Efficient Techniques for Detecting False Origin Advertisements in Inter-domain Routing

Sophie Y. Qiu, Fabian Monrose, Andreas Terzis, and Patrick D. McDaniel*
Dept. of CS, Johns Hopkins University *Dept. of CSE, Pennsylvania State University

Abstract

The Border Gateway Protocol (BGP), and hence the Internet, remains critically vulnerable to a range of prefix forgery attacks. In this paper, we address these attacks by proposing a non-cryptographic, incrementally deployable mechanism to probabilistically detect forged BGP origin advertisements. Upon receiving an advertisement from a "suspicious" origin, the receiving domain intelligently probes other ASes about the received information. Any dissenting information indicates potential forgery or error, and is reported by the polled ASes to the true origin and processed appropriately. In this design, we exploit the fact that the highly connected AS topology makes it difficult to block the dissemination of information as it traverses the Internet. We evaluate the effectiveness of our probing mechanism via simulation on realistic Internet topologies. The experiments show that 98% of forgeries can be detected even when as few as 10% of the ASes participate in the protocol under a naïve polling stratagem. Moreover, we show that judicious node selection can further improve detection rates while minimizing the number of probes.

1 Introduction

The Border Gateway Protocol (BGP) controls how Internet traffic is routed [1]. However, the protocol is vulnerable to a range of route and prefix forgery attacks. In addressing these threats, one must ensure the validity of both the paths and the prefix origins. This work is concerned with the latter: *how can the routing system detect false origin advertisements received via BGP?*

An origin is forged when an AS other than the legitimate/authorized owner incorrectly advertises a prefix either due to misconfiguration [2] or malicious attack [3]. A number of approaches have been proposed to address this origin authentication problem [4, 5, 6, 7], but require cryptographic machinery and often significant router state. These costs are seen as a significant barrier to adoption in environments where router resources are already stretched thin.

We address the limitations of past solutions by introducing a non-cryptographic, incrementally deployable mechanism that probabilistically detects false BGP origin advertisements. Our technique is based on the following observation: the highly connected nature of the Internet's AS topology makes it prohibitively difficult to block *all* routing announcements originated by a valid source.[1] Consider an AS that falsely advertises a prefix p. As long as the true origin AS is active, then some set of ASes, call them $\mathcal{T}$, will accept the correct origin advertisement. Once any node in $\mathcal{T}$ is contacted, the incorrect route can be forwarded to p's true origin. At that point, steps to rectify the conflict can be initiated. The true origin has strong incentives for ensuring the correct use of its address space, and hence is likely to expend all reasonable efforts to ensure proper resolution of the conflict. Note that the present work concentrates on conflict detection only, leaving deep consideration of their resolution to future work.

This paper considers the design, operation, and efficacy of our novel probabilistic algorithm for detecting forged prefix origins. We begin by providing a detailed description of the protocol and its design trade-offs. A number of solution metrics are introduced, and a detailed evaluation of naïve and sophisticated operational strategies is presented. Simulations of the current Internet topology show that even when only 10% percent of the ASes deploy the protocol, the true origin is notified of the false advertisement in 98% of all cases—even when the queried ASes are randomly selected. Further experiments show that enhanced AS selection strategies can further improve the detection capability while reducing the overall polling cost.

The remainder of this paper is structured as follows. We begin by first providing a high-level overview of the proposed mechanism. In Section 3 we present our evaluation methodology and provide our metrics for examining the effectiveness of our approach when applied to real Internet topologies. We present a set of candidate AS selection strategies in Section 4 and discuss their relative merits in improving overall performance. In Section 5 we address issues related to query initiation. Related work is presented in Section 6, and we end with some closing remarks in Section 7.

[1] We disregard the pathological case in which the malicious AS acts as the single upstream node of the origin AS O, since in this case the upstream node does not need to send a false advertisement to blackhole all of O's traffic; it can simply drop the packets before they travel downstream.

2 High-Level Overview

To better clarify the context of our proposed solution, in the following discussion we consider a scenario where some arbitrary AS, denoted as D, receives an announcement for prefix p that is falsely advertised by AS O'. The main problem we explore is how to notify the true origin that some other AS is wrongly advertising the same prefix. In the case where the announcement is indeed fraudulent (*e.g.*, represents a *hijacked* prefix [2]), the true owner can take the necessary corrective measures.

