Route Leak Detection Using Real-Time Analytics on local BGP Information

M. S. Siddiqui[†‡], D. Montero[†], M. Yannuzzi[†], R. Serral-Gracià[†], X. Masip-Bruin[‡], W. Ramirez[‡]
[†]Networking and Information Technology Lab (NetIT Lab)
[‡]Advanced Network Architectures Lab (CRAAX)
Technical University of Catalonia, Spain
Email: {siddiqui, dmontero, yannuzzi, rserral, xmasip, wramirez}@ac.upc.edu

***Abstract*—A route leak can be defined as a security gap that occurs due to the infringement of the routing policies that any two Autonomous Systems (ASes) have agreed upon. Route leaks are seemingly simple, but hard to resolve since the ASes keep their routing policies confidential. Indeed, the traditional palliatives, such as the utilization of route filters, are no longer used by a large number of ASes, given the high administrative burden that they entail. Other alternatives, like BGP monitoring tools, not only require third party information gathered at multiple vantage points, but also they become impotent in many cases, due to their limited view of the interdomain routing state. In this paper, we propose a different approach, which allows to autonomously detect the occurrence of route leaks by solely inspecting the BGP information available at the AS. Our main contributions can be summarized as follows. First, we propose a self-contained Route Leak Detection (RLD) technique, which is based on real-time analytics on the Route Information Bases (RIBs) of the border routers of an AS. Second, we introduce Benign Fool Back (BFB), "a harmless bluff" that can substantially improve the success rate of the RLD technique. Third, we show through exhaustive simulations that our technique can detect route leak incidents in various scenarios with high success rate. In addition, our solution has the following practical advantages: a) no reliance on third party information (e.g., on vantage points); b) no changes required to control-plane protocols (e.g., to BGP); and c) allows non-invasive integration (e.g., using SDN).**

***Index Terms*—BGP, security, route leaks, inter-domain routing.**

I. Introduction

A route leak occurs when the routing policies agreed between two neighbor Autonomous Systems (ASes) are not respected. This type of policy violation takes place during the route advertisement process between these ASes. More precisely, the business relationship between any two ASes steers their export and import routing policies, and a route advertised against the conceded policies is called a route leak. To illustrate this, let us consider the two dominant business relationships between ASes in the Internet, namely, *customer–provider (or provider–customer)* and *peer–peer*. In the former case, the provider normally offers transit for the customer, while in the latter, the two ASes usually cater each others customer's traffic. In this framework, if a customer offers transit between its providers, i.e., it exports a route learned from one provider to the other, then it causes a route leak.

In practice, route leaks can lead to partial paralysis of Internet services, and may affect both local as well as global regions. Indeed, route leaks can be either the result of a misconfiguration or a deliberate attack, and, apart from Internet service disruption, they can lead to sub-optimal routing and traffic hijacking. Therefore, route leaks can be very harmful, and they are considered a security threat for the interdomain routing system. For example, in February 2012, an Internet service failure at national level occurred in Australia, when a multi-homed ISP leaked routes learned from one of its providers to another provider [1]. In November of the same year, Google services were disrupted when one of Google's peers improperly advertised Google routes to its provider [2].

Unfortunately, the protocol for exchanging interdomain routing information, namely, the Border Gateway Protocol (BGP) [3], fails to avoid route leaks, since it lacks an internal security mechanism to that end. The first line of defense for preventing route leaks typically consists of utilizing route filters along with Internet Route Registries (IRRs) information, but this palliative usually becomes futile due to the high administrative cost of maintaining the filters updated. Other stopgap solutions, such as BGP monitoring tools, rely on the information collected at various vantage points, but they are only fruitful when the irregularities are observed at the vantage points themselves. In this regard, the fact that adds to the difficulty of countering route leaks using vantage points is the secrecy of the routing policies among ASes. Although several attempts have been made for inferring the relationships and the policies among ASes (see, e.g., [4], [5], [6]), more recent works are currently questioning the accuracy of these techniques [7]. This is mainly due to the fact that the knowledge base for inferring the AS relationships and their corresponding export policies is limited to the routing information available at the data collection points. In particular, the increase in the number of Internet Exchange Points (IXPs) and their role in the recent "flattening" of the Internet topology, makes that a large fraction of AS relationships cannot be discovered using these data collection points [7].

Based on these observations, this paper makes the following contributions. First, we propose a Route Leak Detection (RLD) technique that allows an AS to detect the occurrence of route leaks, by applying analytics on the routing information available at hand. This information includes: i) the Routing Information Bases (RIBs) of all the border routers within the AS; and clearly ii) the knowledge of the business relation-

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

ship with its neighbors. As we shall show, our solution does not need any vantage point deployed in the internetwork for its operation. Second, we introduce Benign Fool Back (BFB), an ingenious and harmless bluff that can help improving the number of successful detections. And third, we demonstrate through exhaustive simulations the strengths of our RLD technique. To the best of our knowledge, our research introduces the first practical analysis for autonomously detecting route leaks in the Internet.

