Detecting Bogus BGP Route Information: Going Beyond Prefix Hijacking

Jian Qiu and Lixin Gao
Department of ECE, Univ. of Massachusetts,
Amherst, MA 01003
{jqiu,lgao}@ecs.umass.edu

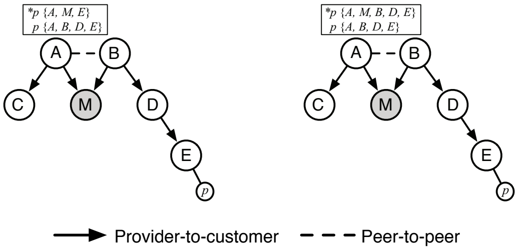
Supranamaya Ranjan and Antonio Nucci
Narus, Inc. 500 Logue Avenue,
Mountain View, CA 94043
{soups,anucci}@narus.com

***Abstract*— Border Gateway Protocol (BGP) is the *de facto* inter-domain routing protocol of the Internet. However, the BGP system has been built based on the implicit trust among individual administrative domains and no countermeasure prevents bogus routes from being injected and propagated through the system. Attackers might exploit bogus routes to gain control of arbitrary address spaces (i.e. prefixes), to either hijack the relevant traffic or launch stealthy attacks. Attackers can directly originate the bogus routes of the prefixes, or even stealthier, further spoof the AS paths of the routes to make them appear to be originated by others. We propose a real-time detection system for ISPs to provide protection against bogus routes. The system learns from the historical BGP routing data the basic routing information objects that assemble BGP routes, and detect the suspicious routes comprised of unseen objects. In particular, we leverage a directed AS-link topology model to detect path spoofing routes that violate import/export routing policies. Moreover, we explore various heuristics to infer the potentially legitimate routing information objects to reduce false alarms. The experiments based on several documented incidents show that our system can yield a nearly 100% detection rate while bounding the false positive rate to as low as 0.02%.**

I. Introduction

The Internet routing system is partitioned into tens of thousands of independently administrated Autonomous Systems (ASs). Border Gateway Protocol (BGP) [1] is the *de facto* inter-domain routing protocol that maintains and exchanges routing information between ASs. However, BGP was designed based on the implicit trust between all participants and does not employ any measures to authenticate the routes injected into or propagated through the system. Therefore, virtually any AS can announce any route into the routing system and sometimes, the bogus routes can trigger large-scale anomalies in the Internet. A canonical example happened in 1997 when AS7007 announced prefixes of a large portion of the Internet and interrupted the reachability to these prefixes for hours [2]. Moreover, bogus routes can be used to enable stealthy attacks in the Internet. For instance, spammers can announce an arbitrary prefix briefly and send spam from the hijacked address space, thereby rendering trace back to the spammer much more difficult [3]. Thus, it is important for ISPs to detect any bogus routing information in their routing system in real-time.

A sophisticated attacker may fabricate bogus routes to gain control of arbitrary address spaces (i.e., prefixes) so that the attacker can access the relevant traffic of the prefixes or even use the addresses to launch stealthy attacks. *Prefix hijacking* routes are one kind of bogus routes in which an attacker AS directly originate the routes of arbitrary prefixes. To gain access to the traffic of an existing prefix that has been announced by other ASs, the attacker can either simply originate the routes of the exact same prefix (called *duplicate-prefix hijacking*) or originate the routes of the subnets (called *sub-prefix hijacking*). Meanwhile, the unused address spaces can be hijacked to launch attacks. In this case, the attacker can hijack prefixes either entirely in unused address spaces (called *independent-prefix hijacking*) or the super-nets that cover both used and unused spaces (called *super-prefix hijacking*).



(a) Spoofing with forged AS link ME (b) Spoofing with route redistribution

Fig. 1. Examples of path spoofing

Even stealthier, an attacker can further spoof the AS paths of the bogus routes such that the routes appear to be originated by someone else while the attacker AS is actually on the paths. Thus, the attacker can perform any malicious activity that they can do with prefix hijacking routes. We refer to the route announcements as *path spoofing* routes. As shown in Figure 1(a), an attacker AS M can spoof the path by bluntly faking a nonexistent path, e.g. ME to E (called *fake-link path spoofing*). Alternatively, M can deliberately redistribute routes between its providers or peers, e.g. from its provider B to another provider A in Figure 1(b), to artificially make M a transit AS. Such routes are illegitimate since they break the commercial agreements that define the route import/export policies between M and the relevant neighbors and enable M to access the traffic that they are not legitimate to.

Bogus routes such as prefix hijacking and path spoofing routes are attractive to the attackers who wish to cover their identities but the completion of attacks relies on two-way communications. Examples are application layer DDoS attacks [4], email spams [3] or phishing scams. Although the destructive effects of bogus routes have raised serious concerns to network operators, prevention of bogus routes largely relies on ad hoc route filters. As a result, various bogus routes still keep emerging. Meanwhile, although several secured extensions of BGP, such as S-BGP [5] and soBGP [6], have been proposed, their comprehensive deployment is still unforeseeable [7]. Hence, it is imperative to provide a practical system to help network operators identify the bogus routing information and thereby to detect malicious activities associated with them.

Besides deliberate manipulation, bogus routes could be generated by BGP misconfiguration [8]. For instance, typos in the route configuration file can lead a BGP router to announce prefixes belonging to other ASs or prefixes in the unused address space. Improperly configured route filter may lead a stub AS to leak the routes learned from one provider to another provider.

Real-time bogus route detection is still a challenging and open issue given the lack of authoritative information about prefixes and ASs in the Internet. To detect prefix hijacking and path spoofing routes, we have to know the assigned/allocated prefixes in the Internet and their legitimate origin ASs, the connectivities between ASs and ASs' import/export routing policies. The WHOIS, which is a collection of routing information databases maintained by the Regional Internet Registries (RIRs), such as ARIN, and several large ISPs, might have the relevant information. Nonetheless, as it relies on voluntary contribution of ISPs to update their own records, given the cooperative nature of the Internet and its enormous size, it is virtually impossible to make the databases complete, accurate and up-to-date. For example, Signos *et al* [9] found that only 28% of ASs registered the WHOIS information that is consistent with their route announcements. Even for the most carefully maintained databases, human-induced errors and delays cannot be avoided [10]. Therefore, we cannot use the WHOIS information to do real-time bogus routes detection.

