its routing policies. In our simulator, and similarly to previous works [26], [30], [31], [34], we select the routing policies based on the classic Gao-Rexford conditions that guarantee global BGP convergence and stability [29].

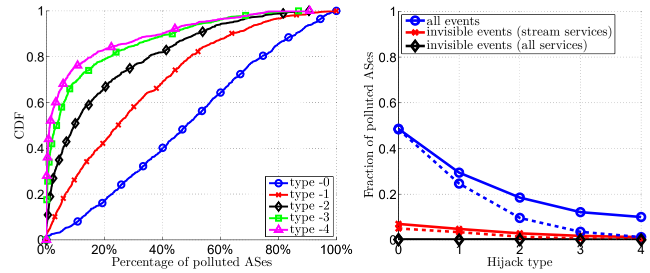
4 Impact and Visibility

In this section, through simulation, we first study the potential impact of different hijacking types on the control plane, *i.e.*, their ability to pollute the routing tables of other ASes. We then evaluate the potential of BGP monitoring services (*e.g.*, RouteViews) to observe these events. Our simulations suggest that the *current BGP monitoring infrastructure is able to observe all the events with significant impact*. These results help us design our detection approach (§ 5) and inform our flexible mitigation approach (§ 6).

4.1 Impact of Hijacks on the Control Plane

An AS receiving routes from two different neighboring ASes for the same prefix, selects one of them to route its traffic. This path selection is based on peering policies, local preferences, and the `AS-PATH` lengths of the received routes. As a result, the impact of an *exact prefix hijacking* event on the control plane depends on such routing selections. To understand how the impact of these events can vary, we perform simulations on the AS-level topology of the Internet (see § 3). For each scenario, we simulate 1000 runs with varying {legitimate-AS, hijacker-AS} pairs[4]. We refer to an AS as *polluted* if it selects a path that contains the ASN of the hijacker. To quantify the impact of a hijack, we calculate the *fraction of ASes polluted by the event*, excluding those ASes that were already polluted before the hijack (*e.g.*, customers of the hijacker AS that always route traffic through it). We limit the analysis in this section to exact prefix hijacking, since *sub-prefix hijacking* pollutes the entire Internet (§ 2.2).

4. 1000 simulation runs provide significant statistical accuracy (*i.e.*, small confidence intervals for mean/median values) in all our scenarios.



(a) (b)

Fig. 1: *Impact* of different hijack types: (a) CDFs, and (b) mean (continuous lines) and median (dashed lines) values of the fraction of polluted ASes over 1000 simulations for different hijack types. Hijacking events of all types can have a large impact, with smaller types being on average more impactful.

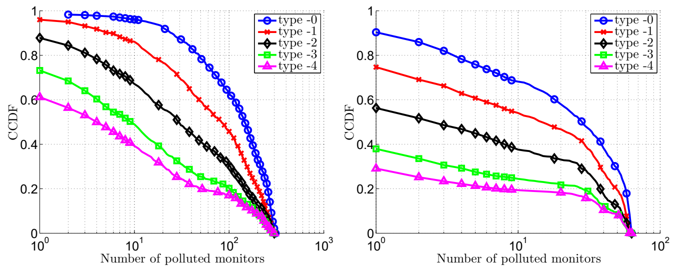
Hijacking events of smaller AS-path type tend to have larger impact. Fig. 1(a) shows the Cumulative Distribution Function (CDF) of the percentage of polluted ASes in our simulations. The farther the position of the hijacker in the announced path (*i.e.*, as the hijack type increases from 0 to 4), the lower the probability that a hijack can affect a large fraction of the Internet. For hijack types larger than Type-2, in the majority of the cases ($> 50\%$) their impact is very limited or negligible (*e.g.*, 4% and 1% for Type-3 and Type-4, respectively).

All types of hijacks can have a large impact. Comparing the *mean* to the *median* values in Fig. 1(b) (blue curves; circle markers) highlights that even with Type-4 hijacks there are events with a large (*i.e.*, $> 80\%$, see Fig. 1(a)) impact. We verified that these corner cases happen not because the hijacker AS has high connectivity, but because of the reciprocal location of the hijacker and victim ASes in the AS-graph and the respective relationships with their neighbors. Hence, network connectivity metrics alone [41], cannot always (*i.e.*, for all attack types) indicate the potential impact (in terms of Internet pollution) of an attacking AS. Since it is difficult to identify the ASes [5] that are capable of launching impactful hijacking attacks (*e.g.*, using the methodology of [55] would require to consider all possible hijacker ASes and attack types), *an operator should be able to defend their networks against every type of hijacking event*.

4.2 Visibility of Hijacks on the Control Plane

Here we study to which extent different types of hijacks are visible by monitors of publicly accessible BGP monitoring infrastructure. Detecting a hijacking event through control-plane monitoring requires the illegitimate path to propagate

5. Ballani *et al.* [20] use simulation to estimate the probability of impact of hijacking attacks against different ASes in the AS graph. They show that besides ASes high in the routing hierarchy, even small ASes can hijack and intercept traffic from a non-negligible fraction of ASes, making identification of attackers challenging.



(a) all monitors (b) streaming monitors

Fig. 2: *Visibility* of different hijack types: CCDFs of the number of monitors that observe an illegitimate route over 1000 simulations for different types, using (a) *all* and (b) *streaming* monitoring services. Hijacking events of smaller type are visible with higher probability and to more monitors.

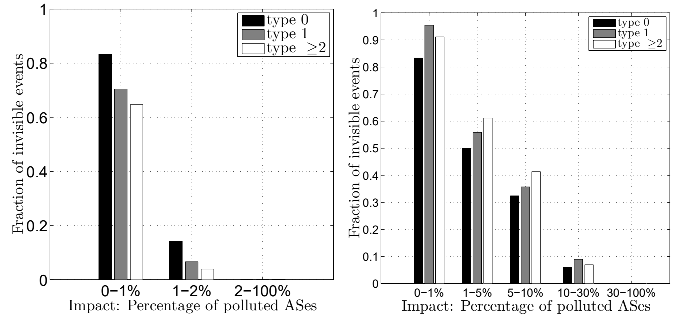
to at least one monitor. Moreover, the more monitors receive such a route, the faster and more robust (*e.g.*, against monitor failures) the detection of a hijack is.

Hijacking events of smaller AS-path type are more visible. Fig. 2 shows the distribution of the number of monitors, from *(a)* all monitoring services, and from *(b)* only RIPE RIS and BGPmon streaming services, that receive an illegitimate path. As expected, hijacking events of smaller type are visible with higher probability and to more monitors (on average), since their impact on the Internet is larger (see Fig. 1(b)). Table 3 gives the percentage of hijacking events that are *invisible* to the different services (*i.e.*, they do not pollute any of the monitors in our simulations). We can see that almost all origin-AS hijacks (Type-0) are visible, whereas hijacks of types 1, 2, 3, and 4 have a higher probability to remain unnoticed, *e.g.*, more than 20% of Type-3 hijacks are not visible by any service. We also find that the combination of different services always leads to increased visibility.

Hijacking events (of every type) with significant impact are always visible to monitoring services. Fig. 3 shows the fraction of hijacking events, grouped by their impact, that are invisible to monitoring services. Hijacking events that pollute more than 2% of the Internet are –in our simulations– always visible to the monitoring services (Fig. 3(a)), and the vast majority (*e.g.*, more than 85% type-0 hijacks) of those with impact between 1% and 2% are also observed. The visibility is low only for events with impact less than 1% when considering all monitors. In total, the mean (median) impact of invisible events is less than 0.2%

TABLE 3: Percentage of *invisible* hijacking events. Hijacks of higher types tend to pollute a smaller portion of the Internet. Combining monitoring services always increases visibility.

	Hijack type				
	0	1	2	3	4
BGPmon (stream)	10.9%	31.6%	53.6%	65.9%	76.1%
RIPE RIS (stream)	7.1%	20.6%	36.7%	50.5%	63.8%
All stream services	4.2%	15.6%	33.1%	47.8%	62.2%
RouteViews	1.5%	4.3%	11.1%	26.5%	38.0%
RIPE RIS	1.8%	4.0%	13.8%	26.4%	40.9%
All services	1.4%	3.0%	9.0%	21.3%	34.4%



(a) all monitors (b) streaming monitors

Fig. 3: Fraction (y-axis) of the hijacking events, grouped by impact (x-axis), that are invisible to (a) all monitoring services, and (b) streaming monitoring services, for different hijack types (denoted as bars of different colors). *Note the differences in the (a)/(b) x-axis.* Existing monitoring infrastructure can always observe hijacking events of significant impact.

(0.1%) as shown in Fig. 1(b). These results suggest that existing infrastructure has already a great potential to enable live detection of significant hijacking events. We find instead that current streaming services have full visibility only for events with impact greater than 30% (Fig. 3(b)), highlighting the potential benefit from RIPE RIS and RouteViews accelerating their transition to live streaming [16], [23].

These findings deliver a promising message for using public BGP monitoring infrastructure to detect hijacks: while a hijacker can employ several means to achieve a stealthy hijack (*e.g.*, to launch Type-N attacks of large N, or append BGP communities to limit its visibility within specific regions), the attack can only be invisible at the cost of limited impact.