The attractiveness of this approach is that domains receiving multiple advertisements do not have to decide which one is the authentic one; they simply report back to the origin(s) and let them remove the fraudulent advertisements from the network. In this way, route announcements do not need to be protected by cryptographic means and the burden of removing malicious announcements falls to the most interested party in the system—the source of the valid prefix announcements. The basic mechanism works as follows:

- *Stage 1: Query initiation.* D decides whether to trigger the detection mechanism for the received advertisement.
- *Stage 2: Node selection.* Upon deciding to initiate queries, select a number of target ASes B_1, ..., B_n, and issue queries to each regarding prefix p. As an optional step, D may delay propagating the received route to neighboring ASes.
- *Stage 3: Notify origin(s) of potential conflict.* Each targeted node, B_i, checks which source it currently believes owns prefix p. If that source differs from O', then B_i forwards the query to the origin it uses. Note that B_i does not need to know it uses the legitimate origin, it simply forwards the path it receives from D, if D uses a different origin. (Optional: B_i can also notify D that it uses a different origin (*e.g.*, AS O).)
- *Stage 4: Detect conflicts and take corrective measures.* If any of the targeted nodes used a different origin for p, then that origin is notified that another AS is advertising its prefix. At that point, the true owner can take corrective measures.

We illustrate the detection technique through the example given in Figure 1. Upon receiving an announcement for address prefix p which it deems suspicious, D initiates the detection process by sending the path it uses to ASes B_1, B_2 and B_3. Since B_2, and B_3 use a different origin (AS O which is the true origin) for prefix p, they forward the query to O. In that way, O is notified that AS O' is wrongly advertising the same prefix and it can take further actions.

We reiterate that the main goal of the protocol is to detect the false origin advertisement. How (and what) corrective measures are taken is beyond the main scope of this paper. However, we note that a few options for integrating effective responses into the above protocol may be used. For example, if during *Stage 3*, B_i notices that the origin it uses for p differs from O', it can reply directly to D with the relevant information about O. In this way, network operators or automatic network management systems can be alerted, and can further explore the problem, *e.g.*, by actively probing and monitoring how the traffic flows, and filtering incorrect routes. Furthermore, should a certificate infrastructure that provides proof of identity, ownership, and authorization become available (see S-BGP [4], SoBGP [5], or Origin Authentication Service [6]), then the real origin AS O can generate an attestation to resolve any conflicts.

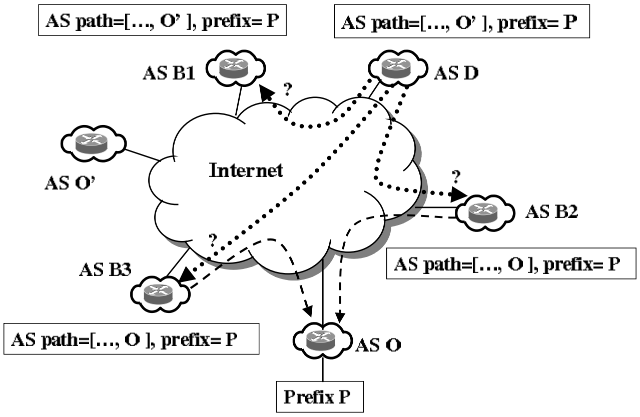
AS path=[…, O'], prefix= P
AS path=[…, O'], prefix= P
AS B1
?
AS D
Internet
AS O'
?
AS B2
AS path=[…, O], prefix= P
AS B3
?
AS path=[…, O], prefix= P
AS O
Prefix P

Figure 1. Detecting false advertisements.

3 Evaluation

We now explore the practical utility of the proposed protocol. Since the effectiveness and efficiency of such a mechanism are closely related to the size and structure of the AS graph, we perform simulations based on the topology of the Internet. We examine the effectiveness of the proposed mechanism in successfully notifying the true origin of the false advertisements, which implies that at least one AS succeeds in contacting the node with the correct information. We further explore the robustness and efficiency of the detection mechanism, examining the number of ASes that succeed in their probing efforts and the number of probes attempted before arriving at a node with the correct information.

3.1 Simulation Methodology

In order to create a simulation environment similar to the real Internet, our first step is to build an AS graph based on the real Internet topology. However, not all paths on such a graph are feasible. Instead, the number of paths connecting any two points on the AS graph are constrained by *routing policies*. Such routing policies are dictated by the commercial agreements between administrative domains, so AS relationships play an important role in determining how traffic flows in the Internet.

Since public information regarding AS relationships is not readily available, we adopt Gao's work [8] on heuristically inferring AS relationships from patterns in routing table entries. To do so, we first build an AS topology graph using multiple BGP routing table snapshots captured by Routeviews [9]. Next, we apply Gao's heuristics to generate an annotated partially-directed graph, where the nodes represent ASes and the edges are labeled based on inferred relationships of the edge nodes, *i.e.*, *provider-customer*, *peer-peer*, and *sibling-sibling*. The inferred AS relationships translate into basic BGP policies, which are followed in the route propagation and selection process. In what follows we discuss how we simulate route propagation and select preferred routes.