It is worth highlighting that, this paper focuses on the "detection" algorithms, that is, aspects such as "remediation" techniques are out of the scope of this work. Anyway, we contend that a remediation solution exploiting our RLD technique can be deployed as an SDN application, which could be used for transparently protecting BGP from route leaks—by "transparently" we mean that BGP would not even be aware about the route leak incidents and their prompt remediation. Moreover, our RLD technique can be implemented either as a centralized or a distributed network application within the AS. In any case, these are mainly implementation decisions, and so they are out of the scope of this paper as well.

The rest of the paper is organized as follows. Section II describes route leaks and the motivation for their resolve. The proposed Route Leak Detection (RLD) technique is explained in Section III. The simulation framework for testing the RLD technique, along with the analysis of results is discussed in Section IV. Section V introduces the Benign Fool Back (BFB). Finally, Section VI concludes the paper.

II. Route Leaks

Route leaks are capable of disrupting Internet service on a large scale and thus require serious considerations for their resolve. Unfortunately, there is no standard definition of the route leak problem in the Internet community. The IETF working group responsible for securing inter-domain routing, SIDR WG, considers the route leak problem out of their scope. In fact, their proposed solutions, including RPKI [8], ROA [9] and BGPSEC [10], fail to counter route leaks. More recently, the task of defining the route leak problem was taken up by Global Routing Operations (GROW) WG.

In the literature, the route leak problem has been referred as a violation of the business relationship that rules the interconnection of domains. In [11], the author defines a route leak as: *"non-customer routes received over a Peer or a Customer link"*. Unlike [12], in this paper we make a clear distinction between route leaks and IP prefix hijacking. We emphasize that a route leak does not require a false route origin or false AS-Path claim to succeed. For example, when Moratel leaked Google routes toward its provider AS, it didn't need to claim ownership of Google routes, neither did it need to claim a false AS-Path to Google. The only violation was that Moratel advertised Google routes toward its provider against the business relationship of the respective link. So, we reiterate that a route leak occurs only when the export policies violate the link classification between two ASes.

As mentioned earlier, the link classification between two ASes can represent either a peer–peer or a customer–provider relation. In the customer–provider case, the customer AS only advertises its own routes and the routes of its customers. On the other hand, the provider AS advertises all its routes toward its customer, hence providing it transit to rest of the Internet. In the peer-peer relation, the ASes only advertise to each other their own routes as well as their customers' routes.

From the business perspective, the provider AS charges its customer AS for forwarding its traffic to and from it. Whereas in the peer–peer relation, the ASes remain revenue neutral for exchanging each other's customer traffic up to an agreed threshold. This is the reason why an AS prefers customer routes (i.e., routes learned from customer ASes) over peer or provider routes (i.e., routes learned from peer ASes or provider ASes, respectively), and it is the principal reason why route leaks become rapidly pernicious. These preferences turn fatal when the customer AS or the peer AS leak routes, i.e., they advertise non-customer routes toward their provider AS or peer AS, respectively. In the rest of the paper, the scenario when an AS advertises non-customer routes toward its provider AS shall be referred as a Customer Route Leak (CRL). Likewise, the scenario when an AS advertises non-customer routes toward its peer AS, shall be called a Peer Route Leak (PRL).

Observe that, due to the routing preferences described above, in many cases these routes will get selected by the neighbors as their best routes, and therefore, their traffic flows will be redirected toward the AS that made the route leak. Hence, apart from misconfigurations, a certain route can be leaked to attract traffic either for sniffing it or hijacking it. Also note that, once the routes are selected as the best ones, BGP routers will further export them to their neighbors (and so on), hence rapidly causing a wide scale incident. All in all, the motivation behind preferring customer routes over peer and provider routes, and preferring peer routes over provider routes is purely economical, since forwarding traffic to a customer produces revenue, to a peer implies no revenue, while to a provider entails a cost.

An AS can receive a route leak either due to an action initiated by a neighbor AS or due to the propagation of a route leak throughout the network. In the former case, the neighbor AS is the one who initiated the route leak, whereas in the latter, the neighbor AS forwarding the route might have received the route leak from one of its neighbors, and so forth. It is worth mentioning that it is more difficult to identify a propagated route leak than a route leak initiated by a neighbor AS. This is because the AS propagating the route leak may forward the route leak to its neighbors according to the relationship it has with them, which makes it more difficult for any AS receiving the propagated route leak to detect it as a route leak.