5 Detection Methodology

5.1 Overview

ARTEMIS is run locally by a network and enables a self-operated (*i.e.*, not involving third parties) detection of hijacking events for its own prefixes. ARTEMIS *(a)* uses a *local configuration file* with information about the prefixes owned by the network, and *(b)* receives as input the *stream of BGP updates* from the publicly available monitoring services and the local routers of the network that operates it. Comparing the prefix and AS-PATH fields in the BGP updates with the information in the local configuration file, ARTEMIS can detect any class of hijacking event, and generate alerts.

The local configuration file is populated by the network operator, and includes lists of owned ASNs and prefixes, ASNs of neighboring ASes, and routing policies (*e.g.*, "prefix p is announced with origin AS_O to neighbors AS_{n1} and AS_{n2}"). For ease of use, the local configuration file can be populated (and updated) automatically; for instance, ARTEMIS can communicate with the BGP routers of the network (with iBGP, ExaBGP [28], or route reflectors).

False Positives (FP) and Negatives (FN). Table 4 summarizes the FP–FN performance of the different detection criteria used in our approach for each attack scenario (discussed in § 5.2, § 5.3, § 5.4). By default, our approach does not introduce FN for any attack scenario. The only possible FN are the events not *visible* by the monitoring infrastructure (§ 4.2), which have very limited impact on the control plane (Figs. 1(b) and 3). We generate potential FP (at a very low rate) only for exact-prefix hijacking events of Type-N, $N \geq 2$; however, for the detection of this class of events, ARTEMIS optionally allows the operator to trade *(i)* speed for increased accuracy, and *(ii)* potential FN related to events with negligible visible impact (*e.g.,* seen by only 1 monitor) for less FP.

5.2 Detecting Sub-prefix Hijacks & Squatting

Sub-prefix hijacks are the most dangerous, since they can pollute the entire Internet due to the longest prefix matching employed by the BGP decision process. They are also among the most problematic when using third-party services, since each time an AS decides to announce a longer prefix or to de-aggregate a prefix, it either needs to communicate this information in advance to the third-party service or it will receive a false-positive alert from it. For this reason, often sub-prefix detection is not even implemented/enabled (§ 8.1).

ARTEMIS returns 0 false positives and 0 false negatives for *all* sub-prefix hijacking events — independently of the Type being 0, 1, 2, ... To detect these events, the network operator stores in the *local configuration file* of ARTEMIS an up-to-date *list of all owned and announced prefixes*. When a sub-prefix hijack takes place, the monitoring services observe BGP updates for this sub-prefix (the entire Internet is polluted), and ARTEMIS immediately detects it, since the sub-prefix is not included in the list of announced prefixes. Such a detection becomes trivial with our approach (*i.e.,* leveraging local information). However, this is an important result: without this detection in place, attackers can remain stealthy by announcing a sub-prefix, which allows them to avoid announcing an illegitimate `AS-PATH` (and can further increase stealthiness by carrying the attack on the data plane as a Man-in-the-Middle [52]). In the following sections we illustrate how ARTEMIS detects the remaining classes of attacks when *exact-prefix* hijacking is involved instead.

ARTEMIS returns 0 false positives and 0 false negatives for *all* BGP squatting events. Checking against the operator's list of actually announced prefixes, has the added benefit of detecting *BGP squatting* as well; a technique commonly used by spammers, in which a (malicious) AS announces space owned but not announced by another AS [46], [58].

5.3 Detecting Type-0/1 Exact Prefix Hijacks

The network operator provides also in the *local configuration file* the following information *per prefix*:

- *Origin ASN(s)*: the ASNs authorized to originate the prefix.
- *Neighbor ASN(s)*: the ASNs with which there are direct BGP sessions established, where the prefix is announced.

For every BGP update it receives from the monitors, ARTEMIS extracts the `AS-PATH` field, and compares the announced prefix, as well as the first and second ASNs in the `AS-PATH`, with the {prefix, origin ASN, neighbor ASN} information in the local file. If the `AS-PATH` does not match the information in the local file, a hijack alert is generated.

ARTEMIS detects all Type-0 and Type-1 hijacks that are visible to the monitors (*i.e.,* 0 false negatives for visible events). As in § 5.2, since ARTEMIS leverages *ground truth* provided by the operator itself, all illegitimate paths that are visible by the monitors are always detected as hijacks.

ARTEMIS returns 0 false positives for Type-0/1 hijacking events. Any BGP update that does not match the local lists {prefix, origin ASN, neighbor ASN}, indicates *with certainty* an announcement originated illegitimately by another network (*i.e.,* without the consent of the prefix owner).

5.4 Detecting Type-N, N≥2, Exact Prefix Hijacks

Detecting Type-N, $N \geq 2$, hijacking events requires a different approach than Type-0/1 events, since the operator might not be aware of all its 2^{nd}, 3^{rd}, ... hop neighbors. To this end, ARTEMIS *(i)* detects all suspicious Type-N, $N \geq 2$, events, *i.e.,* when new links[6] appear in routes towards the operator's prefixes, *(ii)* filters out as many legitimate events as possible, and *(iii)* augments alerts with information about the estimated impact of the remaining suspicious events.

Specifically, ARTEMIS uses a configurable two-stage detection approach, where the operator can trade detection speed (*Stage 1*) for increased accuracy and impact estimation (*Stage 2*). *Stage 1* detects all potential hijacking events as soon as they are observed by a monitor (*i.e.,* typically with few seconds latency), filters out benign events based on information that is available at detection time, and generates alerts for suspicious events. An optional *Stage 2* collects additional information within a (configurable) time window T_{s2} following the detection from *Stage 1*, in order to (a) increase the chance of filtering out a benign event, and (b) provide the operator with an estimate of the impact of the event in case it is still recognized as suspicious.

5.4.1 *Stage 1*

For Type-N, $N \geq 2$, detection, ARTEMIS stores locally the following lists of *directed* AS-links (with related metadata):

- *previously verified AS-links list*: all the AS-links that appear in a path towards an owned prefix and have been verified by ARTEMIS in the past.
- *AS-links list from monitors*: all the AS-links in the AS-path towards *any* prefix (*i.e.,* owned by any AS) observed by the monitors, in a sliding window of the last 10 months. This list represents an historical view of observed (directed) AS-links. The 10-month time frame should accommodate the observation of most of the backup routes [24].
- *AS-links list from local BGP routers*: all the AS-links observed in the BGP messages received by the BGP routers of the network operating ARTEMIS. The list is collected by connecting to the local BGP routers (*e.g.,* via ExaBGP [28] or with BGPStream and BMP [22], [61]), and receiving every BGP update seen at them, or alternatively querying

6. We consider only new links and not policy violations on existing links (as, *e.g.,* [63] [53]), since routing policies are not publicly available, and inferences based on existing datasets would lead to a very high number of false alerts; *e.g.,* [18] shows that around 30% of the observed routes are not in agreement with the available routing policy datasets.

TABLE 4: Detection of the different BGP prefix hijacking attacks by ARTEMIS.

Hijacking Attack			ARTEMIS Detection				
Prefix	AS-PATH (Type)	Data Plane	False Positives (FP)	False Negatives (FN)	Detection Rule	Needed Local Information	Detection Approach
Sub-prefix	*	*	None	None	Config. vs BGP updates	Pfx.	Sec. 5.2
Squatting	*	*	None	None	Config. vs BGP updates	Pfx.	Sec. 5.2
Exact	0/1	*	None	None	Config. vs BGP updates	Pfx. + ASN (+ neighbor ASN)	Sec. 5.3
Exact	≥ 2	*	< 0.3/day for $> 73\%$ of ASes	None	Past Data vs BGP updates (bidirectional link)	Pfx.+ Past AS links	Sec. 5.4 *Stage 1*
Exact	≥ 2	*	None for 63% of ASes ($T_{s2} = 5min$, $th_{s2} > 1$ monitors)	$< 4\%$	BGP updates (waiting interval, bidirectional link)	Pfx.	Sec. 5.4 *Stage 2*

a route server. This list is also updated continuously within a 10-month sliding data window.

The detection algorithm is triggered when a monitor receives a BGP update (for a monitored prefix) whose AS-PATH contains an N-hop ($N \geq 2$) AS-link that is not included in the *previously verified AS-links list*. Let AS_V be the victim AS (operating ARTEMIS), the new AS-link be between ASes AS_X and AS_Y, and the AS-PATH of the BGP update be

$$\begin{aligned} P^{new}_{(X,Y)} &= \{AS_{\ell 1}, AS_{\ell 2}, ..., AS_X, AS_Y, AS_{r1}, AS_{r2}, ..., AS_V\} \\ &= \{\mathcal{L}^{new}, AS_X, AS_Y, \mathcal{R}^{new}, AS_V\} \end{aligned}$$

where $\mathcal{L}^{new} = \{AS_{\ell 1}, AS_{\ell 2}, ...\}$ denotes the set of ASes appearing in the path after (left of) the suspicious link, and $\mathcal{R}^{new} = \{AS_{r1}, AS_{r2}, ...\}$ before (right of) the suspicious link. Note that the type of the attack is $N = 2 + |\mathcal{R}^{new}|$. The observation of $P^{new}_{(X,Y)}$ is considered as a suspicious event (and previous works would raise an alarm [53], [63]). However, it is possible that $P^{new}_{(X,Y)}$ corresponds to a legitimate event (*e.g.*, change of a routing policy) that made the link $AS_X - AS_Y$ visible to a monitor. To decrease the number of false alarms, ARTEMIS applies the following filtering rules.