In this paper, we present a real-time bogus BGP routes detection system. The system is built based on the intuition that although BGP routes are highly dynamic, two basic routing information objects that assemble BGP routes, namely, the associations between prefixes and origin ASs, and the peering status between ASs, are relatively stable over time, and thus can be learned over time. Accordingly, the system extracts and learns the route information objects from the historical BGP routing data and then uses the knowledge to detect suspicious routes that contain unseen objects. However, the routing information objects obtained in such a simplistic learning process are neither clean nor complete. Hence, we first purify the objects by avoiding transient objects for detection. Then, based on the analysis of attacker behaviors and the common practices in network operations, we supplement potentially legitimate objects with heuristic-based inferences. We also explore the correlation among the routes triggered by identical events to calibrate the detection results. These efforts yield low false positive rates and high detection rates. Through a series of experiments, we prove the efficacy of our system in bounding the false positive rate to 0.2% which translates into around 20 alarms daily. On evaluating against several documented incidents, we also show that the system can achieve false negative rates as low as 0 for these incidents with false positive rates no more than 0.02%. Our major contributions are summarized as follows.

- Our system provides comprehensive protection against all sorts of bogus routes ranging from duplicate-prefix hijacking to sub-, super- and independent-prefix hijacking and path spoofing.
- We leverage AS topology model annotated with directed AS links to detect redistribution path spoofing in real-time without inferring AS relationships.
- We propose several heuristics to address the limitations of learning-based detection approaches by filtering transient objects as well as inferring potentially legitimate objects to improve the detection accuracy.

The rest of the paper is organized as follows. We review the related work in section II. In section III, we describe the detection system architecture. Section IV introduce the concept of routing information objects and the basic detection algorithms. We explore various heuristics to improve the detection performance in section V. Experiments and evaluations are described in Section VI. Section VII concludes the paper.

II. Related work

The Pretty Good BGP (pgBGP) [10] and the Prefix Hijacking Alerting System (PHAS) [11] are the recent work on preventing BGP prefix hijacking based on historical BGP routing data. Both systems keep track of the origin ASs of every prefix over time and identify the suspicious routes whose prefixes are originated from unknown ASs. However, they can detect duplicate- and sub-prefix hijacking routes only. Meanwhile, there are increasing evidences pointing to the use of super-prefix hijacking to send email spams [3], [12]. A savvy attacker may be expected to employ not only prefix hijacking but also stealthier path spoofing attacks. Our system provides comprehensive detection of both prefix hijacking and path spoofing. Krügel *et al* have proposed to gather route validation information through passive monitoring of BGP routing traffic to identify BGP anomalies [13], in which they exploit AS clustering based on hierarchies, i.e. either core or peripheral, and geographic locations. In a valid path, two neighboring ASs should be in the same geographic cluster and the path should traverse the core at most once. This method is computationally expensive while the obtained AS topology model is coarse. In contrast, we propose a lightweight yet sophisticated directed AS-link topology model to identify path spoofing. Meanwhile, because the information learned from history is inevitably limited, all above work suffers from high false positives. Hu *et al* [14] proposed to use active probes on data plane to improve detection accuracy. Instead, we explore

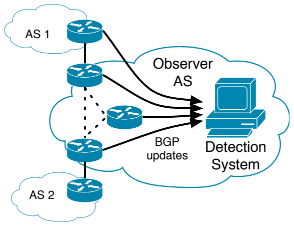


Fig. 2. System Architecture

heuristics to reduce false positives based on the control plane information only.

III. Detection System Architecture

In this paper, we present a real-time bogus BGP routes detection system based on information learned from historical BGP routing data. As a path-vector routing protocol, a route in BGP system mainly consists of a prefix, which represents the destination, and an AS path, which is a sequence of ASs along which the route is propagated from the origin to the local AS. Although BGP routes change dynamically, they are comprised of two basic *routing information objects*, i.e. the association of prefixes and their origin ASs and the AS links that represent the neighboring status between ASs. The objects represent the structure of the inter-domain routing system and are relatively stable over time. The stability comes from the stability of the inter-domain routing infrastructure — ASs have to undergo lengthy and costly procedures to request address spaces, coordinate connections and negotiate commercial agreements. Hence, an AS typically maintains its existing prefixes and its connectivity with neighboring ASs as long as possible. Thus, our detection hypothesis is that since the majority of BGP routes are believed legitimate and stable, our detection system can learn the majority of the legitimate routing information objects over time based on received routes and then use the learned information to identify the bogus routes.

Accordingly, as shown in Figure 2, the detection system peers with several BGP routers and passively receives routing data. It extracts and stores routing information objects from the received routes and in parallel examines whether the routes are bogus. Note that due to the learning-based approach, the system needs an initialization phase to accumulate enough data to build the knowledge base. After that, every route is examined as soon as it arrives and the results are presented to a network operator in real-time.

The detection system can be deployed in two scenarios. It can be deployed by a service provider, typically Tier-1, to protect its routing system. Given that routers within an AS usually have similar views, to diversify the received information, the detection system can peer with not only the routers within the deployed AS but also those in neighboring ASs. However, the routes received from the neighboring ASs should be consistent with those from the local AS. For example, routes from customer ASs should be customer routes only. The system can also be deployed as a bogus route monitoring system for the global Internet, which analyzes the BGP routing data from several public BGP data repositories such as ROUTEVIEWS [15] or RIPE RIS [16].

IV. Basic Detection Algorithm

In this section, we present a basic detection algorithm based on building a historical database of routing information objects to detect the bogus routes.

A. Collecting Routing Information Objects

A BGP route mainly consists of a prefix p and an AS path $\{a_k, \ldots, a_0\}$. a_k is the *observer AS* [1] and a_0 is the *origin AS*. The *direction of an AS path* is defined as from the observer AS to the origin AS.

From a received BGP route of prefix p with AS path $\{a_k, \ldots, a_0\}$, we extract (1) the *prefix-originAS association*, which is the tuple (p, a_0) and (2) the *directed AS-links*, which are *directional* AS pairs $a_i \rightarrow a_{i-1}, i = k, \ldots, 1$ with the same direction as the AS path. a_i is said to be the *upstream* of a_{i-1} and a_{i-1} is the *downstream* of a_i. A prefix-originAS association records the binding between a prefix and one of its origin ASs. A directed AS-link indicates that the two ASs are neighbors. More importantly, the direction encodes the import/export routing policies of the two ASs from the viewpoint of the observer AS — the downstream AS allows routes to be *exported* to the upstream AS while the upstream AS *imports* the routes from the downstream AS. At time t, the extracted prefix-originAS associations and directed AS-links during the observation window with length T, which starts at $t-T$ and ends at t, compose the sets $\mathbb{A}[t-T,t)$ and $\mathbb{L}[t-T,t)$ respectively.