The propagation of routes advertised by origins (both the real and malicious) is simulated by randomly picking two nodes assuming one is the valid origin and the other is malicious.[2] When one node is the sole provider of the other node in the pair, the AS pair is filtered as the traffic to one node always goes through the other. Route propagation is then launched on the annotated partially-directed AS graph starting from both origin ASes, following the BGP export policy. In this way, the propagated AS path always follows the *valley-free* pattern, that is, after a provider-to-customer or peer-to-peer edge, the AS path will not traverse another customer-to-provider or peer-to-peer edge.

In general, route selection in BGP is complex and typically involves a reliance on a number of factors including business relationships and local administrative policies. As it is difficult—if not intractable—to model these behaviors, we simply adopt an acceptable modus operandi where customer-learned routes are preferred over peer-learned or provider-learned routes, and routes with the shortest AS-path length are preferred. For each node in the topology graph, if both routes (for the same prefix) advertised from the two ASes are received, we select the winner based on the above criteria. In the case where the metrics are the same, we take the outcome of a random coin toss to determine the chosen route.[3] We refer to the nodes that select the route advertised by the malicious origin as *origin-deceived* nodes, or simply *deceived*. Similarly, we call the nodes that select the route advertised by the real origin as *authentic* nodes.

3.2 Evaluation Metrics

The remainder of this section discusses our metrics for evaluation. These metrics capture the effectiveness of the mechanism in successfully notifying the true origin of the false advertisements, as well as the robustness and efficiency of the protocol from the perspective of the initiating AS.

3.2.1 Effectiveness

Assume that there are N ASes in the Internet, a small fraction d of which deploy the proposed protocol, and that queries are triggered with probability q if an AS considers the received advertisement suspicious. (We defer the discussion of how an AS may decide whether a received advertisement is suspicious until Section 5.) Moreover, denote the fraction of deceived nodes by f, and assume that any given AS makes a maximum of m probes. Assume further that the ASes that deploy the protocol are uniformly distributed among both the deceived and authentic nodes. Therefore, the number of ASes that would potentially initiate queries is $\delta = Nfdq$. Similarly, the percentage of ASes that have the correct information about the origin and will respond to other ASes' queries is:

$$H = (1 - f) \cdot d$$

Likewise, the probability that all m probes made by a single AS fail to reach any authentic AS is simply:

$$P_{fail} = (1 - H)^m$$

Therefore, the probability that the detection succeeds (*i.e.*, at least one node successfully contacts an authentic AS which in turn notifies the true origin of the false advertisement) is:

$$P_{Succ} = 1 - P_{fail}^{\delta} \tag{1}$$

As an example, consider a graph that contains 23,153 ASes.[4] Suppose that 5% of the nodes in the AS graph deploy the protocol and queries are triggered with probability 10%. Then, in the case where a querying node is restricted to issuing a maximum of $m = 5$ probes and that a malicious origin has already deceived 50% of the nodes, the probability that the valid origin is notified of the false advertisement would be 99%.

To illustrate the effectiveness of the proposed mechanism when applied to the Internet, we randomly generate 5000 different AS pairs, assuming for each pair one AS is valid and the other malicious. We then simulate route propagation process starting from the two ASes along the annotated graph. Again, we apply the detection mechanism assuming an AS makes a maximum of $m = 5$ probes. We examine the effectiveness of the detection mechanism assuming that some proportion, d, of the ASes implement the protocol, and that queries are initiated with probability q. In particular, in every group of experiments, only ASes belonging to that set of participating ASes will initiate or respond to queries.

[2] In all reported experiments, we repeat this process multiple times ($\geqslant$1000) using random nodes to ensure the results are not tied to any particular topological characteristic.

[3] The cases in which metrics tie account for only a small percentage, *e.g.*, no more than 5% in general.

[4] We obtained the number 23,153 based on routing table snapshots captured in May 2006 at Route Views.

Deployment (d)	Query prob. (q)	Detection rate
100%	100%	100.00%
50%	50%	99.97%
50%	10%	99.77%
50%	5%	99.38%
10%	50%	98.95%
10%	10%	95.29%
10%	5%	92.36%
5%	50%	96.28%
5%	10%	85.75%
5%	5%	72.17%

Table 1. Success rate ($m = 5$).

The results in Table 1 show that the detection mechanism is very effective even with lightweight probing. Moreover, even when only a small percentage (*e.g.*, 10%) of ASes deploy the protocol, and queries are initiated with low probability (*e.g.*, 5%), the detection mechanism still achieves a success rate of over 90%.