Thus, it is essential to focus on the export policies among ASes with varying relationships, as the violation of these export policies marks the birth of a route leak. The starting guidelines for exporting routes, i.e., how to advertise routes depending on the type of relationship with the neighbor AS from whom it has received the route, and to whom it plans

forward it, are referred as valley-free rules [4]. The valley-free rules include: 1) routes learned from a customer AS are further advertised to other customer, peer and provider ASes; 2) routes learned from a peer AS are further advertised to customer ASes only; 3) routes learned from a provider AS are further advertised to customer ASes only. In this regard, the valley-free rules can serve as basic guidelines for resolving the route leak problem. That is, if a route is advertised by an AS toward its neighbor AS such that it is in violation of any of the three valley-free rules, then it can be considered as a route leak.

We proceed now to present the proposed Route Leak Detection (RLD) technique, which, as we shall show, it is based on the application of analytics on readily available in-the-box information (i.e., information from the routers' RIBs and the knowledge of the AS relationships with the neighbors).

III. Route Leak Detection

The failure of the traditional countermeasures for detecting route leaks is evident from the frequent occurrences of Internet service disruptions due to these incidents. Learning from the collapse of traditional solutions, we can infer that any approach toward resolving the route leak problem should consider the following factors: 1) minimum reliance on third party information; 2) minimum possible changes to the legacy control-plane protocols; 3) real-time detection; and 4) minimum possible administrative overhead. The minimum reliance on third party information is important not only because of the limited reach of the information gathered at vantage points, but also because of the high administrative cost required to train and maintain the monitoring infrastructure up-to-date with the routing policies. Furthermore, serious efforts are required for trust establishment between the relying party and the third party to avoid bogus information exchanges. The second factor stems from the fact that a solution requiring significant changes to control-plane protocols will meet the same fate as such previous inter-domain security propositions, i.e., resistance in adoption. Then, the real-time detection is a necessity because of the way route leaks operate. As mentioned earlier, detecting route leak initiation is easier than detecting propagated route leaks, hence early detection of a route leak is essential. Moreover, two vital goals to be considered when designing a route leak detection algorithm are, to ensure a low administrative cost for maintaining the system, and that the detection technique itself does not hinder the rest of the network functions.

From this perspective, we propose a very simple yet powerful Route Leak Detection (RLD) technique to counter route leaks. This RLD technique enables an AS to autonomously detect route leak initiations by only relying on readily available information at hand. The work presented in this paper is a major extension of the study published in [13]. The main contributions of this work include, 1) the Route Leak Detection algorithm and its improvements including rigorous cross-path check, Tier-1 AS check and the Benign Fool Back (BFB) strategy, 2) the experimental framework and extensive simulations for verifying the strength of the RLD algorithm. Before introducing the internal workings of the detection technique, we will discuss the assumptions and considerations made in this paper. Our framework makes two sensible assumptions regarding the Internet topology; 1) An AS v does not have a peer-peer relationship with its provider's provider; 2) The Internet topology is free of cyclic chains of customer-provider relationships, e.g., AS v cannot be the provider of its provider's provider. That is, we exclude the scenarios where an AS is peering with a provider of its provider, as well as any cyclic chain of customer-provider relationships.

Moreover, the framework under consideration also excludes uncommon AS relationships such as sibling and hybrid relations, given that such AS relationships are relatively negligible as compared to peer-peer and customer-provider relations. Furthermore, the detection of route leaks in case of hybrid or sibling relationship is out of the scope of this paper.

From an operational point of view, the RLD algorithm works in two stages, namely, training and detection. In the training stage, up-to-date route filters are utilized to ensure valley-free valid RIBs for the initial period of time. That is the routers' RIBs only contain valley-free routes and do not contain any route leak before proceeding to the detection stage. After the training stage, the route filters do not need to be maintained. Now, with valley-free valid RIBs as a reference point, each new coming route advertisement goes through the route leak detection stage.

Let us consider an example for explaining the detection phase of the RLD technique. An AS v (the potential victim) receives a route advertisement with an AS-Path of the form $[l, o, \dots]$ from the neighbor AS l (the potential leaker). For every route advertisement received, AS v examines the type of relationship it has with the neighbor which forwarded the route advertisement—in our example, AS l. If the route advertisement is from a provider AS (i.e., AS l is a provider of AS v), then no further processing is required as by valley-free policies, a provider AS cannot leak any route. But if the route advertisement is from a customer AS or a peer AS, i.e., AS l is either customer (see Fig. 1a), or peer (see Fig. 1b), of AS v, then the origin of the route is inspected. If the route is originated by AS l, then it is not a route leak. This is because it is legitimate for AS l, being peer or customer of AS v, to advertise its own routes to AS v. However, if AS l does not originate the advertised route, then the AS-Path information (i.e., $[l, o, \dots]$) of the route advertisement is compared against the AS-Paths available in the valley-free valid RIBs. Observe that, if a cross-path is found in the RIBs, that is, a path of the form $[\dots, o, l, \dots]$, then the route advertisement can be easily identified as a route leak. In the absence of a cross-path match, the route under consideration needs to be further inspected.

