Pretty Good BGP: Improving BGP by Cautiously Adopting Routes

Josh Karlin
University of New Mexico
karlinjf@cs.unm.edu

Stephanie Forrest
University of New Mexico
Santa Fe Institute
forrest@cs.unm.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

***Abstract*— The Internet's interdomain routing protocol, BGP, is vulnerable to a number of damaging attacks, which often arise from operator misconfiguration. Proposed solutions with strong guarantees require a public-key infrastructure, accurate routing registries, and changes to BGP. While experts debate whether such a large deployment is feasible, networks remain vulnerable to false information injected into BGP. However, BGP routers could avoid selecting and propagating these routes if they were cautious about adopting new reachability information. We describe a protocol-preserving enhancement to BGP, Pretty Good BGP (PGBGP), that slows the dissemination of bogus routes, providing network operators time to respond before problems escalate into a large-scale Internet attack. Simulation results show that realistic deployments of PGBGP could provide 99% of Autonomous Systems with 24 hours to investigate and repair bogus routes without affecting prefix reachability. We also show that without PGBGP, 40% of ASs cannot avoid selecting bogus routes; with PGBGP, this number drops to less than 1%. Finally, we show that PGBGP is incrementally deployable and offers significant security benefits to early adopters and their customers.**

I. Introduction

The Border Gateway Protocol (BGP) [1] has been the Internet's de-facto interdomain routing protocol for the last decade. During the last few years, numerous exploits of BGP have been discovered and documented [2]. Network operators can protect their networks against most of these vulnerabilities by adopting good administrative practices, such as authenticating peering connections with neighbors and giving routing-protocol traffic high priority in the data plane. However, the routers cannot easily verify the *contents* of the BGP messages—by default, routers believe what they hear. This vulnerability allows Autonomous Systems (ASs) to announce false (bogus) routes that lead data packets along the wrong paths. These bogus routes arise for one of two reasons:

Configuration mistakes: Simple BGP configuration errors can have serious global consequences. A classic example is a 1997 incident in which a small ISP originated the first class-C subnet of every IP prefix [3,4]. This created reachability problems for every network, and crashed routers around the world due to increased routing state. Although administrative practices have improved since then, even today's well-managed ASs, such as Verio (AS 2914), cannot completely protect themselves. On January 22, 2006, Con Edison (AS 25706) originated many prefixes it did not own, causing outages for several networks such as Panix (AS 2033) [5]. Verio accepted the false routes and passed them on to others due to stale information in its routing registry.

Malicious attacks: Adversaries can intentionally introduce bogus routes, typically for a small set of destinations. By configuring a router to originate someone else's prefixes, the adversary can start receiving packets destined to these addresses. The adversary can drop the packets (a denial-of-service attack) or snoop the traffic (compromising the user's privacy). Alternatively, the adversary can direct the traffic to a host under its control, to perform identity theft or send spam.

In this paper, we first focus our attention on protecting BGP from the effects of configuration errors, though we also discuss how our solution can defend against malicious attacks.

The cleanest way to prevent bogus routes would be to have a global registry of prefix ownership and connections between ASs, coupled with verification of the contents of BGP update messages. Although such solutions have been proposed [6–9], they have not been deployed in practice. The proposals require full, or at least large-scale deployment to be effective, and some solutions require a central routing authority and public-key infrastructure. This is currently impractical because ASs own their peering information, which they often have an incentive to conceal. Hence the major routing registries [10–12] are incomplete [13]. Even smaller-scale attempts, such as Verio's, to maintain accurate registries within a single organization have not been completely successful, as evidenced by the recent ConEd incident.

In response, the research community has proposed alternatives that apply anomaly-detection algorithms to identify bogus routes early in their propagation [14–17]. Although not provably secure, these methods can potentially improve BGP's security with a minimally invasive, incremental approach. Anomaly detection can be deployed incrementally because it does not require changing the BGP protocol. However, to be effective, the anomaly detector must be coupled with an effective response. Except for Whisper [15], which requires ubiquitous deployment to detect inconsistent routes, the BGP anomaly detectors do not actively stop the progression of attacks. Instead, they simply alert a human operator who often cannot respond quickly enough (e.g., to prevent identity theft or router overload). Bogus routes that are allowed to propagate, even for a short time, can wreak enormous havoc.