Rule 1 (bi-directionality). Check if the new link $AS_X - AS_Y$ has been observed in the opposite direction (*i.e.*, $AS_Y - AS_X$) in the *AS-links list from monitors* and/or *AS-links list from local BGP routers*. If the reverse link $AS_Y - AS_X$ is *not* previously observed, the event is labeled as suspicious.

Rule 2 (left AS intersection). Otherwise (*i.e.*, the reverse link $AS_Y - AS_X$ is previously observed), check the AS paths in all the BGP updates containing the reverse link. Let $\mathcal{P}^{old}$ be the set of all these AS-paths, and denote

$$P = \{\mathcal{L}_P, AS_Y, AS_X, \mathcal{R}_P\} \quad , \quad \forall P \in \mathcal{P}^{old}$$

Then, collect all the sets of ASes $\mathcal{L}_P, \forall P \in \mathcal{P}^{old}$, that appear after (left of) the reverse link, and calculate the intersection of all these sets, i.e., $\mathcal{L}^{old} = \bigcap_{P \in \mathcal{P}^{old}} \mathcal{L}_P$. If $\mathcal{L}^{old}$ is not empty, and at least one AS in $\mathcal{L}^{old}$ appears also in $\mathcal{L}^{new}$ (*i.e.*, $\mathcal{L}^{old} \bigcap \mathcal{L}^{new} \neq \emptyset$) in the new received path $P^{new}_{(X,Y)}$, then the event is labeled as suspicious. If $\mathcal{L}^{old} \bigcap \mathcal{L}^{new} = \emptyset$, the event is labeled as legitimate.

ARTEMIS uses these two filtering rules to identify suspicious announcements of fake links that either contain the attacker's ASN (*Rule 1*) or do not (*Rule 2*). The rationale behind the two rules is detailed in the following.

Rule 1 detects events where the hijacker (*e.g.*, AS_X) is one end of the fake link. While AS_X can fake an *adjacency* with AS_Y, and the link $AS_X - AS_Y$ appears in the polluted routes, the reverse link (*i.e.*, $AS_Y - AS_X$) is not advertised by AS_Y or other networks, and thus not seen by any monitor. It is impossible for an attacker *controlling a single AS* to make such a fake link appear in both directions in order to evade the detection of *Rule 1*[7]. Hence, observing an AS-link $AS_X - AS_Y$ in both directions, eliminates the possibility that AS_X advertises a fake adjacency. On the contrary, observing a new link in only one direction cannot guarantee a legitimate announcement and thus causes ARTEMIS to raise an alert.

Rule 1 can be evaded only if the hijacker *(i) controls at least two ASes*, or *(ii)* announces a fake link not containing its ASN. While the former case violates our threat model and is out of the scope of the paper, we apply *Rule 2* to detect the latter case. For instance, a hijacker AS_Z can announce to its neighboring ASes two paths containing a fake link $AS_X - AS_Y$ in both directions:

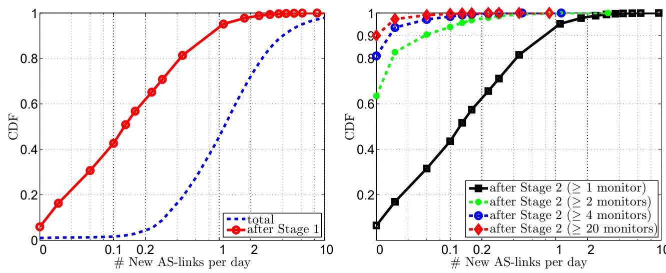
$$\begin{aligned} P_1 &= \{AS_Z, ..., AS_X, AS_Y, ...\} \\ P_2 &= \{AS_Z, ..., AS_Y, AS_X, ...\} \end{aligned}$$

However, in its announcements, the hijacker has to append its ASN as the last (leftmost) AS in the path, before further propagation (see § 2.1 and RFC4271 [35]). Hence, in all BGP updates containing the fake link $AS_X - AS_Y$ in any direction, the AS of the hijacker will appear on the left of the fake link. *Rule 2* identifies whether there exists a common AS in all (new and old) announcements involving any direction of the new (suspicious) link. If at least one AS appears in all paths, then the event is considered suspicious.

ARTEMIS's *Stage 1* returns 0 false negatives. ARTEMIS detects any illegitimate announcement that is seen by the monitors and contains a fake link with (*Rule 1*) or without (*Rule 2*) the hijacker ASN at its ends. It is not possible for an attacker conforming to the threat model of § 2 to evade these rules, as long as its announcements are visible.

The ARTEMIS detection algorithm for Type-N, $N \geq 2$, hijacks, is rarely triggered. To understand how often the detection algorithm would be triggered, we ran our algorithm on 1 month of real BGP data, emulating running ARTEMIS for each and every AS announcing prefixes on the Internet. Specifically, we processed all the BGP updates observed by RIPE RIS and RouteViews monitors (a total of 438 ASes hosting at least 1 monitor each) between April 2016 and March 2017. Then, for each AS that originated

7. The only way for AS_X to announce a path containing $AS_Y - AS_X$ is to announce a path with a loop (*e.g.*, $\{AS_X, ..., AS_Y, AS_X, ...\}$), but ARTEMIS detects and discards announcements with loops instead of adding them to the *AS-links list from monitors* list.



(a) Effects of Stage 1 (b) Effects of Stage 2

Fig. 4: CDF of the number of new AS-links seen at the monitor AS per day, per origin AS: (a) before and after applying Stage 1 - ARTEMIS detection algorithm for Type-N, $N \geq 2$, is rarely triggered and *Stage 1* dramatically reduces the number of FP; (b) after applying *Stage 2* ($T_{s2} = 5$ min), with different thresholds for the minimum number of monitors that see the suspicious event - requiring at least 2 (or more) monitors to see the event, greatly reduces the number of FP.

IPv4 or IPv6 prefixes in March 2017, we identified the links appearing for the first time in paths towards their originated prefixes, during the same month. Fig. 4(a) shows the CDF (blue/dashed curve) of the number of new AS-links an origin AS sees (through the monitor ASes) per day towards its own prefixes: on average, within the month of March 2017, 72% of the origin ASes saw less than 2 new links per day.

***Stage 1* dramatically reduces the number of suspicious events**. We apply the filtering of *Stage 1* to the previous data; we considered only the *AS-links list from monitors* (since we do not have access to the local routers of all the ASes). Fig. 4(a) shows the CDF of the number of the aforementioned events that fail *Stage 1* (red/circles curve): 73% of the origin ASes see less than 1 suspicious event every 3 days.

5.4.2 Stage 2 (optional)

Stage 2 introduces an extra delay (T_{s2}) in exchange for *(i)* refined filtering and *(ii)* the ability to estimate the impact of a suspicious event. To improve filtering of legitimate events, we check if at the end of the T_{s2} period, the new link has appeared in the opposite direction in the BGP updates received from the monitors and/or local routers. In other words, if the new link really exists, then it is probable that it is used also in the opposite direction and a route (containing the opposite direction) will propagate to a monitor or a local router after some time. The waiting interval T_{s2} can be configured by the operator (speed/accuracy trade-off); here, we select $T_{s2} = 5$ minutes, which is enough time for the best BGP paths to converge on most of the monitors [37].

***Stage 2* allows ARTEMIS to further reduce alerts for Type-N, $N \geq 2$, events.** The black curve (square markers) in Fig. 4(b) shows the CDF of the number of events detected as suspicious at the end of *Stage 2* when using the public monitors (RouteViews and RIPE RIS), but not local routers. The improvement only from public monitors is around 1%.

However, considering also the local monitors and the impact of the events, significantly increases the gains from Stage 2, as we discuss in the remainder.

Local routers see significantly more links in the opposite direction than monitors, thus further improving the filtering of *Stage 2*. Using in *Stage 2* the *AS-links list from local BGP routers* as well, would further reduce suspicious events. We investigate this effect through simulation: we introduce a new link in the topology, and after BGP convergence we check whether the new link is seen in the opposite direction by the local routers. Our results show that the *local BGP routers* see the opposite direction of the new link in around 25% (2nd-hop) and 30% (3rd-hop) of the cases, *i.e.,* thus filtering 1-2 orders of magnitude more Type-2 and Type-3 suspicious events compared to the case of using only the *AS-links list from monitors*. This rich information that exists locally, highlights further the gains from the self-operated approach of ARTEMIS.

***Stage 2* provides an estimate of the impact of the suspicious event.** Waiting for BGP convergence allows Stage 2 to further discover how many monitors see the Type-N suspicious event (*i.e.,* the new suspicious link in a route towards the operator's prefix) and, therefore, estimate the extent of the "pollution" in case the event is a hijack. When Stage 2 is enabled, ARTEMIS uses this information to trigger different alert modes and mitigation strategies based on the configuration provided by the operator (§ 6).

***Stage 2* –optionally– allows the operator to almost eliminate false positives at the expense of a few false negatives of negligible control-plane impact.** The impact ("pollution") estimate of *Stage 2* can also be used to further reduce false positives, by raising an alert only if the number of monitors seeing the event is above a (user-selected) threshold. In this way, ARTEMIS can completely ignore a large number of uninteresting events (*e.g.,* legitimate changes in routing policies that appear as new links) at the expense of potentially introducing false negatives that have negligible visible impact on the control plane. This is demonstrated in Fig. 4(b), which shows that the majority of the suspicious events we observe in the Internet (same experiment as in Fig. 4(a)) are seen by only *a single* monitor.