Let us analyze the cross-path matching for the case where AS l is a peer of AS v. In compliance with the valley-free rules, AS l can only advertise a route with AS-path $[l, o, \dots]$ to AS v if AS o is a customer of AS l. This is because AS l is not allowed to advertise non-customer routes to its peer AS v.

But the existence of an AS-Path $[\ldots, o, l, \ldots]$ in the valley-free valid RIBs at AS v is only possible if AS v belongs to the customer cone of AS o, where customer cone is the set of customer ASes and customer's customer ASes down to the edge of the network topology. This is because according to valley-free rules, AS o would advertise its provider routes of AS l only to its customers. But if AS v belongs to the customer cone of AS o, then AS v has a peer relation with the provider of its provider, which, as we assumed above, is not possible in our RLD framework. Hence, AS o is not a customer of AS l and the route advertisement with AS-Path $[l, o, \ldots]$ is a route leak.

Similarly, if AS l is a customer of AS v, then AS l can only advertise the route with AS-path $[l, o, \ldots]$ to AS v if AS o is a customer of AS l. However, the existence of an AS-Path $[\ldots, o, l, \ldots]$ in the valley-free valid RIB at AS v is only possible when AS v belongs to the customer cone of AS o. Following valley-free rules, AS o would advertise its provider routes of AS l only to its customers. But if AS v belongs to the customer cone of AS o, then there is a cyclic chain of provider relationships among AS v, AS l, and AS o. Hence, AS o cannot be a customer of AS l, implying that AS o is either a peer or a provider of AS l. Thus, the route advertised by AS l toward AS v with AS-path $[l, o, \ldots]$ is a route leak.

In order to make the cross-path checking more rigorous, we can generalize the cross-path check in the form $[\ldots, o, \ldots, l, \ldots]$ in the valley-free valid RIBs. In this case, a received route from a customer or a peer AS l of the form $[l, o, \ldots]$ can be declared as a route leak if the route $[\ldots, o, \ldots, l, \ldots]$ exists in the valley-free valid RIBs.

To further enhance the RLD technique, we have observed that using the knowledge about the overall topology of the network, the route leak detection may be improved. To this end, we can use the set of Tier-1 ASes in our algorithm. We consider Tier-1 ASes, also referred to as core of the Internet, as ASes not having any other provider ASes. Then, regarding the RLD technique, we claim that any route received from a customer or peer AS containing a Tier-1 AS in the AS-Path (excluding the first hop in the latter case) can be declared as a route leak. This is because a customer or peer AS would not have a Tier-1 AS in its customer cone. It is important to notice, that even if we stated above that relying on third party information was not recommended, in this case we contend that the set of Tier-1 is a mostly static and reliable list that may be obtained from publicly available information. It is worth mentioning that if a received route is not detected as a route leak by the RLD technique, then it is inducted in the valley-free valid RIB, as show in step 14 of Algorithm 1. Algorithm 1 summarizes the RLD logic for identifying route leaks. The outcome of the algorithm allows the AS to either accept the route update or discard it. We now proceed to describe the methodology used for analyzing the accuracy of our RLD technique.

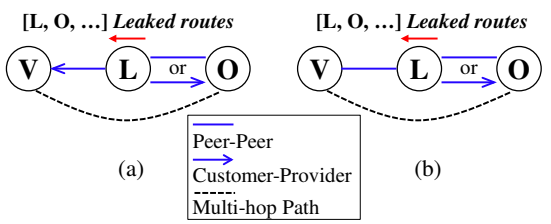

Fig. 1. (a) Customer Route Leak (CRL); (b) Peer Route Leak (PRL).

Algorithm 1 RLD identifies whether a new route advertisement $\mathcal{R}$ received by AS v is a leak.

Input: Valley-free $RIBs$ - Routing Information Bases at v
$\mathcal{N}_c$: Set of customer neighbors of v
$\mathcal{N}_{pe}$: Set of peer neighbors of v
$\mathcal{N}_{pr}$: Set of provider neighbors of v
$\mathcal{T}$: List of Tier-1 ASes
A new route advertisement $\mathcal{R}$ of the form $[l, o, \ldots]$.

Output: **true** if the new route received is a leak
false otherwise.