The significantly high detection rate implies that in a majority of situations *at least one* node succeeds in contacting an authentic node. In the following section we pay close attention to characterizing the percentage of ASes that do in fact succeed in probing authentic nodes. A high ratio of such nodes would suggest that the detection mechanism is robust, implying that this task can be accomplished effectively.

3.2.2 Determining the Hit-ratio

Ideally, queries should be first initiated at location(s) close to where the incorrect information starts to propagate. In the best case, the neighboring ASes can successively probe other ASes and the invalid announcement can be detected quickly (and thus not propagate further). We now examine the ratio of ASes that succeed in their probing efforts. In the next section we examine the number of probes attempted before arriving at a node with the correct information. Clearly, a high ratio of ASes succeeding in probing with a small number of attempts would suggest that the detection mechanism is indeed robust and efficient.

In the following experiment we generate 1000 different AS pairs at random and simulate route propagation as before.[5] We then examine the number of deceived nodes—*i.e.*, nodes that select the route advertised by the malicious origin—and calculate the ratio of such nodes out of all ASes. We denote this as the *deceived ratio*. We then apply the detection mechanism under several parameters and examine the ratio of nodes that succeed in their probing efforts. We call this ratio the *hit ratio*.

Figure 2 depicts the hit ratio for two groups of experiments where a node (to satisfy its suspicion) can only randomly query a maximum of $m = 5$ or $m = 10$ nodes. The results suggest that the detection mechanism is robust in that for the majority of deceived ratios the hit ratio is substantial. Not surprisingly, the hit ratio is closely related to the deceived ratio, in that the fewer nodes that believe the false announcement, the higher hit ratio we attain.

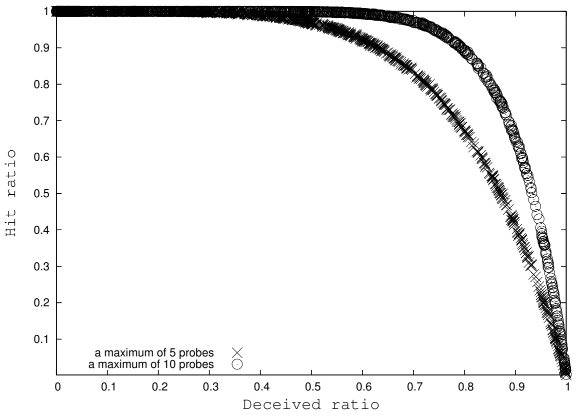

Figure 2. Hit ratio vs deceived ratio

The plots show that when the percentage of deceived nodes is not excessive, *e.g.*, less than 60%, one can attain a hit ratio of over 90%, by randomly probing a small number of nodes (< 5). This implies that in cases where the malicious origin has not deceived an overwhelming number of ASes, lightweight probing is very efficient. Moreover, in cases where the deception is widespread (say over 80%) we can still achieve a hit ratio of roughly 90% by allowing no more than 10 random queries by suspicious nodes.

It would appear that in some cases the deceived ratio is exceedingly high. However, we expect that the actual number of authentic nodes is likely far greater than that in our simulations. This is because most ASes adopt different levels of protective measures themselves (*e.g.*, the use of route filters and not accepting advertisements from customers) and so route selection is normally more intelligent than what we adopt here. In addition, our simulations implicitly assume that by the time queries are initiated the incorrect information has already disseminated to the entire Internet. However, in reality, one can expect that queries will be first initiated at the point(s) close to where the incorrect information starts to propagate before it has the chance to spread across the Internet. Therefore, situations in which most ASes are deceived should be uncommon.

3.2.3 Number of Required Probes

A natural question that arises is how large must m be in order to achieve a high detection rate. Clearly, it is desirable to have a node contact an authentic node by issuing as few probes as possible. Therefore, understanding query efficiency plays an important role in evaluating the performance of the proposed mechanism, and helps in providing guidance on how many probes an AS should attempt in general.

[5] For comparison purposes, we use the same 1000 origin pairs in subsequent experiments unless otherwise specified.

To answer that question, we examine the actual number of attempts made by querying nodes before contacting an authentic node. As an example, in the previous simulation, 5,923,863 out of a total of 8,321,420 cases (*i.e.*, 71%) resulted in success, 47% of which succeed after the first attempt, 24% more after the second, and 14% after the third. This implies that the proposed detection mechanism is efficient in that most cases only require a small number of probes to detect the conflict. In the next section, we explore this question in more detail and present strategies to further improve performance while reducing the number of required probes.

4 On Improving Node Selection

The performance of our algorithm is greatly affected by the selection of ASes to be queried. The previous analysis used randomly selected ASes. Next, we explore strategies for selecting the target AS for our queries. The following discusses these strategies and provides a quantitative comparison of the algorithm under each approach.