Specifically, according to our experiment in Fig. 4(b) (see x-axis for $x \rightarrow 0$), by ignoring all new links observed at only one monitor, *Stage 2* would have generated *at most one* (or, *zero*) alert in the whole month of March 2017 for 83% (63%) of the origin ASes (green curve). Increasing the threshold further decreases alerts: if the operator decides to ignore events seen by less than 4 monitors (blue curve) then the percentage of origin ASes without at most one (zero) alerts reaches 94% (81%), and for a threshold of 20 monitors (red curve) it is 97% (90%). Finally, Fig. 2(a) provides an indication of the rate of potential false negatives this threshold would yield: *e.g.,* for Type-2 hijacks and a threshold of at least 2 monitors, the percentage of false negatives (*i.e.,* percentage of hijacks with negligible visible impact on the control plane, seen by exactly one monitor) would be less than 4%.

6 Mitigation Methodology

Ultimately, a network operator needs to quickly mitigate a hijacking event. To this end, a timely detection is not the only *necessary* condition. Low false positives, information about the event (*e.g.*, estimated impact, relevance of the affected prefix), and an automated system are also key requirements. In this section, we present the ARTEMIS unified approach for detection and mitigation, which satisfies all these conditions, and enables a configurable and timely mitigation.

6.1 ARTEMIS Mitigation Approach

ARTEMIS provides an informative detection of hijacking events that enables automated and fast mitigation. The ARTEMIS detection module can provide the following information –as output– for each detected event:

1) affected prefix(es);
2) type of the hijacking event;
3) observed impact (*e.g.*, number of polluted monitors);
4) ASN(s) of the AS(es) involved in the event;
5) confidence level (reliability) of the detection.

Note that a detection is always accurate (no false positives; confidence level = "certainty") for any type of sub-prefix hijacking events (*cf.*, § 5.2) and for exact-prefix Type-0 and Type-1 hijacking events (*cf.*, § 5.3), *i.e.*, the events with the highest impact on the control plane. In contrast, the confidence level of an exact-prefix Type-N, $N \geq 2$, hijacking event can be quantified by the result of the detection *Stages 1/2* (§ 5.4) and allows ARTEMIS to classify an event as more or less suspicious (*e.g.*, confidence level = "alert by Stage 1" and/or "alert by Stage 2").

This rich information is sufficient in most cases for an operator to decide how to configure the network's reaction to a hijacking or suspicious event. As a result, ARTEMIS enables the automation of mitigation: (i) the operator preconfigures ARTEMIS (mitigation module) to proceed to different mitigation actions based on the detection output; for instance, the following mapping could be used[8]:

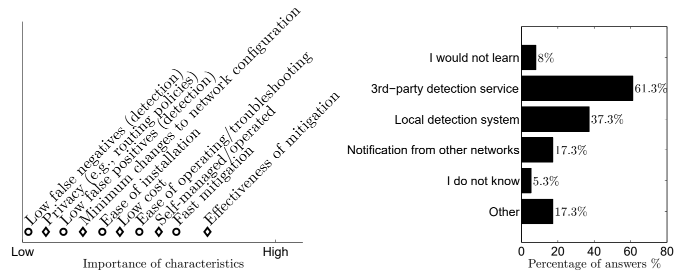
$$\{\text{Prefix, Impact, Confidence level}\} \rightarrow \text{Mitigation action;}$$

(ii) ARTEMIS executes the pre-selected action *immediately* after the detection of an event, not requiring manual actions.

Examples of applying this approach are: (a) the operator selects to handle an event of limited impact (squatting, few polluted monitors, etc.) manually instead of triggering an automated mitigation process; (b) for sensitive prefixes (*e.g.*, web-banking), the operator selects to always proceed to mitigation (*e.g.*, even for low-confidence alerts for Type-N$\geq$ 2 hijacks), since the cost of potential downtime (or even compromise in the case of traffic interception attacks) is much higher than the mitigation cost for a false alert.

ARTEMIS satisfies operators' needs and outperforms current practices. We conducted a survey among 75 network operators (see details in § 8.1) that shows that the majority of networks rely on third parties for detecting hijacks against their own prefixes (Fig. 5(b)): 61.3% outsource detection to services such as [8], and 17.3% expect to be notified by colleagues or mailing lists. However, the employment of third parties may lead to false alerts, delayed (inferred) detection and thus delayed mitigation (§ 8.1). In contrast, ARTEMIS provides a reliable and fast detection that also enables fast mitigation, which is one of the main concerns of operators (cf. "Fast mitigation" - Fig. 5(a)). Moreover, self-operated approaches like ARTEMIS are highly desirable (cf. "Self-managed/operated" - Fig. 5(a)); we believe that its characteristics (lightweight, no cost for public monitoring services, flexible and configurable) render it ideal for – at least – two thirds of the networks not currently employing any local detection system (Fig. 5(b)).

8. In this example, the hijack type and hijacker's ASN are wildcards. In a more specific mapping, all five fields of the information presented above could be distinctly used.



(a) (b)

Fig. 5: Survey results: (a) ranking of characteristics of a hijacking defense system, based on their importance, by network operators; (b) practices for detecting/learning about hijacking incidents against owned prefixes.

In the following section, we focus on the crucial aspect of the mitigation effectiveness (Fig. 5(a)). We study and propose mitigation techniques that build on current practices and can be incorporated in the ARTEMIS approach.

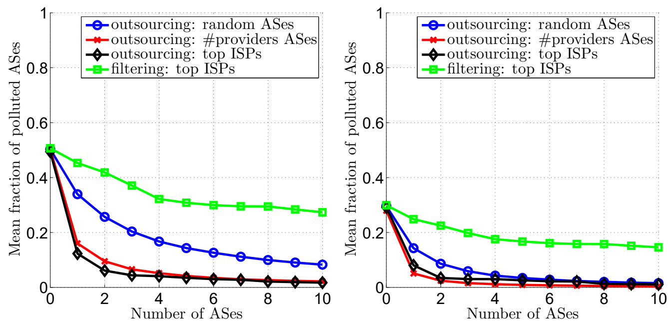
6.2 Mitigation Techniques

We propose two mitigation techniques that can be used with ARTEMIS (other techniques could work as well). Specifically, the victim AS can counteract a hijack with its own resources by *deaggregating* the hijacked prefix (Section 6.2.1), or *outsource* the mitigation to a third party organization, which will announce the prefix on behalf of the victim to reduce the impact of the hijack (Section 6.2.2).

6.2.1 Self-operated mitigation with prefix deaggregation

After receiving an alert for an ongoing hijacking event, operators replied in our survey that they would react by *contacting other networks* (88% of the participants) and/or *deaggregating the affected prefix* (68% of the participants). While the former action involves a significant delay (up to many hours, or even days [48]), the latter can be automated and applied immediately after the detection step using the ARTEMIS approach.

Prefix deaggregation is the announcement of the more specific prefixes of a certain prefix. For example, upon the detection of a hijack for the prefix *10.0.0.0/23*, the network can perform prefix deaggregation and announce two more-specific sub-prefixes: *10.0.0.0/24* and *10.0.1.0/24*. These sub-prefixes will disseminate in the Internet and the polluted ASes will re-establish legitimate routes, since more-specific



(a) Hijack type 0

(b) Hijack type 1

Fig. 6: Efficiency of mitigation via *outsourcing BGP announcements* to organizations selected (i) randomly, and based on their (ii) number of providers and (iii) customer cone (top ISPs), and via (iv) *filtering* at top ISPs. Outsourcing mitigation -even to a single organization- is very effective and significantly outperforms current practices (filtering).

prefixes are preferred by BGP. Prefix deaggregation is *effective* for */23* or less-specific (*/22*, */21*, ...) hijacked prefixes (since */25* prefixes and more-specifics are filtered by most routers [21]). Moreover, it can be operated by the network itself without any added cost. The automation of prefix deaggregation over ARTEMIS is simple, *e.g.*, using ExaBGP [28] or custom scripts that are triggered immediately after the detection. A potential mapping could be:

$$\{\text{Prefix length} < /24, *, *\} \rightarrow \text{Deaggregation}$$

where * denote wildcards (*i.e.*, any impact/confidence level).

To mitigate hijacking events involving */24* prefixes, in the following we examine alternative mechanisms, which require the involvement of additional networks besides the one operating ARTEMIS.

6.2.2 Outsourcing mitigation with MOAS announcements

It is common practice for networks to outsource various security services to a single (or a few) third-party organization(s). A prominent example is the DDoS mitigation service offered by organizations to networks that are unable to handle large volumetric DDoS attacks [19]. Moreover, 39% of the participants in our survey do not reject the possibility to outsource hijacking mitigation. We also expect that the higher level of accuracy offered by ARTEMIS and the per-prefix configurability would make more operators consider outsourcing mitigation, when triggering it is under their control (*e.g.*, allowing them to carefully manage the cost *vs* security risk trade-off). We thus propose a mitigation technique that presents several analogies with the current practice of DDoS mitigation services, and study its efficiency.