1: **if** AS $l \in \mathcal{N}_{pe} \cup \mathcal{N}_c$ **then**
2: **for all** $a_i \in \mathcal{R}$, where $0 < i \leq \mathcal{R}.length$ **do**
3: **if** $a_i \in \mathcal{T}$ **then**
4: $\mathcal{R} \leftarrow \emptyset$
5: **return true**
6: **end if**
7: **end for**
8: $\mathcal{R}' \leftarrow [\ldots, o, \ldots, l, \ldots]$
9: **if** $\mathcal{R}' \in RIBs$ **then**
10: $\mathcal{R} \leftarrow \emptyset$
11: **return true**
12: **end if**
13: **end if**
14: $RIBs \leftarrow RIBs \cup \mathcal{R}$
15: **return false**

IV. RLD Simulations and Results Analyses

This section introduces the testing framework and the different simulations performed in order to rigorously validate the RLD technique.

A. RLD Simulations Framework

In order to make possible the utilization of event-driven simulations at a large scale, a number of practical decisions were needed for our testing framework, such as considering a scaled down Internet-like topology. The topology used in this paper consisted of 1007 ASes and 1753 inter-domain links. This topology as well as the relationships between neighbor ASes was extracted from the global-scale ARK's Internet graph [14]. The graph reduction technique that we used for passing from the complete ARK's Internet graph to the 1007 AS topology was based on [15], and the goal in this process was twofold. Firstly, we tried to preserve some of the essential topological properties of the complete Internet graph supplied by ARK, so that the results obtained can be reasonably extrapolated to larger topologies. Secondly, and most importantly, we ensured that the graph used was actually a subgraph of the ARK graph. In other words, all the domains, links, and the AS relationships used in our simulations, are actually present in ARK's Internet graph [14]. Due to the constraints on the scale for

carrying out event-driven simulations, we considered a single router per AS. Observe that, our RLD technique is applied using analytics on the RIBs of all the border routers in the AS and the external advertisements that they receive, so the internal transit routes and the iBGP implications are not expected to considerably influence the detection results described in this section. Indeed, with multiple border routers per AS, i.e., with more than one RIB belonging to the same AS but with different Internet route views, we would actually expect improvements in the detection rates.

The simulations were setup and run using the network simulator NS2 [16] along with BGP++ [17]. BGP++ is based on the standard GNU Zebra routing software and complements NS2's lack of native BGP capabilities.

With this topology and by using the business relations among neighbor ASes obtained from ARK, we first inferred the total number of possible route leak scenarios. We observed that 4409 different route leak cases could be studied for the considered topology. Thus, to evaluate the effectiveness of our RLD technique, a total of 4409 different simulations were conducted over the same topology, covering in this way one route leak scenario per simulation. The methodology used is as follows. As shown in Fig. 2, each route leak scenario involves three participants: 1) the Leaker AS (L); 2) the Victim AS (V); and 3) the Owner AS (O). The Leaker is the AS's router that is configured to leak the routes. The Victim is the AS that will receive the leaked routes, and the Owner is the AS whose routes were improperly advertises toward the Victim. In each simulation, the BGP protocol was initially configured according to the policies and relationships with its neighbors as obtained from ARK (i.e., compliant with the valley-free rules), and it was allowed to converge. This is important to ensure the utilization of valley-free RIBs in the initial state. Once BGP converges, the detection process is activated on the Victim AS. Once our RLD technique is operative, we explicitly reconfigured one AS's BGP router (the leaker L) with export rules that violated the conceded relationship found in ARK—all this was done during the simulation runtime. Clearly, as new BGP updates are received, the detection technique will be analyzing them.

Considering the fact that, from the victim's perspective, a route leak may only be initiated by a customer or a peer neighbor, we have categorized the route leak scenarios into three groups (see Fig. 2):

- **Customer Route Leak (CRL) scenario:** this scenario includes all possible combinations of leaks in which an AS leaks its provider's or peer's routes toward other providers (see Fig. 2(a)).
- **Peer Route Leak (PRL) scenario:** this scenario consists of all possible combinations of leaks in which an AS leaks its provider's or peer's routes toward other peers (see Fig. 2(b)).

Apart from the above two route leak scenarios, we identify a specific case of customer route leaks, namely:

- **Stub Route Leak (SRL) scenario:** this scenario defines all possible combinations of leaks in which a multi-homed stub AS leaks a provider route toward other providers (see Fig. 2(c)). We consider an AS which has no customer or peer ASes and has at least two distinct provider ASes as a multi-homed stub AS.

For the considered topology, the total number of all possible CRLs add up to 2041, out of which 1292 also belong to the SRL scenario. The total number of all possible PRLs is then 4409 – 2041 = 2368. The classification of route leaks into CRL, PRL and SRL, will allow us to analyze the performance and results under different route leak scenarios.