B. Bogus Routes Detection Algorithm

Given, a route $(p, \{a_k, \ldots, a_0\})$, we use the procedure isBogusRoute in Figure 3 to verify its legitimacy. The algorithm first verifies the AS links sequentially in the direction of AS path and then the prefix-originAS association. It stops at the first illegitimate object and returns the object. As routes are propagated in the reverse direction of its AS path, i.e., from the origin AS to the observer AS, an AS can forge anything downstream but nothing upstream. Thus, the detection algorithm qualifies the upstream AS of a suspicious directed AS-link or the origin AS of an illegitimate prefix-originAS association as the potential attacker. Note that this procedure returns the first encountered suspicious objects only even though a route might be manipulated by several attackers and contain multiple suspicious objects.

The detection procedure is based on the *legitimate* routing information objects, which are precisely determined with procedures isLegitimateLink and isLegitimateAssociation in Figure 3, which are based on the simplistic assumption that anything seen in the past is valid at present. Thus, the

[1] Depending on type of peering session between the detection system peer and the BGP routers, the observer AS number might not appear in the AS path. In this case, the detection system should add the number.

```
isBogusRoute($p, \{a_k, \ldots, a_0\}$)
for $i \leftarrow k$ to $1$
do if $\neg$isLegitimateLink($a_i \rightarrow a_{i-1}, t$)
      then return True, $a_i \rightarrow a_{i-1}$
if $\neg$isLegitimateAssociation($(p, a_0), t$)
   then return True, $(p, a_0)$
return False

isLegitimateLink($a_i \rightarrow a_{i-1}, t$)
if $a_i \rightarrow a_{i-1} \in \mathbb{L}[t-T, t)$
   then return True
   else return False

isLegitimateAssociation($(p, a_0), t$)
if $(p, a_0) \in \mathbb{A}[t-T, t)$
   then return True
   else if $(P, a_0) \in \mathbb{A}[t-T, t) \wedge p \Subset P$
          then return True /* de-aggregation */
          else if $(p_i, a_0) \in \mathbb{A}[t-T, t) \wedge p_i \Subset p \wedge p = \cup_i p_i$
                 then return True /* aggregation */
                 else return False
```

Fig. 3. Pseudocode of detection algorithm

routing information objects in $\mathbb{A}[t-T, t)$ and $\mathbb{L}[t-T, t)$ are considered legitimate. Moreover, as an AS can aggregate or de-aggregate its prefixes, the prefix-originAS associations derived from prefix aggregation or de-aggregation can also be considered as legitimate. However, the caveat is that if a prefix P and its subnet p are assigned to two different ASs, say A and B, then the associations that are derived from the de-aggregation of (P, A) might not be legitimate since the relevant prefixes might be the subnet of p, which is in the address space of B. Thus, we introduce the term immediate subnet. A prefix p is the *immediate subnet* of a prefix P, denoted by $p \Subset P$, if no legitimate prefix-originAS association has prefix that is the subnet of P and the super-net of p.

Apparently, given that we have the "perfect" sets of legitimate routing information objects that the algorithm is based on. The sets are "perfect" in the sense that all the legitimate routing information objects that should be visible to the observer AS at the moment that the detection is perform are given, the above detection algorithm which classifies a route as bogus if it fails the presence test, must have 100% detection accuracy. However, the correctness of our detection algorithm for path spoofing routes, which requires that a valid AS path must be comprised of directed AS-links in the same direction as the path, is not as straightforward. Nonetheless, we find that the algorithm can capture any path spoofing routes with zero false positives and negligible false negatives. Please refer to our technical report for the detailed justification the correctness of the algorithm for path spoofing detection [17].

C. Classification of Bogus Routes

A route is identified as *path spoofing* if the illegitimate object found by procedure isBogusRoute is an AS-link. We further characterize it as either redistribution or fake-link path spoofing. Since the direction of a directed AS-link implies the import/export policy of the relevant ASs, if a directed AS-link is not legitimate, but its reversed counterpart is legitimate, the reversal of the AS-link direction might indicate a policy violation. Further, redistribution path spoofing routes might come along with not only reversed AS-links but also some hidden AS links that are invisible in both directions under normal configurations. Therefore, given a path spoofing path, we check all the illegitimate links. If one of them reverses direction, the route is classified as redistribution path spoofing. Otherwise, the path is fake-link path spoofing.

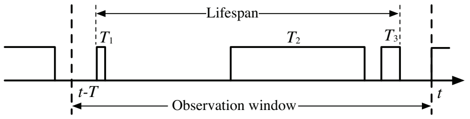


Fig. 4. Illustration of $H_o(t)$

If the AS path of a route is valid but the prefix-originAS association, say (p, a_0), is found illegitimate, the route is deemed as *prefix hijacking*. In addition, if there exists a legitimate association (p, x) with identical prefix but different origin AS, (p, a_0) is duplicate-prefix hijacking. Otherwise, if there exists a legitimate association (P, x), $x \neq a_0$, and p is the immediate subnet of P (p, a_0) is sub-prefix hijacking. Otherwise, if there exists a legitimate association (q, x), $x \neq a_0$, and q is the immediate subnet of p, (p, a_0) is super-prefix hijacking. In any other cases, (p, a_0) is deemed as independent-prefix hijacking.

V. Refining Detection Algorithm

The basis of the detection algorithm is the legitimate routing information objects that are learned from BGP routing data during the sliding window $[t-T, t)$. The quality of the objects determines the detection accuracy. However, the sets $\mathbb{A}[t-T, t)$ and $\mathbb{L}[t-T, t)$ used in our basic detection systems are directly obtained from the BGP data and may not be accurate. On the one hand, the sets are not clean. Beside legitimate objects, these sets may also contain the illegitimate ones carried by the bogus routes, which would make the future announcements of the bogus routes that contain the same objects undetectable. To eliminate these illegitimate objects, we strengthen the criteria to determine the legitimate objects. On the other hand, the sets are not complete. It is impossible to obtain all routing information objects in the Internet from history. First, the observation window and the views of neighboring routers limit the available routing information objects. Second, the Internet keeps growing while the objects are learned from the history. The new arrivals are naturally missing. To address these limitations, we properly lengthen the observation window and increase views to obtain more objects with moderate cost. Moreover, we explore various heuristics to infer additional routing information objects are are possibly hidden or new.

A. Removing Transient Routing Information Objects

Intuitively, bogus routes are naturally stealthy and thus can not be expected to last long. Accordingly, the routing objects carried by the bogus routes are very likely transient. Therefore, we remove transient objects to clean up the routing information object sets.

We first identify the metrics that measure the stability of the routing information objects. The detection system maintains a routing table $\mathbf{R}(t)$ that stores all routes from its peering BGP routers at time t. $\mathbf{R}(t)$ keeps being updated with the routing updates from the neighboring routers. We say a routing object o exists at time t if there is at least one route in $\mathbf{R}(t)$ having o. Otherwise, o does not exist. Given an observation window $[t-T,t)$, the *(accumulative) uptime* of o, denoted by $u_o[t-T,t)$, is the sum of the durations of all the periods that o exists. Further, the *lifespan* of o during the window, denoted by $l_o[t-T,t)$, is the time span when o first and last exists in the window. For example, in Figure 4, during the observation window $[t-T,t)$, the uptime is $u = T_1 + T_2 + T_3$ and the lifespan is the length of the shown interval.