Outsourcing BGP announcements is similar to the outsourced DDoS protection security model, where the organizations that mitigate the attacks redirect the traffic (using BGP/MOAS or DNS) to their locations and scrubbing centers, remove malicious traffic, and forward/relay the legitimate traffic to the victim. In the case of BGP hijacking, the mitigation organization receives a notification from the network operating ARTEMIS, and immediately announces from their location/routers the hijacked prefix. In this way, the organization attracts traffic from parts of the Internet and then tunnels it back to the legitimate AS (through, *e.g.*, MPLS tunnels, direct peering links, or its upstream providers). The automation of this process could be implemented, *e.g.*, with ARTEMIS-triggered MOAS on the control plane and traffic tunneling on the data plane; a corresponding mapping in ARTEMIS (potentially only for the most security-sensitive prefixes owned by the organization) could thus be:

$$\{\text{Prefix length} = /24, *, *\} \rightarrow \text{Outsource BGP announcements}$$

More than one external organization can be employed for more effective mitigation. In the following, we investigate the efficiency of this technique for different *selection criteria* and *number* of mitigation organizations. In Fig. 6 we present simulation results for the remaining number of polluted ASes (y-axis) after announcing the prefix from different numbers of mitigation organizations (x-axis) in addition to the network operating ARTEMIS. We consider three cases where we select the outsourcing organizations (i) *randomly*, and based on their (ii) *number of providers* (which correlates with their mitigation efficiency [41]) and (iii) *customer cone* (“top ISPs”) that corresponds to large ISPs [6].

Outsourcing mitigation even to a single organization is very effective, and significantly reduces the impact of hijacking. Fig. 6(a) shows that outsourcing BGP announcements to the top ISPs outperforms a selection of ASes with many providers, while randomly selecting organizations is always less efficient. However, even a single randomly selected organization can considerably reduce the impact of the hijacking event (on average), from 50% to 34% and from 28% to 14% for Type-0 (Fig. 6(a)) and Type-1 (Fig. 6(b)) events, respectively, which clearly indicates an effective and robust mitigation technique. Outsourcing to more than one organization simultaneously and/or carefully selecting the mitigation organization can further increase the mitigation benefits, *e.g.*, leading to less than 5% polluted ASes (one order of magnitude lower compared to the initial impact) with only 3 top ISPs for Type-0 events.

Outsourcing BGP announcements outperforms current practices. In Fig. 6 we compare the efficiency of outsourcing against *prefix filtering*, a proactive defense that needs cooperation of networks and is currently partially deployed (§ 8.1). We consider filtering of the illegitimate routes from the top ISPs; while filtering applies to origin-AS hijacks today, in Fig. 6(b) we assume a potential filtering for Type-1 hijacks as well. Our results show that filtering is much less efficient than outsourcing BGP announcements: even with 10 filtering ASes, the mitigation efficiency is almost equal to (Fig. 6(a)) or not better than (Fig. 6(b)) using a single randomly selected outsourcing AS. Increasing the number of filtering ASes to a few dozens, barely helps.

Existing industry security models can provide highly effective outsourced mitigation. In Table 5, we present the hijacking mitigation efficiency of different organizations that currently provide DDoS protection services. We selected, as

TABLE 5: Mean percentage of polluted ASes, when outsourcing BGP announcements to organizations providing DDoS protection services; these organizations can provide highly effective outsourced mitigation of BGP hijacking.

	without outsourcing	top ISPs	AK	CF	VE	IN	NE
Type0	50.0%	12.4%	2.4%	4.8%	5.0%	7.3%	11.0%
Type1	28.6%	8.2%	0.3%	0.8%	0.9%	2.3%	3.3%
Type2	16.9%	6.2%	0.2%	0.4%	0.4%	1.3%	1.1%
Type3	11.6%	4.5%	0.1%	0.4%	0.3%	1.1%	0.5%

examples, 5 organizations of varying sizes[9] and simulated BGP announcements originating from them for the hijacked prefix. Mitigation with any of them is efficient, outperforming even top ISPs. Specifically, mitigation from Akamai is the most efficient, reducing the percentage of polluted ASes to 2.4% (from 50% originally) on average for Type-0 hijacks. This holds also for the other hijack types, where the average percentage of polluted ASes is reduced to 0.3% or less.

7 Real-World Experiments

We setup and conduct *real* BGP prefix hijacking experiments in the Internet (§ 7.1) using the PEERING testbed [10], [60]. We implemented a prototype of ARTEMIS, which we use to detect and mitigate the hijacking events, and study the actual *detection and mitigation times* observed (§ 7.2).

7.1 Experimental Setup

ARTEMIS prototype. The current prototype implementation of ARTEMIS interacts with the streaming services through the RIPE RIS `socket.io` API and `telnet` for BGPmon. It receives streams of BGP updates (formatted in plain text from RIPE RIS and XML format from BGPmon), and keeps/filters only the BGP updates concerning the network-owned prefixes. CAIDA's BGPStream will soon support reading from multiple streaming data sources simultaneously [22], [51] (including RIPE RIS `socket.io` and BMP feeds, which RouteViews and others plan to make available at the same time). We envision replacing the BGP feed interface of our ARTEMIS implementation using CAIDA's BGPStream API.

Testbed. PEERING [10], [60] is a testbed that connects with several real networks around the world, and enables its users to announce routable IP prefixes from real ASNs to the rest of the Internet; the IP prefixes and ASNs are owned by PEERING, hence, announcements do not have any impact on the connectivity of other networks.

In our experiments, we use the connections to three real networks/sites (Table 6; data of Jun. 2017) that provide transit connectivity to PEERING, which we select due to their Internet connectivity characteristics. GRN and ISI resemble the connectivity of a typical small ISP in the real Internet, while AMS resembles a large ISP. We are granted authorization to announce the prefix *184.164.228.0/23* (as well as its two */24* sub-prefixes), and use the AS numbers *61574* for the legitimate AS, *61575* for the hijacker AS, and *61576* for the outsourcing AS.

9. Namely: Akamai (AK; ASNs: 20940, 16625), CloudFlare (CF; ASN: 13335), Verisign (VE; ASNs: 26415, 30060, 7342, 16838), Incapsula (IN; ASN: 19551), and Neustar (NE; ASNs: 7786, 12008, 19905).

TABLE 6: PEERING sites used in the experiments.

ID	Network	Location	ASNs (transit)	#peers (IPv4)
AMS	AMS-IX	Amsterdam, NL	12859, 8283	74
GRN	GRNet	Athens, GR	5408	1
ISI	Los Nettos	Los Angeles, US	226	1

Methodology. Using the aforementioned ASNs, we create three virtual ASes in PEERING: (i) the legitimate (or victim) AS, (ii) the hijacker AS, and (ii) the outsourcing AS. For each experiment, we connect each virtual AS to a different site/network of Table 6, and proceed as follows.

1. *Legitimate announcement.* The legitimate (victim) AS announces the /23 IP prefix at time t_0, using ARTEMIS to monitor this prefix for potential hijacking events.
2. *Hijacking Event.* The hijacker AS hijacks (*i.e.*, announces) the /23 IP prefix at time $t_h = t_0 + 20min$.
3. *Detection.* When a hijacked (illegitimate) route arrives at a monitor, ARTEMIS detects the event at a time t_d ($> t_h$), and immediately proceeds to its mitigation.
4. *Mitigation.* The legitimate AS announces the /24 sub-prefixes (*deaggregation*), or the outsourcing AS announces the /23 prefix (MOAS announcement) at time t_m ($t_m \approx t_d$).

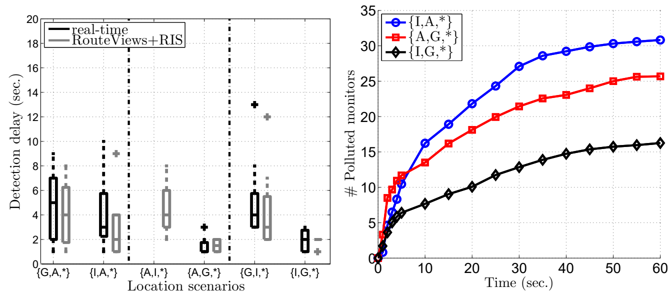
Scenarios. We conduct experiments in several scenarios of different hijacking and mitigation types, considering all combinations of the following parameters:

- *Location* (*i.e.*, connection to PEERING sites) of the legitimate, hijacker, and outsourcing ASes.
- *Hijacking event types*: 0 (origin-AS), 1, and 2.
- *Mitigation* via deaggregation or MOAS announcements.

For brevity, we denote a scenario with three letters $\{V, H, M\}$, indicating the location of the *victim*, *hijacker*, and *mitigator* PEERING sites, respectively. For instance, "{G,A,I}" denotes the experiment where the victim and hijacker ASes are connected to GRN and AMS sites, respectively, and mitigation is performed through BGP announcements from an outsourcing AS connected to ISI. In deaggregation scenarios, the mitigation is self-operated by the victim AS, thus the first and third letters are the same, *e.g.*, "{G,A,G}". When we consider only the hijacking and not the mitigation phase, we use only the first two letters, *e.g.*, "{G,A,*}".

Monitoring the Experiments. In the ARTEMIS prototype we use the BGPmon [7] and the RIPE RIS [12] streaming services for the continuous real-time monitoring of the Internet control plane and the detection of hijacking events. In our experiments, we use the same services to monitor the mitigation process as well.

The BGPStream framework provides BGP updates from *all* the monitors of RIPE RIS and RouteViews, currently with a delay of several minutes (see § 3). Hence, we use BGPStream for a post-analysis of the experiments: after the experiment we collect the BGP updates received by the monitors during the experiment and analyze them. We present these results, and compare them with those from the current real-time monitors, to demonstrate the performance of ARTEMIS when more monitors turn real-time.