B. Results and Analysis

The results obtained using our RLD technique in the different route leak scenarios are summarized in Table I. In order to provide further insight into the results, we have split the outcomes into two subcategories, depending on the Owner type. In the first one, we consider the case when L leaks the routes learned from one of its providers only, i.e., O is a provider of L (see Fig. 3(a)). In this case, the set of routes leaked from L to V might be potentially large, since it may include O's own routes, as well as its provider, peer and customer routes. In the second subcategory, we consider the case when L leaks the routes learned from one of its peers only, i.e., O is a peer of L (see Fig. 3(b)). In this second case, the set of leaked routes may include O's own routes, as well as those of its customers. Table I summarizes the detection results for the CRL, PRL, and SRL scenarios based on this classification of Owner type.

The number of cases in which O is a provider of L is 1830 for the CRL scenarios, and 410 for the PRL scenarios, out of which our RLD technique detects 97.98% and 99.76% of the leaks, respectively. We can observe that for the SRL scenarios, the RLD technique performs even better within the CRLs, with a detection success rate of 98.14%. This is because in this case, the leaker is a stub AS whereas the victim, being a provider, is topologically well positioned with a broader view of the Internet and the different routes to reach the leaker. Thus, our RLD technique is able to detect most of the leak cases in this

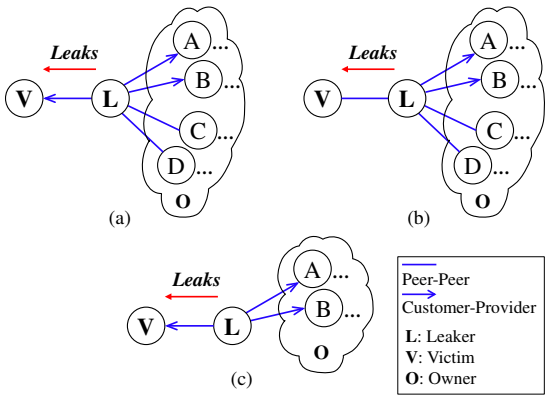

Fig. 2. Categories of route leak scenarios: (a) Customer Route Leaks (CRLs); (b) Peer Route Leak (PRLs); (c) Stub Route Leaks (SRLs).

Categories of Leak Scenarios	Leaks when O is a Provider of L			Leaks when O is a Peer of L			Overall Results (weighted %)		
	# Leaks	# Leaks Detected	% Leaks Detected	# Leaks	# Leaks Detected	% Leaks Detected	# Leaks	# Leaks Detected	% Leaks Detected
CRL: L is a customer of V	1830	1793	97.98%	211	112	53.08%	2041	1905	93.34%
PRL: L is a peer of V	410	409	99.76%	1958	509	26.00%	2368	918	38.77%
SRL: L is a stub and customer of V [a]	1292	1268	98.14%	N/A[b]	N/A[b]	N/A[b]	1292	1268	98.14%

[a] Observe that SRL is a particular case of CRL.
[b] N/A: Not Applicable.

TABLE I
DETECTION RESULTS FOR DIFFERENT ROUTE LEAK SCENARIOS ON A SUBGRAPH OF THE INTERNET TOPOLOGY COMPOSED OF MORE THAN 10^3 ASES.

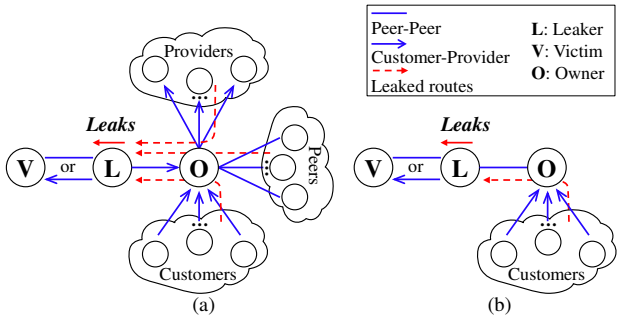

Fig. 3. (a) Leaks toward V when O is a Provider of L; (b) Leaks toward V when O is a Peer of L.

scenario. Observe that the ultimate number of routes leaked by L to V will actually depend on the BGP decision process at L, and the export rules configured toward V. In general, when O is a provider of L, the routes leaked may provide reachability to a broad transit-like block of the Internet, hence observance of cross-paths is more likely.

On the other hand, the number of CRLs in which O is a peer is 211, while for the PRL scenarios is 1958. The detection results for the CRL, and PRL scenarios in this case are, 53.08%, and 26.00%, respectively. The main reason for this low performance is that, when L leaks the routes learned from a peer, the number of routes announced are far less than when the leaked routes are from a provider. These routes provide reachability to a narrower stub-like block of the Internet compared to the former case. Thus, observance of cross-path is less likely, and as reflected in Table I, poor detection success rates are obtained in this case.