We can use the uptime to redefine legitimate objects by requiring that a legitimate object should have uptime longer than a threshold θ_u in the observation window $[t-T,t)$. At the same, we can also use the lifespan by requiring that a legitimate object should have a lifespan longer than θ_l.

We apply the two criteria to the prefix-originAS associations and the directed AS-links respectively. As the uptime of an object is always no longer than its lifespan, the uptime criterion is more stringent than that with lifespan. Compared with prefix-originAS associations, directed AS-links have less visibility because the network topology and routing policies can limit the visibility of an AS-link to the observer AS. For example, a mutlihomed stub AS announces its prefixes through its primary and backup links alternatively. From the viewpoint of the observer AS, the links show up intermittently while the prefix-originAS associations of the AS appear continuously. Therefore, prefix-originAS associations are more likely persistent over time compared with directed AS-links. Thus, we apply the uptime to the former and the lifespan to the latter.

Accordingly, in the procedures in Figure 3, we should replace set $\mathbb{A}[t-T,t)$ with the refined set $\mathbb{A}'[t-T,t) = \{o|o \in \mathbb{A}[t-T,t), u_o[t-T,t) > \theta_u\}$ and $\mathbb{L}[t-T,t)$ with $\mathbb{L}'[t-T,t) = \{o|o \in \mathbb{L}[t-T,t), l_o[t-T,t) > \theta_l\}$.

B. Inferring Potentially Legitimate Objects

By analyzing the behavior of attackers and the common practices in prefixes assignment/allocation and AS peering in the Internet, we use the following heuristics to explore those possibly hidden or new routing information objects. These heuristics would be used as supplement of the procedures in Figure 3 to further justify the legitimacy of objects. The objects that are not legitimate based on the procedures but comply with some of these heuristics are said *potentially legitimate*.

1) Attacker behavior heuristics: As an attacker announces bogus routes to gain control of address spaces, if a suspicious route cannot help the attacker achieve the goal, it should not be a prefix hijacking or path spoofing route but potentially legitimate. Accordingly, the relevant objects might be legitimate.

a) Path Extension heuristic (PE): Suppose that the AS path of a prefix p is extended from the origin AS to a new AS, e.g. the AS path changes from $\{A,B,C\}$ to $\{A,B,C,D,E\}$. The route cannot let D and E access p's traffic as long as C is the legitimate origin because any traffic from A to p would stop at AS C. Therefore, this kind of routes should not be announced for malicious purposes but more likely caused by legitimate operations, such as address sub-allocation. Therefore, we consider this new route valid and the relevant new objects, e.g., $(C \rightarrow D)$, $(D \rightarrow E)$ and (p,E) are legitimate. Note that this kind of rotes might also result from misconfiguration. We find that typos in AS path prepending can cause this kind of routes. An AS, say A, typos its own number as A' when configuring prepending lists. If its own AS number is added later than prepending list, the resulting AS path will be like $\{\ldots,A,A'\}$.

b) En-route AS heuristic (EA): The ASs in the path to a prefix are called the *en-route ASs* of the prefix. Since the en-route ASs of a prefix have already had the access to the traffic of this prefix, they have no motivation to further hijack or spoof routes to this prefix. Therefore, if the AS path to a prefix contains a new directed AS-link whose upstream AS is an en-route AS of the prefix, the link should be legitimate. Similar to the previous heuristic, misconfiguration such as typos in AS prepending could result in this kind of routes.

In order to capture the legitimate en-route ASs of prefixes, we introduce a new type of routing information object called *prefix-enrouteAS association*, which is a tuple $<p,a_i>$ of prefix p and one of its en-route ASs a_i. We also use the lifespan of the associations with threshold θ_e to identify the legitimate prefix-enrouteAS associations because, similar to the directed AS-links, the prefix-enrouteAS associations also have limited visibility.

2) Common-practice Heuristics: Further, we explore several common practices that are widely adopted in the Internet to infer some reasonably hidden or new routing information objects.

a) Address Expansion Heuristic (AE): In order to optimize routing table size, RIRs try to assign ISPs new address spaces that can be aggregated with their existing prefixes [18]. Meanwhile, after ISPs obtain a large block of addresses they may initially announce part of them and then gradually announce others. Thus, an AS is likely to announce new prefixes that can be aggregated with their existing prefixes in the same "virtual" super-net. We allow an AS to expand its existing prefixes to a virtual super-net by at most 2^δ times, where δ is called *expansion factor*. New prefix-originAS associations in the expanded space are deemed legitimate.

b) Neighboring heuristic (NB): For two neighboring ASs, either of them can originate routes to the colocated prefixes. Meanwhile, an AS can sub-allocate its address space to its customers. Therefore, two neighboring ASs might be able to originate the prefixes of each other. Therefore, given two neighboring ASs A and B, if A has prefix p, B might be also legitimate to announce the route of prefix p.

c) Address Sharing heuristic (SH): On the other hand, if two ASs share prefixes they might be neighbors. The heuristic can help find some hidden links. For example, a customer AS has a subnet of the provider AS while the

AS-link between them are invisible since the provider AS announces the aggregated route instead of the more-specific route originated from the customer.

Further, since two ASs that share address space might be neighbors, according to the neighboring heuristic, the two ASs can further share other address spaces.

d) Backbone AS Heuristic (BA): The Internet backbone ASs have world-wide presence and can virtually peer with any AS. Thus any new directed AS-link from a backbone AS to another AS might be legitimate. The key to this heuristic is to identify the backbone ASs. Usually, the backbone ASs have dense connectivity. If the in-degree of an AS, i.e. the number of its upstream ASs, is more than a threshold G, it is considered as a backbone AS. We choose in-degree instead of out-degree is because the in-degree of an AS is harder to forge.

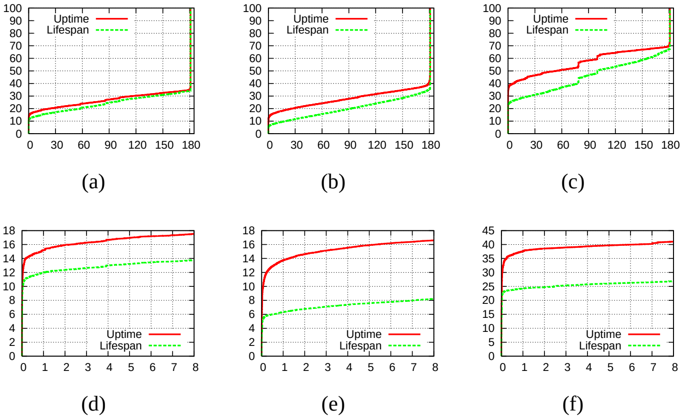
Besides the above heuristics, we can explore some other common practice heuristics to infer more objects to reduce false positives. However, applying these heuristics would introduce false negatives. For example, if we apply the neighboring heuristics, we would not be able to detect the case in which ASs hijack their neighbors' prefixes. We will use experiments to justify these heuristics.