(a) detection delay (b) # polluted monitors

Fig. 7: (a) Detection delay for different *location scenarios* (x-axis), and origin-AS (type-0) hijacks. (b) Average number of real-time monitors that observed hijacked routes over time. Boxplots/curves correspond to average values over 10 experiment runs per scenario. ARTEMIS detects hijacks within a few seconds (usually < 5s), while the hijack is observed by most of the monitors in less than 10s.

7.2 Experimental Results

We next analyze the results of our experiments, w.r.t. the *time* needed by ARTEMIS to detect and mitigate hijacking events in various scenarios.

Detection

We consider the *detection delay*, $t_d - t_h$, *i.e.*, the time elapsed between the hijacker's announcement (t_h) and the detection of the event by ARTEMIS (t_d).

In Fig. 7(a) we present the distribution of the detection delay for different location scenarios, under type-0 hijacking events. Boxplots correspond to the values of 10 runs per scenario, for either real-time services (black boxplots) or all services (*i.e.*, post-analysis with BGPStream for RouteViews and RIPE RIS monitors; gray boxplots). We note that the following insights are valid across hijacking event types, since we observed (results omitted for brevity) that the type does not significantly affect the detection delay; small increases (no more than a few seconds) can though occur because in high-type hijacks, less hijacked routes eventually reach the monitors (due to the preference of shorter AS-paths). Moreover, the tunable waiting time of *Stage* 2 (in case *Stage 1* does not suffice, see § 5.4) for type-$\{N \geq 2\}$ hijacks can be added to the detection delay.

ARTEMIS achieves near real-time detection, within a few seconds of the hijacker's announcement. The ARTEMIS detection process is lightweight and thus a hijack event is detected almost instantaneously after the reception of an illegitimate BGP update. Hence, the detection delay is equivalent to the delay of the monitoring services. Specifically, Fig. 7(a) shows that the detection via the *real-time* services is extremely fast, and in some cases *only* 1*s* is required. In all cases the *median of the detection delay is at most* 5*s*. The delay is almost always less than 10s, and in the worst case 13s ({*G*,*I*,*} scenario in Fig. 7(a)). In fact, the 1s delay in some experiments, indicates that the ARTEMIS approach reduces the detection delay to the propagation time of BGP updates (from the hijacker to the monitors): the detection takes place upon the *first* BGP update that reaches any monitor. This propagation time depends on the location/connectivity of the hijacker, *e.g.*, we observe that the detection is on average $2-3$s faster when the hijacker is the GRN site ({*A*,*G*,*} and {*I*,*G*,*} scenarios).

Adding monitors decreases detection delay and increases visibility of hijacks. If all RouteViews and RIPE RIS monitors provided real-time streams (gray boxplots), detection delay could further decrease; the improvement is small in our experiments, since the detection with real-time services is already fast. Moreover, as already discussed in § 4, adding more monitors increases the visibility of hijacks. For instance, in the {*A*,*I*,*} scenarios where the victim (AMS) has much higher connectivity than the hijacker (ISI), while the (exact prefix) hijack is not detected by real-time services, using all monitors would enable a timely (< 6s) detection.

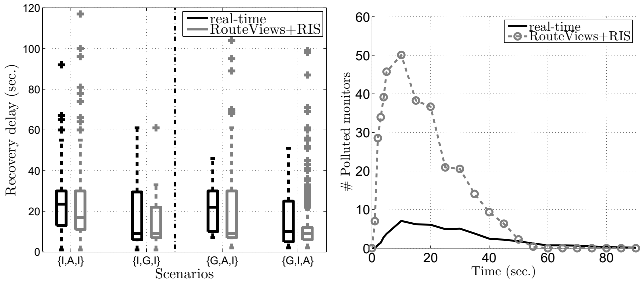
Detection is robust. In Fig. 7(b) we present the average number of real-time monitors that observed a hijacked route over time for different scenarios. While ARTEMIS is able to detect an event from a single (*i.e.*, the first seen) hijacked route, its robustness (*e.g.*, against monitor failures) increases with the number of observed routes. The experimental results in Fig. 7(b) demonstrate that the detection delay would remain low even under multiple monitor failures: while the number of observed hijacked routes differs among scenarios (due to the connectivity of the hijacker), in all of them *(i)* more than 5 monitors observe the event within 5s, and *(ii)* almost half of the monitors that eventually observe the event, see the hijacked route within 10s. Our post analysis with BGPstream shows a similar trend (with the respective number of monitors being $3-4$ times higher).

Mitigation

We next study how fast the hijacking event is mitigated when using the ARTEMIS approach. To quantify the speed of the mitigation, we define the *recovery delay* as the time elapsed between the pollution of an AS/monitor by a hijacked route, until it receives again a legitimate route (*e.g.*, to the deaggregated sub-prefixes).

ARTEMIS achieves almost complete mitigation of the hijacking event within a minute. In Fig. 8(a) we present the distribution of recovery delay (over different ASes and experiment runs) for different location and mitigation scenarios. The time for hijacked ASes to recover is similarly distributed for different mitigation types, and it tends to be higher when the hijacker is a well connected AS (see, *e.g.*, {*I*,*A*,*I*} and {*G*,*A*,*I*} scenarios where AMS is the hijacker). Nevertheless, the following main observations hold for all scenarios: (i) half of the ASes (see medians) recover from a hijacking event in *less than* 30*s*, and the vast majority of them *in less than a minute* (with some outliers reaching up to 2min.). This clearly demonstrates the benefits of the automated mitigation of ARTEMIS, compared to current practices that usually need several hours to mitigate such events (see § 8.1).

Fig. 8(b) shows the average number of polluted monitors (*i.e.*, with a route to hijacker) over time for the {*I*,*A*,*I*} scenario. We observe that the number of polluted monitors increases fast after the event and reaches its peak within 10s. After the event is detected (typically in 3-6s for the {*I*,*A*,*I*} scenario; see Fig. 7(a)), the mitigation starts immediately.



(a) recovery delay (b) # polluted monitors

Fig. 8: (a) Recovery delay of the the real-time (black) and all (Routeviews+RIS) (gray) monitors; type-0 hijacks and mitigation through deaggregation ($\{I,A,I\}$ and $\{I,G,I\}$) and outsourcing BGP announcements ($\{G,A,I\}$ and $\{G,I,A\}$) scenarios. (b) Average number of polluted real-time (black) and all (RouteViews+RIS) (gray) monitors for the $\{I,A,I\}$ scenario. Boxplots/curves correspond to average values over 10 experiment runs per scenario. With the automated mitigation of ARTEMIS, the vast majority of the ASes recover from the hijack within 60s.

The routes for the deaggregated prefixes start propagating, leading to a fast recovery of the polluted monitors within 10-50s after the event; *within a minute* the vast majority of monitors have recovered. We observed similar behavior in all scenarios, indicating that the real performance of ARTEMIS in practice would be similar to the results of Fig. 8(b).

8 State of the Art

8.1 Real-world Problems with BGP Hijacking

We now look at the reasons for which BGP prefix hijacking, although extensively studied, remains a serious threat to Internet operators and users. To this end, we discuss the current practices and complement our discussion with findings of a survey we conducted among network operators. The survey [14], [62] was launched in April 2017, targeting mailing lists of network operator groups to improve our understanding of the: *(i)* real impact of BGP prefix hijacking, *(ii)* currently used defenses, as well as *(iii)* the concerns, needs and requirements of network operators. We received answers from 75 participants operating a broad variety of networks (ISPs, CDNs, IXPs, etc.) around the world (the detailed results of the survey can be found in [62]).

Operators are reluctant to deploy proactive defenses, since they offer limited protection against hijacking. Several modifications to BGP to protect networks against prefix hijacking have been proposed [38], [39], [42], [64], but are not implemented due to several political, technical, and economic challenges. Proactive defenses that are deployed mainly comprise prefix filtering and RPKI [30], [33], [34]. *Prefix filtering* can be used by ISPs to discard route announcements for prefixes that their customers are not allowed to originate. However, prefix filtering is currently applied by a small number of ISPs (due to lack of incentives, poor trust mechanisms, need for manual maintenance, etc. [33]) and offers protection only against a few potential hijackers (*i.e.*, their customers) and hijacking events (origin-AS). *RPKI* enables automated route origin authentication (ROA) in the Internet, to prevent origin-AS hijacks. However, the small percentage of prefixes covered by ROAs (around 7% in Oct. 2017 [50]) and the limited deployment of RPKI route origin validation (ROV) [30], [50], [66] leaves the vast majority of networks unprotected [26], [30]. Our survey results and previous studies [30] reveal the main reasons that hinder the deployment of RPKI: little security benefits, mistrust issues, inter-organization dependencies for issuing ROA certificates, operating costs, and complexity.

Hijacking events, under current practices, have a lasting impact on the Internet's routing system. Due to the insufficiency of proactive mechanisms, networks mainly defend against hijacking events in a *reactive* manner. The speed of the reactive defenses is crucial; even short-lived events can have severe consequences [3]. However, the reality shows that, currently, hijacking events are not quickly mitigated. For instance, back in 2008, a hijacking event affected YouTube's prefixes and disrupted its services for 2 hours [15]. More recently, in Sep. 2016, BackConnect hijacked, at different times, several ASes; the events lasted for several hours [47]. In Apr. 2017, financial services, like Visa and Mastercard, and security companies, like Symantec, were hijacked by a Russian company for seven minutes [4]. Moreover, past experience of operators who participated in our survey shows that their networks were affected for a long time by hijacking events: more than 57% of events lasted *more than an hour*, and 25% lasted even more than a day; only 28% of the events were short-termed, lasting a few minutes (14%) or seconds (14%).