The overall results are shown on the right hand-side of Table I. We observe that for all the CRL scenarios evaluated, our detection technique achieves a success rate of 93.34%; and, within these scenarios, the success rate for the SRL cases increases up to 98.14%. On the contrary, for the PRL scenarios, the results obtained are considerably low, achieving only a global success rate of 38.77%. The main conclusion that can be drawn is that our detection technique has high accuracy when the victim is detecting leaks initiated by its customer ASes, but a more creative approach is needed for detecting route leaks initiated by a peer. Indeed, the challenge arises when L is a peer of V, and the routes leaked by L belong to one of its peers O. This is precisely the motivation for the proposal introduced in next section.

V. BENIGN FOOL BACK

In order to improve the success rate in the detection when L leaks its peer routes toward V, we propose *Benign Fool Back (BFB)*. We assume that, in general terms, the leaker follows the principle of preferring customer routes over peer and provider routes, and that it prefers a shorter AS-path route over a longer one. The term "in general" means that this policy might not necessarily apply to all L's routes, but at least applies for a fraction of them. We also assume that the ASes involved in the potential route leak incident are not using IP prefix origin verification mechanisms, such as ROA [9]. We claim that these are realistic assumptions, since most of the route leaks reported in the Internet are due to apparent misconfigurations rather than deliberate attacks, and ROA is not used by the large majority of the ASes in the Internet.

To illustrate BFB, let us consider the example shown in Fig. 4 (a). If the potential victim V starts receiving new routes from a peer L, for which V had never had any route through L, then V can be suspicious of these new routes, and trigger the BFB strategy if the RLD technique described in the previous sections did not detect any leak. For this, V chooses one or more routes, e.g., toward the IP prefix $w.x.y.z$, according to the following criteria. First, the IP prefix $w.x.y.z$ should be one of the prefixes reachable through the newly advertised routes by the peer L. And second, the AS-path advertised by L to reach $w.x.y.z$ should be of the form $[L, O, E, \dots]$, i.e.: a) it is not advertised as owned by L—otherwise is not a "leak", since L can advertise its own routes to V; and b) it is at least two AS hops away from L.

In this framework, if V suspects this could be the result of a route leak, then V could advertise $w.x.y.z$ back to L, that is, V could try to fool back its peer L (see Fig. 4 (b)). Once L receives the fake advertisement for $w.x.y.z$ from V, there are two options, L could either accept this route as its best path or not. If it does, then L would send a withdrawal for the route it sent earlier for IP prefix $w.x.y.z$ toward V. On reception of the withdrawal from L, V can infer that the route received earlier from L for $w.x.y.z$ was a leak—that is, it was a non-customer route received by V on its peering link with L. This is because if $w.x.y.z$ belongs to the customer cone of L, then L would have not selected the fake route sent by its peer V, since, according to our hypothesis, customer routes are preferred over peer routes. Also observe that, the decision of choosing candidate routes that are at least two AS hops away from L increases the chances of BFB to succeed, since thanks to the shortest-path principle, the Fool Back ad-

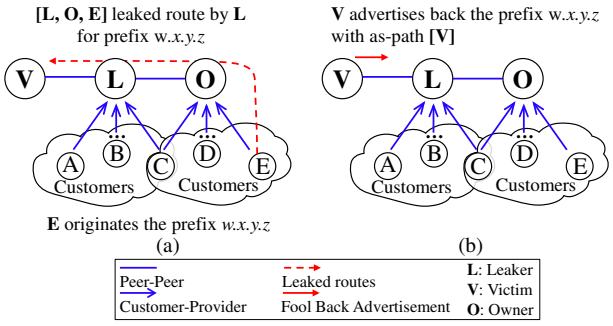

Fig. 4. Benign Fool Back: (a) L leaks O's routes to V; (b) The potential victim V sends a Fool Back advertisement to L.

vertisement $[V]$ for $w.x.y.z$ will prevail over the alternative peer route $[O, E, \dots]$ at L.

Let us now consider the example when the potential victim V initiates the BFB strategy on a false suspicion. For the case of PRL, even if V sends the Fool Back advertisement to the alleged leaker L, this would not prefer it over its legitimate customer route, and hence the fool back advertisement would stay harmless in legitimate cases—this is why we call this strategy "benign". In other words, the fool back advertisement would only poison the route for customers of L in the case that L had leaked a route to V. Also observe that once the withdrawal is received by the victim, it can start the remediation actions and withdraw the Fool Back advertisement.