C. Event-base Clustering and Calibration (EC)

Because the introduction of heuristics can make some actual bogus routes undetectable and introduce false negatives, we further utilize the concept of event clusters to calibrate the detection results. Because the routes triggered by the same cause likely share the same characteristics, if some of them are found bogus, others are also likely bogus even if they have been identified as potentially legitimate with certain heuristics. We use this *Event Calibration* to correct the mistakes because of overusing heuristics. We use the following clustering process to group routes into event clusters. First, the routes in the same cluster must be temporally correlated. Suppose the routes in a cluster are announced at time $t_1 \leq t_2 \leq \ldots \leq t_n$, then $t_{i+1} - t_i \leq d$ and $t_n - t_1 \leq D$, i.e., the two consecutive routes should not be spaced out more than d and the whole cluster should not span a period longer than D. The temporal clustering of BGP routing updates have been intensively studied [19], we use the typical value of $d = 70$ seconds and $D = 600$ seconds [20]. Second, routes in the same cluster share the identical cause. Since, the detection algorithm (see Figure 3) can pinpoint the possible attacker AS of a bogus route, which is either the upstream AS of the suspicious link or the origin AS of the suspicious association, hence, the routes in the same cluster should also share the attacker AS.

D. Incorporate other routing information

Besides the aforementioned measures, we also exploit other information to improve the quality of the detection results. For example, the system can incorporate a priori knowledge of bogus routes, which can be materialized as manually maintained lists of malicious or legitimate objects, to supervise the detection results. The WHOIS database can also be used as a reference in further investigation. Further, the data plane information related to the suspicious prefixes can also help further identify the malicious routes [14]. Due to our focus on the history-based approach, we will not discuss these techniques any further in this paper.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Distribution of Uptime and Lifespan (01/01/2006 - 07/01/2006), X axis shows uptime/lifespan in days and Y axis shows the CDFs; (a), (b) and (c) show the full view and (d), (e) and (f) show the zoom in view when $x \leq 8$ days.

VI. Experiments

In this section, we first use BGP routing data to investigate the values of various parameters used in the detection algorithm and validate the heuristics. Then, we evaluate the performance of the detection system.

We use the BGP routing data from ROUTEVIEWS servers in the experiments. For every BGP route, we first filter out and report the immediately apparent bogus routes such as bogon prefixes [21]. Further, even though a prefix is in the allocated address space, if its prefix length is shorter than 8 or equal to 32 [22], we also consider it a bogon. Note that the list of bogon prefixes might be different for different ASs. For example, some stub customer ASs might allow its provider ASs to announce a default route 0.0.0.0/0 plus portions of the BGP route table instead of a full BGP routing table. In the case, 0.0.0.0/0 is not a bogon prefix. For the AS path, we remove the private and unassigned AS numbers [23] from the AS path and further remove the continuously duplicate AS numbers. For example, an AS path like $\{2914, 19029, 26362, 65535, 65534, 65532, 65531, 26362\}$ will be cleaned up as $\{2914, 19029, 26362\}$. These private AS numbers are typically used within ASs but forget to be trimmed off when being exported outside. Finally, if the AS path still contains loops, the route will be filtered.

A. Determining thresholds for legitimate objects

We build the routing information object sets $\mathbb{A}$ and $\mathbb{L}$ over a period of time and investigate the settings for inferred legitimate routing information objects.

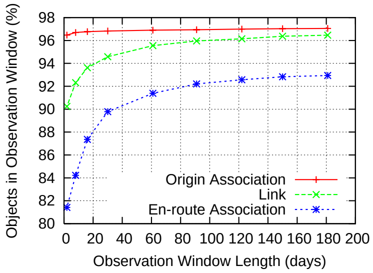


Fig. 6. Percentage of objects during July 2006 found in the observation windows of size T.

1) Thresholds for legitimate objects: Figure 5 shows the CDFs of uptime and lifespan of prefix-originAS associations, directed AS-links and prefix-enrouteAS associations in the observation window 01/01/2006−07/01/2006 based on the data from EQIX server. From the full view of the distributions in Figure 5 (a),(b) and (c), we find that majority of the routing objects are either extremely short-lived or persistent for the entire duration of the window.

The distributions suggest the proper values for the thresholds θ_u, θ_l and θ_e. If we zoom into the first 8 days, as shown in Figure 5(d), (e) and (f), and suppose that we set the thresholds to a value longer than 1 day, then the difference in percentages of objects classified as legitimate is marginal. Thus, for simplicity, we apply a 1 day legitimate threshold for all the three objects, i.e., $\theta_u = \theta_l = \theta_e = 1$ day. We also investigate the distributions for uptime and lifespan in several other observation windows with varying length. It turns out that the distributions are similar and the 1 day threshold is always a good choice.

It is worth noting that the gap between the curves of uptime and lifespan distributions for the three objects characterizes their differences in visibilities. As mentioned before, the prefix-originAS associations have better visibility to an observer and hence show up more persistently i.e., their uptime is almost equal to lifespan in most of the cases. In contrast, the directed AS-links and prefix-enrouteAS associations have a larger gap between the two distributions, which can be attributed to their intermittent visibility in that their uptime is less than lifespan in most of the cases.

2) Observation window size: Given the current time t, we try to find an observation window $[t - T, t)$ with proper size T that can account for as many legitimate objects as possible when we compare route announcements occurring in the "future" after t. Based on the routing data from EQIX, Figure 6 shows the percentage of legitimate objects in July 2006 that can be found legitimate in various observation windows that immediately preceded July and lasted from 1 day to 6 months. It shows that for the three objects, the longer the observation window, the more legitimate objects can be found. However, the growth becomes marginal when the window size is longer than $30 \sim 60$. Since, the window size should be small to save storage space, hence, we set the observation window size $T = 30$ days for all three objects.

In addition, compared with the prefix-enrouteAS associations, the prefix-originAS associations and the directed AS-links are relatively stable over time. This can be explained by the fact that these two objects represent the stable structure of the routing infrastructure while the prefix-enrouteAS associations do not.

B. Validation of Common-Practice Heuristics

In this section, we validate the common practice heuristics by examining whether they can help identify the potentially legitimate routing objects while rendering few real malicious objects undetectable. We are not going to validate the attacker behavior heuristics since we believe that as long as the attackers are rational they will not use that kind of routes to do hijacking or spoofing.