The mitigation of hijacking events is delayed primarily due to third-party detection and lack of automation. Reactive defenses comprise two steps: *detection* and *mitigation*. Several systems have been proposed for prefix hijacking detection [25], [36], [40], [59], [63], [68], [69], with most of them being designed to operate as third-party services; they monitor the Internet control/data plane and upon the detection of a suspicious event or anomaly, notify the involved ASes. Our survey reveals a similar trend in practice: more than 60% rely on third-parties (*e.g.*, [8]) to get notified about suspicious events involving their prefixes. Although state-of-the-art third-party detection services can quickly notify networks about suspicious events[10], the alerts are not always accurate (*i.e.*, false positives), as discussed in [48] and self-reported in our survey [62]. False alerts might be triggered by third-parties for legitimate events (*e.g.*, MOAS, traffic engineering, change of peering policies), due to missing/inaccurate/stale information. As a result, operators need to manually verify the alerts received by third party services; this process introduces significant delay in the detection step, and prevents networks from automating their mitigation counteractions. Finally, extra delay is added to the process of mitigation itself, which frequently takes place in an ad-hoc way: for example, upon the detection of a hijack, operators start contacting other operators to identify the root of the problem, and resolve it.

10. However, 17% of the participants in our survey expect to get notified for hijacking events by receiving notification from colleagues, clients, mailing lists, etc., which implies significantly delayed detection.

Interestingly, this is the only action that 25% of operators in our survey would take to mitigate the hijack; however, with this approach the resolution of the problem might require several hours or even days [48].

8.2 Related Literature

8.2.1 Detection of BGP Hijacking

BGP hijacking detection approaches can be classified based on the type of information they use, as: (i) control-plane, (ii) data-plane, and (iii) hybrid. Each type has its own strengths and weaknesses, which we analyze in the following. For convenience, we also summarize in Table 1 the classes of hijacking events that can be detected by existing systems.

Similarly to ARTEMIS, *control-plane approaches* [8], [25], [40], [53] collect BGP updates or routing tables from a distributed set of BGP monitors and route collectors, and raise alarms when a change in the origin-AS of a prefix, or a suspicious route is observed. Since they passively receive BGP feeds, they are considered quite lightweight. They can detect Type-0 (and Type-1) hijacking events, both for exact prefixes and subprefixes, independently of how the hijacker handles the attracted traffic on the data plane (blackholing-*BH*, imposture-*IM*, man-in-the-middle-*MM*). However, in contrast to ARTEMIS, state-of-the-art systems [8] miss advanced type-N, N≥2 hijacking events that are harder to detect and can be used by a sophisticated attacker. Furthermore, since they are designed as third-party detection services, they have to deal with the real-world problem (§ 8.1) of keeping what they observe consistent with the ground truth on the operator's side, to achieve low false-positive rates while preserving their real-time performance.

Data-plane approaches [68], [69] follow complementary methods to ARTEMIS, using pings/traceroutes to detect hijacks on the data plane. They continuously monitor the connectivity of a prefix and raise an alarm, when significant changes in the reachability [68] of a prefix or the paths leading to it [69] are observed. iSpy [68] can be deployed by the network operator itself (similar to ARTEMIS). However, it cannot reliably detect sub-prefix hijacking events, since it targets few IP addresses per prefix, and can be severely affected by temporary link failures or congestion near the victim's network, increasing its false positive rates. Finally, since data-plane approaches require a large number of active measurements to safely characterize an event as a hijack, they are more heavyweight than control-plane-assisted approaches [63].

Hybrid approaches [36], [59], [63], [55], [58] combine control and data plane information, and sometimes query external databases (*e.g.*, Internet Routing Registries, IRR) [59], [63], to detect multiple classes of hijacking events. HEAP [59] can detect any type of `AS-PATH` manipulation on the control plane, but is limited to sub-prefix hijacking events which result in blackholing or imposture on the data plane. Thus, it misses *MM* attacks both for exact prefix and sub-prefix hijacks. The state-of-the-art detection system Argus [63], is able to achieve few false positives/negatives and timely detection, both for exact prefixes and subprefixes, by correlating control and data plane information. However, Argus considers only *BH* attacks, whereas ARTEMIS is able to detect a hijack even if the hijacker relays traffic (*MM*) or responds (*IM*) to it. The same issue is faced by [36], where only *BH* and *IM* attacks can be detected for any kind of prefix, while *MM* attacks remain under-the-radar. LOCK [55] locates attacker ASes by actively monitoring control/data-plane paths towards the victim prefix. It relies on the evaluation of AS adjacencies to detect *BH*, *IM* or *MM* attacks, but it might miss sub-prefix and stealthy Type-U hijacks. Schlamp *et al* [58] analyze and focus on a specific hijack case where certain conditions, such as the attack on unannounced BGP prefixes (BGP squatting), apply; data sources such as IRRs or DNS could be used to warn vulnerable ASes.

Finally, while there is no consistent ground truth or dataset with which to compare all the claimed FN/FP rates of the aforementioned approaches, we stress that the detection approach of ARTEMIS (§ 5, summarized in Table 4) is the first to combine *all* the following characteristics: self-operated, ground truth-based, lightweight detection, allowing for increased accuracy of alerts (0 false positives for most classes, virtually 0 false negatives), and comprehensiveness in terms of attack class coverage, no matter how the attacker manipulates the control and data planes to execute the hijack.

8.2.2 Mitigation of BGP Hijacking

Several proposals exploit cryptographic mechanisms to prevent BGP hijacking [39], [42], [43], [64]. Others [38] delay the installation of suspicious BGP routes, in order to allow network administrators to verify first and then install them. However, these approaches require modifications to BGP and/or global adoption, as proactive countermeasures to hijacking events; this has been shown to be infeasible due to important technological, political and economic factors. In contrast, we propose reactive self-operated mitigation (prefix deaggregation) or outsourcing it to a single (or, a few) organization(s), which are based on security models used in practice and -as shown in our study- can be very efficient, without requiring large-scale coordination. In fact, we show (see Fig 6(a)) that using only a handful top ISPs for outsourcing BGP announcements, the attained benefit ($< 5\%$ attacker success rate) would require two orders of magnitude more top ISPs to coordinate and perform Route Origin Validation (ROV) in RPKI [30].

Zhang *et al.* [67] propose a reactive mitigation mechanism based on the purging of illegitimate routes and the promotion of valid routes. Compared to outsourcing BGP announcements, the approach of [67] requires one order of magnitude more mitigator ASes ("lifesavers") to achieve similar benefits, as well as complex coordination among these ASes. A similar approach to outsourcing BGP announcements is introduced in [54], whose focus is on selecting an optimal set of ASes as monitor/relay "agents" per victim-hijacker pair. Those results are complementary to our study which considers *existing* monitoring infrastructure and organizations that *currently* offer outsourced security services.

9 Conclusions

BGP prefix hijacking, based on accidental misconfiguration or malicious intent, is a problem that continuously pests

Internet organizations and users, resulting in high-profile incidents. State-of-the-art solutions, proposed in research or adopted in daily operations, are not able to counter this situation due to issues related to: *(i)* attacker evasion (*i.e.*, comprehensiveness of detection, e.g., for MitM attacks), *(ii)* problematic accuracy of detection alerts (resulting in *(iii)* slow manual verification and mitigation processes), and *(iv)* incompatibility with the real-world needs of network operators for information privacy and flexibility of countermeasures.

In this work, we proposed ARTEMIS, a *self-operated* control-plane approach for defending against BGP prefix hijacking. ARTEMIS departs from the common approach of third-party detection/notification systems, and exploits local information and real-time BGP feeds from publicly available monitoring services in order to provide an *accurate*, *comprehensive* and *timely* detection. This detection approach enables a *potentially automated*, *configurable*, and *timely* mitigation of hijacking events, that satisfies the needs and requirements of operators (as *e.g.*, expressed in our survey) and is highly effective, based on currently used practices and outsourcing security models. Moreover, as part of our study, we demonstrated the high capability of public monitoring infrastructure for hijacking detection, and showed that the planned transitions to more pervasive real-time streaming can bring substantial benefits. Our simulation results support the feasibility of the ARTEMIS approach while our real-world experiments show that it is possible to neutralize the impact of hijacking attacks within a minute, a radical improvement compared to the defenses used in practice by networks today.

ACKNOWLEDGEMENTS

This work was supported by the European Research Council grant agreement no. 338402, the National Science Foundation grant CNS-1423659, and the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) via contract number HHSP233201600012C.