We propose the following AS selection strategies:

- Strategy I: Restricting queries to nodes in particular levels of the AS hierarchy. Clearly, AS connectivity, relationships, and the associated routing policies play key roles in determining how Internet traffic flows. To some extent, these factors are incorporated in the work of Subramanian *et al.* [10] which groups ASes to five hierarchy levels, namely dense core (tier 1), transit core (tier 2), outer core (tier 3), small regional ISPs (tier 4) and customers (tier 5). In general, the ASes at high levels of the hierarchy (*e.g.*, tier 1) represent part of the core of the Internet, have high degree of connectivity, and are closer to more ASes than those ASes at low hierarchy levels (*e.g.*, tier 5). Obviously, the core ASes tend to have a relatively complete view of the Internet and are therefore less likely to be deceived by malicious advertisements. Given this knowledge, a natural strategy may be to select nodes randomly from ASes of certain hierarchy levels, instead of from the entire AS set.

- Strategy II: Querying nodes far from oneself. With all other metrics being equal, the shortest AS-path is always preferred in BGP. Therefore, if a node selects the route advertised by the malicious origin, that node is likely closer to the malicious origin than to the real origin. One may suspect that its neighbor ASes are also in danger of being close to the malicious AS, and so likely to select the incorrect route as well. A natural strategy is therefore to avoid querying neighboring ASes. This strategy assumes that an AS can have an AS topology graph or a rough approximation thereof, *e.g.*, derived from the routing tables, and can infer the approximate distance of a node to itself.

- Strategy III: Querying nodes far from the suspicious origin. Similarly, an alternative strategy is to avoid querying ASes that are close to the suspicious origin, since if an AS advertises false information, the ASes close to it are more likely to believe the route as the advertised AS path they receive would tend to be short.

- Strategy IV: Querying nodes far from the suspicious origin and from oneself. An alternative strategy is simply to combine strategies II and III, avoiding querying ASes that are either close to the suspicious origin or to oneself.

- Strategy V: Avoiding querying downstream nodes. Since downstream nodes reach other parts of the Internet through the upstream nodes, then a natural strategy for a given node might be to avoid querying its downstream nodes.

We evaluate these selection strategies under the same 1000 origin pairs used in Section 3.2.2.[6] Note that in this section the simulations are performed under the assumption that all ASes participate in the protocol and queries are initiated with probability 100%. The purpose is to reduce randomness introduced by other parameters while exploring different node selection strategies.

AS Selection Strategy	Better	Same (ratio=1.0)	Worse
Strategy-**I** (tier 1)	452	195 (193)	353
Strategy-**I** (tiers 1-2)	426	169 (166)	405
Strategy-**I** (tiers 1-3)	461	178 (168)	361
Strategy-**I** (tiers 1-4)	465	186 (172)	349
Strategy-**II** (> 4 hops)	622	194 (192)	184
Strategy-**III** (> 4 hops)	790	196 (195)	14
Strategy-**III** (> 5 hops)	758	196 (196)	46
Strategy-**III** (> 6 hops)	676	197 (197)	127
Strategy-**IV** (> 3 hops)	751	194 (192)	55
Strategy-**IV** (> 4 hops)	758	196 (195)	46
Strategy-**V**	416	160 (150)	424

Table 2. A comparison of AS selection strategies with random selection.

Our results are summarized in Table 2. Each experiment shows the number of times (out of the 1,000 AS pairings) that the strategy performs *better*, the *same* or *worse* than random selection. In the cases that they attain the same ratio, the tie occurs mostly because random probing already achieves a hit ratio of 1 and thus leaves no room for improvement.[7]

[6] We perform experiments using the same 1000 origin pairs for all the strategies. For each pairing, we compare the results of the strategies in question with that using random selection.

[7] The number in the parentheses of Column 3 shows the times they both attain a hit ratio of 1.

It appears that in some cases under strategy **I** we can attain an improved hit ratio, while for others, this approach yields little (if any) improvement. In particular, when the node candidate pool is extended to include more ASes at lower hierarchy levels, we achieve results similar to that attained by randomly querying ASes from the entire AS set. This makes sense as there are far more ASes at lower hierarchy levels (*e.g.*, tier 5). In contrast, this strategy does not lead to guaranteed improvements over random probing because the hierarchy classification does not characterize topological relationships of ASes with sufficient accuracy. For example, in many cases, a tier-5 AS may be directly connected to a tier-1 AS, which places the tier-5 AS at a comparable status with other higher level ASes that also directly connect to that tier-1 AS.