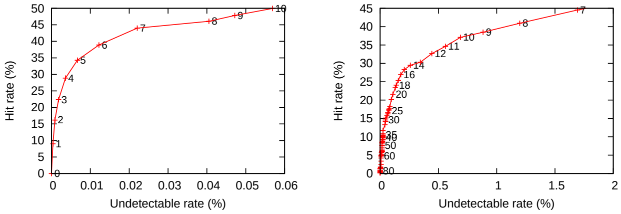
1) Validation Metrics: We validate each heuristic with the following two metrics.

a) Hit rate: The heuristics can help identify potentially legitimate routing objects from the bogus ones. Consider an object which has a time point t such that it is not legitimate based on its past history during $[t-T, t)$ but will be legitimate based on the future history during $[t, t+T)$. If a heuristic when applied against the past window, can justify the legitimacy of such a "will-be" legitimate object, the heuristic is said to *hit* the object. The percentage of the hit objects in the set of "will-be" legitimate objects is called the *hit rate* of the heuristic. Thus, the hit rate quantifies the power of a heuristic to predict the "will-be" legitimate objects.

b) Undetectable rate: However, the potentially legitimate routing information objects inferred with a heuristic might actually be bogus. For instance, if we employ the neighboring heuristic while an AS hijacks the prefixes of its neighbors, the hijacking would not be detected. The extent to which a heuristic renders bogus routes undetectable depends on how the attack is performed. We assume a *random attack model*, in which the malicious AS hijacks prefixes randomly in entire IPv4 address space or spoofs AS-links to randomly chosen ASs in the Internet. Accordingly, we define the term *undetectable rate (under random attack)* for an AS as the probability that a random attack launched by this AS becomes undetectable under a heuristic. Further, the *average undetectable rate* is the average over all ASs in the Internet.

In the rest of this Subsection VI-B, our experiments on BGP data from EQIX in March 2006 show that by effectively choosing the parameters of a heuristic, we can increase the hit rates to higher values, while not compromising on the undetectable rates too much. Moreover, in the next Subsection VI-C, we will further present detection strategies that combine these heuristics such that the detection performance improves even more.

2) Heuristics for inferring prefix-originAS associations: Next, we examine the address expansion heuristics, neighboring heuristics and address sharing heuristics, all of which infer potentially legitimate prefix-originAS associations. The heuristics actually expand the address space that an AS can legally claim. Suppose that with the heuristics, the size of



(a) Address Expansion (b) Backbone AS

Fig. 7. Average undetectable rate v.s. hit rate when applying heuristics

TABLE I

HIT RATE AND MISS RATE WHEN APPLYING DIFFERENT HEURISTICS

Heuristics	AE ($\delta = 5$)	NB	NB + SH	AE(5) + NB	AE(5) + NB + SH
Undetectable rate	0.006	0.39	0.46	0.39	0.47
Hit rate	34	31	42	55	59

address space that an AS can claim is expanded from x to x'. Assume that the AS randomly chooses an address block in the IPv4 address space to hijack, the probability that the hijacking goes undetected is $(x' - x)/(2^{32} - x)$.

a) Address Expansion Heuristic: Figure 7(a) shows the relevant change trends of hit rate and the average undetectable rate with the growth of the expansion factor δ. The points along the curve in Figure 7(a) from left to right corresponds to the expansion factor δ from 0 to 10. It shows that when the expansion factor is between 0 and around 5, the hit rate grows faster than the false negative rates and reaches around 35% while the average undetectable rates are bounded by 0.01%. However, after that, the increasing of the false negatives dominants. Therefore, we choose $\delta = 5$.

b) Neighboring and Address Sharing Heuristics: In March 2006, there are 12,814 pairs of ASs that are not only neighbors but also share address space. As a result, 59% of 21,661 pairs of ASs sharing address space were neighbors; meanwhile, 26% of 48,520 pairs of neighboring ASs shared address spaces. Here, two ASs are said to be neighbors if there is a legitimate directed AS-link from one to another; a prefix is said to be owned by an AS if the corresponding prefix-originAS association is legitimate. The observations show that the address boundary between ASs are vague. It is common that two neighboring ASs share address space of each other. Two ASs who share address spaces are very likely neighbors. Further, Table I shows the hit rate and average undetectable rate after applying different combination of the address expansion, neighboring and address sharing heuristics.

3) Heuristics for inferring directed AS-links: We examine the address sharing and backbone AS heuristic that infer directed AS-links. Suppose that a heuristic increases the number of AS-links in the Internet from x to x' and there are N ASs in the Internet. Then, the chance that a fake link is undetectable is $(x' - x)/[N(N - 1) - x]$.

a) Address sharing heuristic: As mentioned before, with the address sharing heuristic, the undetectable rate is about 0.003%. At the same time, the hit rate of the heuristic is 7.8%.

b) Backbone AS Heuristic: Figure 7(b) shows the relative growth trend between the undetectable rate and the hit rate when the backbone AS in-degree threshold G is decreasing. The lower the threshold, the more ASs are qualified as backbone ASs, the more AS links are inferred. Thus, the hit rate becomes higher but the undetectable rate is also growing. It shows that after G is no more than 14, the growth of the undetectable rate becomes faster than that of the hit rate. Therefore, we set $G = 14$, which corresponds to a hit rate of 29% and an average undetectable rate of around 0.3%.

C. Evaluation of Detection Algorithm

After exploring the parameter settings for the detection algorithm, we evaluate their performance next. We first specify the metrics for evaluation and then apply our detection algorithm to detect bogus routes under various detection strategies.

1) Evaluation Metrics: By applying the system to the BGP routing updates in a given period, we use the term *false positive rate* to represent the percentage of legitimate routes that are misidentified as bogus among all legitimate routes and the term *false negative rate* to indicate the percentage of bogus routes that are not identified as bogus among all bogus routes. Meanwhile, although the detection system reports every bogus route, the network operators actually investigate the corresponding suspicious objects since they are the "root-causes". So, each new suspicious object can be seen as an alarm. The number of alarms indicates the workload that the network operators need to perform for verification and mitigation. Accordingly, we define the *number of false alarm* as the number of legitimate objects that are misidentified as bogus and the *number of missed alarms* as the number of bogus objects that are not identified.