REFERENCES

[1] https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study.
[2] http://www.bgpmon.net/chinese-isp-hijacked-10-of-the-internet/.
[3] https://www.wired.com/2014/08/isp-bitcoin-theft/.
[4] https://arstechnica.com/security/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/.
[5] http://dyn.com/blog/iran-leaks-censorship-via-bgp-hijacks/.
[6] AS-rank, CAIDA. http://as-rank.caida.org.
[7] BGPmon (Colorado State University). http://www.bgpmon.io.
[8] BGPmon (commercial). http://www.bgpmon.net.
[9] BGPStream. https://bgpstream.caida.org/.
[10] The PEERING testbed. https://peering.usc.edu.
[11] RIPE RIS. http://ris.ripe.net/.
[12] RIPE RIS - Streaming Service. https://labs.ripe.net/Members/colin_petrie/updates-to-the-ripe-ncc-routing-information-service.
[13] The Route Views Project. http://www.routeviews.org/.
[14] Survey on BGP prefix hijacking. http://tinyurl.com/hijack-survey.
[15] YouTube Hijacking: A RIPE NCC RIS case study. http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study, March 2008.
[16] Ripe ncc global technical services update, ripe 74. https://labs.ripe.net/Members/kranjbar/ripe-ncc-technical-services-2017-part-three-focus-on-tools, May 2017.
[17] The CAIDA AS relationships dataset. http://data.caida.org/datasets/as-relationships/, Nov. 2016.
[18] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. Investigating interdomain routing policies in the wild. In *Proc. ACM IMC*, pages 71–77, 2015.
[19] Arbor. Worldwide Infrastructure Security Report. https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf.
[20] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the internet. In *ACM SIGCOMM CCR*, volume 37, pages 265–276, 2007.
[21] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Internet optometry: assessing the broken glasses in internet reachability. In *Proc. ACM IMC*, 2009.
[22] CAIDA. BGPStream V2 Beta. https://bgpstream.caida.org/v2-beta.
[23] CAIDA. CAIDA BGP Hackathon 2016. https://www.caida.org/workshops/bgp-hackathon/1602/index.xml.
[24] Kai Chen, David R Choffnes, Rahul Potharaju, Yan Chen, Fabian E Bustamante, Dan Pei, and Yao Zhao. Where the sidewalk ends: Extending the Internet AS graph using traceroutes from P2P users. In *Proc. ACM CoNEXT*, pages 217–228, 2009.
[25] Ying-Ju Chi, Ricardo Oliveira, and Lixia Zhang. Cyclops: the as-level connectivity observatory. *ACM SIGCOMM CCR*, 38(5):5–16, 2008.
[26] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. Jumpstarting bgp security with path-end validation. In *Proc. ACM SIGCOMM*, 2016.
[27] Danny Cooper, Ethan Heilman, Kyle Brogle, Leonid Reyzin, and Sharon Goldberg. On the Risk of Misbehaving RPKI Authorities. In *Proc. ACM HotNets*, 2013.
[28] Exa-Networks. exabgp: The BGP swiss army knife of networking. https://github.com/Exa-Networks/exabgp.
[29] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *IEEE/ACM TON*, 9(6):681–692, 2001.
[30] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are we there yet? on RPKI's deployment and security. In *NDSS*, 2016.
[31] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Let the market drive deployment: A strategy for transitioning to bgp security. In *ACM SIGCOMM CCR*, volume 41, pages 14–25, 2011.
[32] V. Giotsas, S. Zhou, M. Luckie, and k. claffy. Inferring multilateral peering. In *Proc. ACM CoNEXT*, 2013.
[33] Sharon Goldberg. Why is it taking so long to secure internet routing? *Communications of the ACM*, 57(10):56–63, 2014.
[34] Sharon Goldberg, Michael Schapira, Pete Hummon, and Jennifer Rexford. How secure are secure interdomain routing protocols? volume 70, pages 260–287. Elsevier, 2014.
[35] Susan Hares, Yakov Rekhter, and Tony Li. A border gateway protocol 4 (bgp-4). https://tools.ietf.org/html/rfc4271, 2006.
[36] Xin Hu and Z Morley Mao. Accurate real-time identification of ip prefix hijacking. In *IEEE Symposium on Security and Privacy*, pages 3–17, 2007.
[37] Geoff Huston. Bgp in 2016. http://www.ipaddressnews.com/wp-content/uploads/2017/02/bgp2016.pdf, 2017. in ISP Column.
[38] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Pretty good bgp: Improving bgp by cautiously adopting routes. In *Proc. IEEE ICNP*, 2006.
[39] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (s-bgp). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, 2000.
[40] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. Phas: A prefix hijack alert system. In *Usenix Security*, 2006.
[41] Mohit Lad, Ricardo Oliveira, Beichuan Zhang, and Lixia Zhang. Understanding resiliency of internet topology against prefix hijack attacks. In *Proc. IEEE/IFIP Dependable Systems and Networks*, 2007.
[42] Matt Lepinski. BGPSEC protocol specification. https://tools.ietf.org/html/rfc8205, 2015.
[43] Matt Lepinski, Richard Barnes, and Stephen Kent. An infrastructure to support secure internet routing. https://tools.ietf.org/html/rfc6480, 2012.

[44] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, et al. As relationships, customer cones, and validation. In *Proc. ACM IMC*, 2013.
[45] Robert Lychev, Sharon Goldberg, and Michael Schapira. BGP Security in Partial Deployment: Is the Juice Worth the Squeeze? In *Proc. of ACM SIGCOMM*, 2013.
[46] NANOG mailing list archives. Another day, another illicit SQUAT. http://seclists.org/nanog/2016/Oct/578, Oct. 2016.
[47] NANOG mailing list archives. "Defensive" BGP hijacking? http://seclists.org/nanog/2016/Sep/122, Sep. 2016.
[48] NANOG mailing list archives. BGP IP prefix hijack detection times. http://seclists.org/nanog/2017/Feb/293, Feb. 2017.
[49] Stephanos Matsumoto, Raphael M. Reischuk, Pawel Szalachowski, Tiffany Hyun-Jin Kim, and Adrian Perrig. Authentication Challenges in a Global Environment. *ACM Trans. Priv. Secur.*, 20:1:1–1:34, 2017.
[50] NIST. RPKI Monitor. https://rpki-monitor.antd.nist.gov/, 2017.
[51] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. Bgpstream: A software framework for live and historical bgp data analysis. In *Proc. of ACM IMC*, pages 429–444, 2016.
[52] A. Pilosov and T. Kapela. Stealing The Internet: An Internet-Scale Man In The Middle Attack. https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf, 2008. in Defcon 16.
[53] Jian Qiu, Lixin Gao, Supranamaya Ranjan, and Antonio Nucci. Detecting bogus bgp route information: Going beyond prefix hijacking. In *Prov. IEEE SecureComm*, pages 381–390, 2007.
[54] Tongqing Qiu, Lusheng Ji, Dan Pei, Jia Wang, and Jun Xu. Towerdefense: Deployment strategies for battling against ip prefix hijacking. In *Proc. IEEE ICNP*, pages 134–143, 2010.
[55] Tongqing Qiu, Lusheng Ji, Dan Pei, Jia Wang, Jun Jim Xu, and Hitesh Ballani. Locating prefix hijackers using lock. In *USENIX Security Symposium*, pages 135–150, 2009.
[56] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. *ACM SIGCOMM CCR*, 36(4):291–302, 2006.
[57] RIPE NCC. RIPEstat. https://stat.ripe.net/.
[58] Johann Schlamp, Georg Carle, and Ernst W Biersack. A forensic case study on as hijacking: the attacker's perspective. *ACM SIGCOMM CCR*, 43(2):5–12, 2013.
[59] Johann Schlamp, Ralph Holz, Quentin Jacquemart, Georg Carle, and Ernst Biersack. HEAP: Reliable Assessment of BGP Hijacking Attacks. *IEEE JSAC*, 34(06):1849–1861, 2016.
[60] Brandon Schlinker, Kyriakos Zarifis, Italo Cunha, Nick Feamster, and Ethan Katz-Bassett. Peering: An AS for us. In *Proc. ACM HotNets*, 2014.
[61] John Scudder, Rex Fernando, and Stephen Stuart. BGP monitoring protocol. https://tools.ietf.org/html/rfc7854, 2016.
[62] Pavlos Sermpezis, Vasileios Kotronis, Alberto Dainotti, and Xenofontas Dimitropoulos. A survey among network operators on bgp prefix hijacking. *ACM SIGCOMM CCR*, 48(1):64–69, 2018.
[63] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu. Detecting prefix hijackings in the Internet with Argus. In *Proc. ACM IMC*, 2012.
[64] Lakshminarayanan Subramanian, Volker Roth, Ion Stoica, Scott Shenker, and Randy Katz. Listen and whisper: Security mechanisms for bgp. In *Proc. NSDI*, 2004.
[65] Pierre-Antoine Vervier, Olivier Thonnard, and Marc Dacier. Mind your blocks: On the stealthiness of malicious bgp hijacks. In *Proc. NDSS*, 2015.
[66] Matthias Wählisch, Robert Schmidt, Thomas C Schmidt, Olaf Maennel, Steve Uhlig, and Gareth Tyson. Ripki: The tragic story of rpki deployment in the web ecosystem. In *Proc. ACM HotNets*, 2015.
[67] Zheng Zhang, Ying Zhang, Y Charlie Hu, and Z Morley Mao. Practical defenses against bgp prefix hijacking. In *Proc. ACM CoNEXT*, 2007.
[68] Zheng Zhang, Ying Zhang, Y Charlie Hu, Z Morley Mao, and Randy Bush. iSPY: detecting ip prefix hijacking on my own. In *ACM SIGCOMM CCR*, volume 38, pages 327–338, 2008.
[69] Changxi Zheng, Lusheng Ji, Dan Pei, Jia Wang, and Paul Francis. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *ACM SIGCOMM CCR*, volume 37, pages 277–288, 2007.