Most other strategies offer improved hit ratios when compared to naïve node selection. However, the most significant improvement is achieved under strategy **III**—*i.e.*, querying nodes far from the suspicious origin. By contrast, strategies **IV** yields no noticeable improvement over adopting strategies **II** and **III** separately. One reason for this is that both **II** and **III** strategies independently gain substantial improvement on their own, and so the additional benefit from adopting both is minimal. Moreover, in many cases the pool of candidate nodes is severely limited under this combined strategy and thus offers no clear benefit. Lastly, strategy **V** appears to be of little added value. This is because the majority of ASes are of a low hierarchy, and therefore have a small number of downstream nodes. Consequently, for those ASes, the overall impact will be minimal.

In general, an effective strategy is to query those ASes that are as far as possible from the suspicious origin. However, one also needs to guarantee that there are enough candidates to query. For example, if we limit the candidate pool to those that are more than 6 hops away, in most cases we achieve a hit ratio of 1.0. However, in some cases we attain a low hit ratio and the improvement is actually less than that of limiting the candidate pool to those more than 5 hops away (see Table 2). In most of these cases this is due to having few (if any) ASes to query.[8] A fallback strategy when this situation arises, denoted as Strategy-**III-A**, is to extend the pool of potential targets to those that are for example, more than 5 hops away.

As discussed in Section 3.2.3, we are also interested in the average number of probes an AS must attempt before contacting an authentic node. Table 3 shows the distribution of the probes attempted upon success for the case of $m = 5$ for the best strategy (namely, **III**). As depicted in Table 3, when nodes are randomly selected from the entire AS set—out of total 8,321,420 deceived cases across 1000 sets of experiments—5,923,863 (71.19%) successfully contact an authentic node, out of which 47% succeed after making a single query. When the AS candidate pool is limited to those more than 4 hops away from the suspicious origin, for the queries that succeed, 55% do so after the first attempt. Likewise, restricting the pool of candidates to nodes 6 hops away, results in 66% being successful after the first attempt.

Note that the number of probes necessary to reach an authentic node is reduced by the successful selection strategies. Figure 3 illustrates the overall success ratio of probing using different number of probes. For example, achieving a 70% success ratio under the naïve strategy would require 4 probes, but that is reduced to 2 (50%) when Strategy-**III** is applied and restricted to nodes 5 hops away.

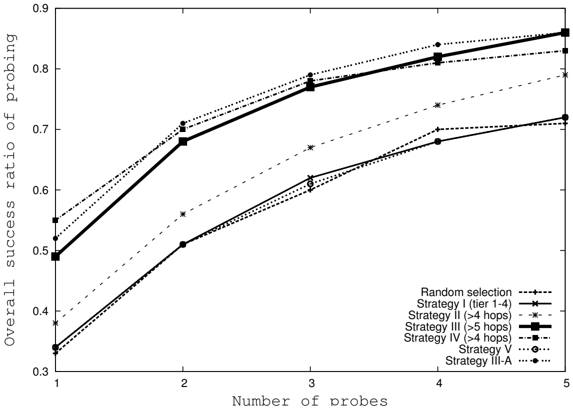

Figure 3. Overall success ratio of probing as a function of the number of probes.

5 On Query Initiation

Deciding when to initiate queries is an important issue. Since an AS typically receives a large number of BGP updates, out of which few are likely due to malicious intent or misconfiguration, it would be desirable if the AS did not need to verify each announcement. In particular, recent work [11] has shown that a significant portion of prefixes have high origin stability. In fact, that study showed that origin changes account for less than 2% of the BGP update traffic, with more than 90% of the prefixes being consistently originated by the same AS for an entire year.

The results of Qiu *et al.* [11] indicate that information based on historical data could be fairly reliable. Therefore, one potentially useful method for an AS to decide when to initiate queries might be to create a list of known (and trustworthy) origin ASes based on historical BGP data (i.e., from BGP table snapshots, or BGP updates). Then, upon receiving an announcement, the AS simply checks its list and raises an alarm if the origin AS is not one of the "legal" origins for this prefix. At this point, the AS can make use of one of the strategies outlined earlier in order to efficiently inquiry about the announcement (and notify the true origin in case of a conflict). Note that in this case, it is unwise to simply query the known origins in the list directly;

[8] In most situations the size of the AS candidate pool is large enough when going 5 or 6 hops away; however one risks not having enough candidates in the pool if going further away.