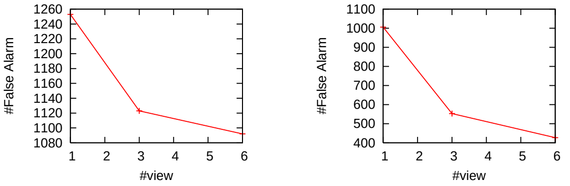
2) Detection Strategies: By applying heuristics, we are able to infer potentially legitimate routing objects to reduce false positives but increase the false negatives. In order to achieve the best trade-off, we explore the following four detection strategies in which different sets of heuristics are chosen and different levels of sensitivities are achieved: (1) *Basic Strategy (B)* in which only the legitimate objects determined by the pseudocode in Figure 3 are used for detection, which yields the highest false positives but the lowest false negatives; (2) *Rational Attack Strategy (R)* where in addition to the previous Strategy, the potentially legitimate objects inferred by the attacker behavior heuristics are also used for detection; (3) *Common Practice Strategy* where in addition to the previous strategy, the potentially legitimate objects inferred by the common practice heuristics are used for detection too and; (4) *Partial Protection Strategy (P)* based on the observation that the reported incidents in the Internet are usually duplicate-prefix hijacking and redistribution path spoofing, so while using the previous strategy, we detect duplicate-prefix hijacking and redistribution path spoofing routes only, which yields the lowest false positives but highest false negatives. Finally, to each of the four strategies, we also apply the Event Calibration to adjust the detection performance. As a result, each strategy has two sub-strategies: with or without *Event Calibration (EC)*.

3) Baseline Evaluation: We first apply the algorithm onto the BGP updates during a period, in which no prefix hijacking or path spoofing incidents was reported. With the assumption that there was no any prefix hijacking and path spoofing during the period, we can estimate the baseline false positives that our system can produce. In this experiment, we choose the BGP updates during Dec 23 through 29, 2006 from ROUTE-VIEWS. The routing information objects are inferred from the updates from six tier-1 ISPs: AS701(Alternet), AS1239 (Sprint), AS7018 (AT&T), AS2914 (NTT-America), AS3356 (Level3) and AS3549 (Global Crossing) while only the routes from AS1239 (Sprint) is inspected for bogus routes detection. Figure 8 shows the value of the four metrics achieved under different detection strategies. It shows that when the most stringent Basic Strategy is applied, around 0.5% and 1.4% of the total routes are identified as prefix hijacking and path spoofing routes and around 1092 suspicious prefix-originAS associations and 427 directed AS-links raise alarms. Thus, on average, the network operators need to verify $156 + 61 = 217$ false alarms per day. If heuristics are incorporated, the false positives are reduced. For example, if the least stringent, Partial Protection Strategy is employed, the false positive rates are reduced to 0.04% and 0.16% for prefix hijacking and path spoofing for a 90% reduction when compared with the Basic Strategy. Moreover, the number of false alarms is reduced to about 19 per day.

However, contrary to our assumption, some of the alarms raised by our system did look suspicious, which implies that our system can achieve even lower false positives. For instance, after manual inspection of the suspicious objects, we find a very likely redistribution hijacking attack on 12/27/06 in which an Indian ISP AS9498 (BBIL-AP) redistributed routes of 2703 prefixes all over the world from AS5511 (France Telecom) to AS1239 (Sprint) for almost an hour. If we consider these routes as actually bogus, then the false positive rate for path spoofing can be further reduced to 0.01%. In addition, among the reported prefix-originAS associations, we find ten very suspicious ones. For instance, a New Jersey ISP (CYBERNET, AS6073) hijacked prefixes $204.117.112.0/24$ of a Pennsylvania site (Big Brothers & Big Sisters of America, AS31906). ARIN WHOIS shows that this prefix is a "non-portable" prefix of Sprint.

To summarize, our baseline evaluation establishes the efficacy of our detection system in not misclassifying routes as bogus, since we were able to reduce the false positives for both prefix hijacking and path spoofing to values close to 0.

4) Impact of Number of Views on detection performance: We redo the previous experiments by performing the detection based on the routing information objects inferred from the routing updates of 1 view (Sprint) and 3 views (Sprint, AT%T and Alternet) respectively. Figure 9 shows the number of false alarms reported when the Basic Strategy is used. It shows when the number of view increases, number of false alarms decreases. Nevertheless, gradually the gain becomes marginal. Thus, we believe number of views around 3 $\sim$ 6 yield good enough performance with moderate cost. Also, we can observe that after the number of views change from 1 to 3, the number of false alarms for path spoofing is reduced almost half while that for prefix hijacking is reduced only about 1/10. This further confirms that unlike prefix-originAS associations, the visibility of directed AS-links can be drastically increased by increasing the number of views.



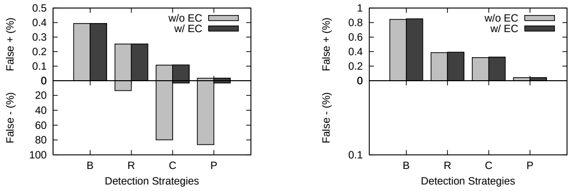
(a) Prefix Hijacking Detection (b) Path Spoofing Detection

Fig. 9. With the number of view increase, the false positive rates decrease

TABLE II
DETECTION PERFORMANCE FOR DOCUMENTED INCIDENTS

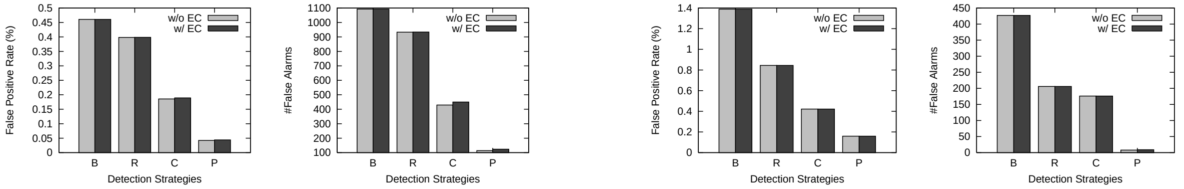
Name	Time	ASN	Strategy	+ (%)	- (%)	#F	#M
Prefix Hijacking							
TTNet [24]	12/24/04	9121	B P+EC	0.81 0.11	0 0.04	633 101	0 4
ConEd [25]	1/22/06	27506	B P+EC	0.39 0.02	0 3.2	614 57	0 10
TTNet2 [26]	2/26/06	9121	B P+EC	0.53 0.07	0 0	470 48	0 0
NWNet [27]	6/7/06	23520	B P+EC	0.37 0.02	0 0	822 61	0 0
Path Spoofing							
IS-AP-HK [28]	11/11/04	9729	P P+EC	0.85 0.04	0 0	335 11	0 0
PK-TEL [29]	12/12/05	17557	B P+EC	1.00 0.01	0 0	532 15	0 0