As shown in Table II, we have implemented BFB and assessed its impact over all the PRL scenarios. Our results show that BFB can actually duplicate the success rate of route leak detection for PRLs. We can contend that, autonomous RLD techniques, using solely analytics on the routing information available at an AS are sufficient for detecting the large majority of the route leaks initiated by a neighbor, especially, when they are not the result of premeditated and elaborated attacks—BFB will clearly not fool a prepared attacker.

It is important to mention that, the two rational assumptions made in the route leak detection framework, including non-existence of 1) peer-peer relationships between an AS and its provider's provider, and 2) cyclic chains of customer-provider relationships, guarantees no false positives. However, the assumptions that an AS prefers a customer route over a peer or a provider route and an AS prefers shorter AS-Path for a given destination may not always be true as it entirely depends on the internal policy of an AS. Hence, in scenarios where the latter two norms are violated, the occurrence of a false positive is a possibility, however the violation of the mentioned norms will reduce the effectiveness of the route leak as well.

VI. Conclusion

In this paper, we proposed a novel approach that allows an AS to autonomously detect route leaks. Our analysis reveals that high accuracy in the detection seems feasible for customer route leaks, but for the peer case, additional mechanisms seem mandatory. In this regard, we proposed an ingenious and harmless strategy, namely, Benign Fool Back (BFB), and showed that, under realistic conditions, BFB can substantially improve the detection success rate for the peer case. The main advantages of our approach include: 1) self-contained route leak detection in real time; 2) no changes required to the BGP protocol; and 3) ease of integration into the existing inter-domain routing system, since remediation techniques may leverage the advent of SDN.

For future work we plan to explore the potential of running real-time analytics not only on the control-plane (i.e., BGP) but also on the data-plane traffic. For example, an owner can attempt to detect a route leak, i.e., the owner applies an adapted BFB, if it believes there is something suspicious after crossing data between the BGP and traffic analyses. The compound analytics approach would allow to apply BFB in broader leak scenarios, specifically CRL, aiming to achieve considerable improvements in route leak detections.

Leak Scenarios	# Leaks	RLD	RLD + BFB
CRL	2041	93.34%	93.34%
PRL	2368	38.77%	76.90%
SRL	1294	98.14%	98.14%

TABLE II
Final Results now including RLD + BFB.

Acknowledgment

This work was supported by the Spanish Ministry of Science and Innovation under contract TEC2012-34682, project partially funded by FEDER, and in part by an RFP granted by Cisco Systems, Inc.

References

[1] G. Huston, "Leaking Routes," http://www.potaroo.net/ispcol/2012-03/leaks.html, March 2012.
[2] T. Paseka, "Why Google Went Offline Today and a Bit about How the Internet Works," Nov. 2012. [Online]. Available: http://blog.cloudflare.com/why-google-went-offline-today-and-a-bit-about
[3] Y. Rekhter *et al.*, "A Border Gateway Protocol 4 (BGP-4)," IETF, RFC 4271, 2006.
[4] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001.
[5] L. Subramanian *et al.*, "Characterizing the Internet Hierarchy from Multiple Vantage Points," Berkeley, CA, USA, Tech. Rep., 2001.
[6] X. Dimitropoulos *et al.*, "AS Relationships: Inference and Validation," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 29–40, Jan. 2007.
[7] B. Ager *et al.*, "Anatomy of a Large European IXP," in *Proceedings of the ACM SIGCOMM 2012 Conf. on Apps., Tech., Archit., and Protocols for Comp. Comm.* NY, USA: ACM, 2012, pp. 163–174.
[8] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing," IETF, RFC 6480, 2012.
[9] M. Lepinski *et al.*, "A Profile for Route Origin Authorizations (ROAs)," IETF, RFC 6482, 2012.
[10] M. Lepinski, "BGPSEC Protocol Specification," draft-ietf-sidr-bgpsec-protocol, February 2013.
[11] B. Dickson, "Route Leaks – Requirements for Detection and Prevention thereof," draft-dickson-sidr-route-leak-reqts, March 2012.
[12] Q. Ju *et al.*, "Large Route Leak Detection," NANOG'49, http://www.nanog.org/meetings/nanog49/presentations/Tuesday/LRL-NANOG49.pdf, June 2010.
[13] M. S. Siddiqui *et al.*, "Route leak identification: A step toward making Inter-Domain routing more reliable," in *10th Intl. Conf. on Design of Reliable Comm. Networks (DRCN 2014)*, Ghent, Belgium, Apr. 2014.
[14] CAIDA's Archipelago Measurement Infrastructure, http://www.caida.org/projects/ark/.
[15] V. Krishnamurthy *et al.*, "Sampling large internet topologies for simulation purposes," *Computer Networks*, vol. 51-15, pp. 4284–4302, 2007.
[16] The Network Simulator - NS-2, http://www.isi.edu/nsnam/ns/.
[17] BGP++, http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/.