Node selection	Total succeeded		1	2	3	4	5
	#	ratio	probe	probes	probes	probes	probes
Random Probing	5,923,863	71.19%	47.35%	23.67%	13.86%	8.93%	6.19%
Strategy-**III** (> 4 hops)	6,960,678	83.65%	55.26%	22.37%	11.44%	6.67%	4.26%
Strategy-**III** (> 5 hops)	7,121,792	85.58%	57.67%	21.48%	10.75%	6.18%	3.91%
Strategy-**III** (> 6 hops)	6,590,845	79.20%	66.39%	20.06%	8.74%	3.20%	1.61%
Strategy-**III-A**	7,186,827	86.37%	60.07%	22.26%	9.58%	5.00%	3.09%

Table 3. The number of probes attempted upon success.

for one, the retrieved information kept by oneself could be stale and inaccurate. Moreover, an AS should avoid completely trusting the few origins in its list. Indeed, in the absence of a trust model that provides proof of identity, ownership, and authorization, we argue that it is beneficial to intelligently probe other ASes.

Advertisements of a prefix that has not been seen before should be given particular attention, especially when it is a sub-prefix of a known prefix. A previous study by Mahajan *et al.* [2] found that about 75% of all new prefix announcements result from misconfigurations. The longest matching prefix rule in BGP ensures that these routes would be selected over known ones–hence, this could represent a malicious attempt to "punch a hole" in another AS's address space. One option here is to initiate queries upon seeing such advertisements (e.g., for new prefixes or sub-prefixes). In the case of a sub-prefix advertisement, queries can also be sent to the currently known origins for its "parent" prefix.

For advertisements where the origins do not conflict with the known origin ASes, it is still useful to trigger queries with a reasonably low probability. Since change in ownership of prefixes does occur, and some prefixes are legitimately affiliated with multiple ASes [12], doing so helps maintain an up-to-date list of the origin ASes.

Lastly, another useful measure that can reduce the number of ASes that initiate queries in response to the same suspicious advertisement, is to delay propagation of such routes. Recently, Karlin *et al.* [13] also proposed the idea of delaying the propagation of suspicious routes to provide network operators time to respond before the problem escalates. For the most part, their results showed that slowing the acceptance of new routes is a safe and effective method, and doing so induces no noticeable loss of reachability for legitimate routes.

6 Related Work

The reliability and security of BGP is one of the most critical concerns of the operators of the current Internet. In this vein, Mahajan *et al.* [2] studied origin and export misconfiguration errors and found that configuration errors were pervasive, with 200–1200 prefixes experiencing errors every day. Besides misconfigurations, intentional attacks have also been another concern [3]. Murphy [14] highlighted that the security risks in BGP arise from the complete lack of message integrity and freshness, as well as from the fact that BGP neither verifies an AS's authority to advertise a prefix or validates the announced AS path. These last two issues are generally referred to as origin authentication and path verification, respectively.

While RFC 1930 [15] advocates that a prefix should belong to only one AS, this is not so in practice. For instance, Huston [16] observed a number of multi-origin prefixes, and Zhao *et al.* [12] subsequently showed that most multiple origin AS conflicts are short-lived (lasting a small number of days).

The need to address the security risks in BGP has spurred much activity in recent years. Authentication of the origin ASes of prefixes is one of the primary goals of most of the security designs proposed thus far [4, 5, 6, 7, 17]. In Secure BGP (S-BGP) [4], an address allocation public key infrastructure (PKI) was proposed to support origin authentication. Similarly, Secure Origin BGP (SoBGP) [5] introduced Authentication Certificates to verify the origin AS and the prefix, but mandated less authority. Recently, Aiello *et al.* [6] formalized the semantics of address advertisement and proofs of origin authentication, and proposed a model of IP address delegation graph.

Of late, a different body [18, 19, 20, 21] of security solutions for BGP have emerged based on the notion of anomaly detection. Listen-and-Whisper [20] and MOAS list [18] put extra information into BGP community attributes and monitor BGP messages exchanged between routers. Kruegel *et al.* [19] proposed a detection mechanism based on geographical information kept in a central registry. PHAS [21] requires the prefix owner to register with their system. The approach taken by PHAS is to maintain a current origin set for each registered prefix and notify the owner the origin change events through emails. Impediments of adopting these approaches include the requirement to change the current BGP protocol and a complex management infrastructure to guarantee the information in the registry is fresh, accurate, and complete.

Most closely related to the current work is the Internet Routing Validation (IRV) system [17] which proposed a decentralized query system. Each participating AS in IRV publicizes a server for answering route-relevant queries, and received information is vetted by querying unspecified IRV services. The IRV work was principally concerned with the development of the query system and protocols, and did not consider how it should be deployed. In that sense, this work can be seen to provide important suggestions for its use.

Finally, we note that should a certificate infrastructure that provides proof of identity, ownership, and authorization become available [22], as proposed in previous solutions [4, 5, 6], our detection mechanism could utilize it in an efficient way. For example, prefix attestations can be generated by the real origin and verified by relevant parties only when conflicts come up. As a result, the cryptographic operation overhead and impact to the current BGP protocol will be minimal.