5) Evaluation with Documented Incidents: In this section, we apply our detection system to several documented incidents in 2004∼2006 as the "ground truth" to examine the detection performance. The detection is based on the routing updates from EQIX during the three-day period around the incidents. Table II shows the date and the attacker ASs of 4 prefix hijacking incidents and 2 path spoofing incidents. The detection performance is depicted by the metrics under the most stringent Basic Strategy (B) and the least stringent Partial Protection + Event Calibration Strategy (P+EC). Comparing the two strategy, it shows that the heuristics can help reduce the false positives by almost 90% while keeping the false negatives close to 0. In particular, Figure 10(a) shows the detection performance for ConEd incident. Unlike other prefix hijacking



(a) ConED (b) IS-AP-HK

Fig. 10. Prefix Hijacking Detection Performance for different incidents



(a) Prefix Hijacking Detection

(b) Path Spoofing Detection

Fig. 8. Baseline detection performance during December 23 through 29, 2006

incidents, in which the attackers hijacked fractions of prefixes in the Internet and could be modeled as random attacks, the attacker in ConEd hijacked prefixes of its customers mainly. Thus, the common practice heuristics do not identify more than 80% hijacked routes. Fortunately, the Event Calibration heuristics corrects most of the misidentification and adjusts the false negative rate to 3.2%. In the two path spoofing incidents, both ISPs in Hong Kong and Pakistan leaked routes of google.com and other sites to the Internet backbone and made large portion of the Internet experience huge latencies for half an hour. Figure 10(b) shows that the detection system was able to detect the spoofing routes under all detection strategies.

VII. Conclusions

This paper proposes a bogus route detection system via a persistence-inference method on route information objects. Even though the BGP routes are changing highly dynamically, the routing system has a relatively stable structure. We capture the routing objects that represent this stable structure, such as the association between prefixes and origin ASs and the AS-links, and use the knowledge to detect the bogus BGP routes. In particular, the system is able to detect path spoofing with the aid of directed AS-links without knowing AS relationships. Further, in order to address the inherited shortcoming of the history-based approach, we take several measures to improve the detection knowledge base. First, we filter out transient objects. Second, we explore heuristics to infer the potentially legitimate routing objects. Further, we calibrate the detection results based on the intuition that the routes triggered by the same events share the same characteristics. Finally, experiments show that our system can achieve false positive rates as low as 0.02% and raise no more than 20 alarms per day. Further, we show that our system can detect bogus routes of several documented incidents with almost 100% detection rate and about 0.02% false positive rates.

References

[1] Y. Rekhter, T. Li, and S. H. Ed., “A Border Gateway Protocol 4 (BGP-4),” IETF, RFC 4271, January 2006.

[2] “Wow, AS7007!” http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html.

[3] A. Ramachandran and N. Feamster, “Understanding the Network-Level Behavior of Spammers,” in *Proceedings of ACM SIGCOMM*, Pisa, Italy, September 2006.

[4] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, “DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection,” in *Proceedings of IEEE INFOCOM*, Barcelona, Spain, 2006.

[5] S. Kent, C. Lynn, and K. Seo, “Secure Border Gateway Protocol (Secure-BGP),” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582 – 592, April 2000.

[6] R. White, “Architecture and Deployment Considerations for Secure Origin BGP (soBGP),” IETF, Internet draft draft-white-sobgp-architecture-01, May 2005.

[7] H. Chan, D. Dash, A. Perrig, and H. Zhang, “Modeling Adoptability of Secure BGP Protocols,” in *Proceedings of ACM SIGCOMM*, Pisa, Itality, September 2006.

[8] R. Mahajan, D. Wetherall, and T. Anderson, “Understanding BGP Misconfiguration,” in *Proceedings of ACM SIGCOMM*, August 2002.

[9] G. Siganos and M. Faloutsos, “Analyzing BGP Policies: Methodology and Tool,” in *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.

[10] J. Karlin, S. Forrest, and J. Rexford, “Pretty Good BGP: Improving BGP by Cautiously Adopting Routes,” in *Proceedings of IEEE ICNP*, November 2006.

[11] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, “PHAS: A Prefix Hijack Alert System,” in *Proceedings of 15th USENIX Security Symposium*, 2006.

[12] J. S. Sauver, “Route Injection and Spam,” Toronto, Ontario, Canada, October 2006.

[13] C. Krügel, D. Mutz, W. K. Robertson, and F. Valeur, “Topology-Based Detection of Anomalous BGP Messages,” in *RAID*, 2003, pp. 17–35.

[14] Xin Hu and Z. Morley Mao, “Accurate Real-time Identification of IP Prefix Hijacking,” in *Proceedings of IEEE Security and Privacy*, Oakland, 2007.

[15] “Route Views Project Page,” http://www.routeviews.org.

[16] “RIPE Route Information Service,” http://www.ripe.net/ris/index.html.

[17] J. Qiu, L. Gao, S. Ranjan, and A. Nucci, “Detecting Bogus BGP Route Information: Going Beyond Prefix Hijacking,” Department of ECE, University of Massachusetts, Amherst, Tech. Rep. TR-07-CSE-04, June 2007.

[18] “Policies for IPv4 address space management in the Asia Pacific region,” APNIC, Tech. Rep. APNIC-086, December 2005.

[19] D.-F. Chang, R. Govindan, and J. S. Heidemann, “The Temporal and Topological Characteristics of BGP Path Changes,” in *ICNP*, 2003.

[20] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, “Finding a Needle in a Haystack: Pinpointing Significant BGP Routing Changes in an IP Network,” in *Proceedings of Networked Systems Design and Implementation*, May 2005.

[21] N. Feamster, J. Jung, and H. Balakrishnan, “An Empirical Study of ”Bogon” Route Advertisements,” *ACM SIGCOMM Computer Communications Review*, January 2005.

[22] B. Woodcock, “Best Practices in IPv4 Anycast Routing,” 2002, http://www.pch.net/resources/papers/ipv4-anycast/ipv4-anycast.ppt.

[23] “AUTONOMOUS SYSTEM NUMBERS,” http://www.iana.org/assignments/as-numbers.

[24] “Anatomy of a Leak: AS9121 (or, ”How We Learned To Start Worrying and Hate Maximum Prefix Limits”),” http://nanog.org/mtg-0505/underwood.html.

[25] “Con-Ed Steals the ’Net,” http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml.

[26] “Pretty Good BGP and the Internet Alert Registry,” http://www.nanog.org/mtg-0606/karlin.html.

[27] J. Karlin, “a fun hijack: 1/8, 2/8, 3/8, 4/8, 5/8, 7/8, 8/8, 12/8 briefly announced by AS 23520 (today),” http://www.merit.edu/mail.archives/nanog/2006-06/msg00082.html.

[28] “Goofle/Sprint having problems?” http://www.cctec.com/maillists/nanog/current/msg05514.html.

[29] “www.google.com latency/packet loss/very slow thru savvis,” http://www.merit.edu/mail.archives/nanog/2005-12/msg00219.html.