7 Summary and Discussion

In this paper, we propose a mechanism for detecting false BGP origin advertisements. Our mechanism exploits the characteristics of the Internet in that its scale renders attempts to block the dissemination of information practically infeasible. We show how ASes can coordinate to inform the valid origin of a prefix p about the existence of other origins that are erroneously advertising p. We focus on evaluating the extent to which such a mechanism is viable in the real Internet, performing simulations based on real Internet topologies. Our results show that lightweight probing is highly effective, 98% of all invalid source announcements are detected with only 10% of the ASes deploying the proposed mechanism.

Finally, we explore the benefits that additional information can provide. When domains judiciously forward their information to other domains far away from the suspected malicious origin, the number of probes necessary to reach an authentic node is reduced dramatically. Our results indicate that by using this improvement we can achieve the same success ratio while reducing the number of probes by 50%.

8 Acknowledgments

This work is supported in part by NSF grant SCI-0334108.

References

[1] Y. Rekhter and T. Li, “A Border Gateway Protocol (BGP-4),” *RFC 1771*, 1995.

[2] R. Mahajan, D. Wetherall, and T. Anderson, “Understanding BGP misconfiguration,” in *Proc. of SIGCOMM*, pp. 3–16, 2002.

[3] O. Nordstrom and C. Dovrolis, “Beware of BGP attacks,” *ACM Sigcomm Computer Communications Review*, vol. 34, pp. 1–8, April 2004.

[4] S. Kent, C. Lynn, and K. Seo, “Secure Border Gateway Protocol (Secure-BGP),” *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 582–592, April 2000.

[5] J. Ng, “Extensions to BGP to support secure origin BGP (draft),” in *Network Working Group*, 2003.

[6] W. Aiello, J. Ioannidis, and P. McDaniel, “Origin authentication in Inter-domain routing,” in *Proc. of ACM Conference on Computer and Communications Security*, pp. 165–178, 2003.

[7] Y. Hu, A. Perrig, and M. Sirbu, “SPV: Secure path vector routing for securing BGP,” in *Proc. of SIGCOMM*, pp. 179–192, 2004.

[8] L. Gao, “On inferring autonomous system relationships in the Internet,” *IEEE/ACM Trans. on Networking*, vol. 9, no. 6, pp. 733–745, 2000.

[9] *University of Oregon Route View Project*. `http://routeviews.org`.

[10] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, “Characterizing the Internet hierarchy from multiple vantage points,” in *Proc. of INFOCOM*, pp. 618–627, 2002.

[11] S. Qiu, P. McDaniel, F. Monrose, and A. Rubin, “Characterizing address use structure and stability of origin advertisement in Inter-domain routing,” in *Proc. of IEEE Symposium on Computers and Communications*, pp. 489–496, 2006.

[12] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, “An analysis of BGP multiple origin AS (MOAS) conflicts,” in *Proc. of Internet Measurement Workshop*, pp. 31–35, 2001.

[13] J. Karlin, S. Forrest, and J. Rexford, “Pretty good BGP: Protecting BGP by cautiously selecting routes,” in *Proc. of IEEE International Conference on Network Protocols*, (to appear) 2006.

[14] S. Murphy, “BGP security vulnerabilities analysis,” in *Internet Research Task Force*, 2002.

[15] J. Hawkinson and T. Bates, “Guidelines for creation, selection, and registration of an autonomous system (AS),” *RFC 1930*, 1996.

[16] *Multi-origin Prefixes*. `http://bgp.potaroo.net/as6447/bgp-multi-orgas.txt`.

[17] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, “Working around BGP: An incremental approach to improving security and accuracy in Inter-domain routing,” in *Proc. of Network and Distributed System Security Symposium*, pp. 75–85, 2003.

[18] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, “Detection of invalid routing announcement in the Internet,” in *Proc. of Dependable Systems and Networks*, pp. 59–68, 2002.

[19] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, “Topology-based detection of anomalous BGP messages,” in *Proc. of International Symposium on Recent Advances in Intrusion Detection*, pp. 17–35, 2003.

[20] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, “Listen and Whisper: Security mechanisms for BGP,” in *Proc. of Network Systems Design and Implementation*, pp. 127–140, 2004.

[21] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, “PHAS: A prefix hijack alert system,” in *Proc. of USENIX Security*, pp. 153–166, 2006.

[22] K. Seo, C. Lynn, and S. Kent, “Public-Key Infrastructure for the Secure Border Gateway Protocol (S-BGP),” in *Proc. of DARPA Information Survivability Conf. and Exposition II*, pp. 239–253, June 2001.