In this paper we present Pretty Good BGP, a system that *automatically* delays the use and propagation of suspicious routes in favor of familiar alternatives. In PGBGP, the routers identify suspicious routes by consulting a table of trusted routing information learned from the recent history of BGP update messages. Introducing delay gives the human operators, and automated systems, time to investigate the suspicious route; in some cases, the suspicious route may disappear on its own [13]. We evaluate PGBGP's effectiveness by studying its behavior on two of the most common BGP exploits—prefix hijacks and sub-prefix hijacks—using a sliding history window to construct a list of trusted (prefix, origin AS) pairs from the BGP update stream. Because our design does not require any protocol changes, PGBGP is incrementally deployable via software updates to the routers in participating ASs. Given the many impediments to deploying strong BGP security, it is important to evaluate how much of the problem can be addressed by weaker solutions such as anomaly detection. Such an evaluation would contribute to the ongoing debate about how to secure BGP. In this paper we explore both the strengths and the limitations of PGBGP, a very simple example anomaly detector.

In the remainder of the paper, we discuss the challenges of detecting bogus BGP routes (Section II) and present PGBGP (Section III). In Section IV, we describe a simulator for evaluating PGBGP. Section V reports simulation results that assess PGBGP's effectiveness under various deployment scenarios. Section VI discusses the implementation overhead and options for incremental deployment. Section VII describes an efficient method to quickly confirm true positives. Section VIII discusses PGBGP's response to a host of routing scenarios and how PGBGP might stand up to an intelligent adversary. Section IX reviews related work, and Section X presents our conclusions and directions for future research.

II. Challenges of Detecting BGP Attacks

In this section, we briefly review the BGP protocol and describe how routes are propagated. We then discuss some of its vulnerabilities.

A. Border Gateway Protocol (BGP)

Internet routing operates at the level of IP address blocks, or *prefixes*. Regional Internet Registries (RIRs), such as ARIN, RIPE, and APNIC, allocate IP prefixes to institutions such as Internet Service Providers. These institutions may, in turn, subdivide the address blocks and delegate these smaller blocks to other ASs, such as their customers. Ideally, the RIRs would be notified when changes occur, such as an AS delegating portions of its address space to other institutions, two institutions combining their address space after a merger or acquisition, or an institution splitting its address space after a company break-up. However, the registries are notoriously out-of-date and incomplete. Ultimately, BGP update messages and the BGP routing tables themselves are the best indicator of the active prefixes and the ASs responsible for them. BGP tables today contain around 170,000 active prefixes, with prefixes appearing and disappearing over time.

ASs exchange information about how to reach destination prefixes using the Border Gateway Protocol (BGP). A router learns how to reach external destination prefixes via BGP sessions with routers in neighboring ASs. BGP has two kinds of update messages—announcements and withdrawals. Upon receiving an announcement for a destination prefix, the router overwrites the old route (if any) from the neighbor with the new information. Announcements contain information such as the destination prefix, the announcer's IP address, and the AS path the route will take. As the route announcement propagates, each AS adds its own unique AS number to the AS path. The router responds to a withdrawal message by deleting the previously announced route from its routing table and propagating the withdrawal to its neighbors. BGP routing changes can occur for many reasons, such as equipment failures, software crashes, policy changes, or malicious attacks. Inferring the cause directly from the BGP update messages is a fundamentally difficult, if not impossible, problem.

A router with multiple neighbors would likely learn multiple routes for each prefix. A single "best" route is chosen by applying the BGP *decision process*. The decision process is a non-standard sequence of about a dozen rules that compare one route to another [1]. Generally, a router prefers routes that conform to the policies of the local network operator. Next, the router prefers routes with the lowest AS path length. If multiple equally good routes remain, the router can apply additional rules, ultimately resolving ties arbitrarily to ensure a single answer. Because the decision process does not consider traffic load or performance metrics, the selected route is not necessarily optimal from a performance point of view.

In practice, routes are often selected and propagated according to local routing policies, which are based on the business relationships with neighboring ASs [18, 19]. The most common relationships are customer-provider and peer-peer. In a customer-provider relationship, the provider ensures that its customer can communicate with the rest of the Internet by exporting its best route for each prefix, and by exporting the customer's prefixes to other neighboring ASs. In contrast, the customer does not propagate routes learned from one provider to another as it pays for transit to its providers. In a peer-peer relationship, two ASs connect solely to transfer traffic between their respective customers. An AS announces only the routes learned from its customers to its peers. These business relationships drive local preferences, which in turn influence the decision process. Typically, an AS prefers customer-learned routes over peer-learned routes, and peer-learned routes over provider-learned routes.

B. BGP Vulnerabilities

BGP has three major vulnerabilities. The first, a *prefix hijack*, occurs when an AS announces itself as the originator of a prefix it does not own. Some ASs will reroute to the hijacker instead of the legitimate host, making the prefix unreachable for themselves and their customers. The second, a *sub-prefix hijack*, occurs when an announced prefix is wholly contained within another announced prefix owned by another AS. It is more dangerous than a prefix hijack and more difficult to stop because more specific routes are preferred at traffic forwarding time. Finally, there is a *man-in-the-middle attack*. Unlike prefix hijacks and sub-prefix hijacks, man-in-the-middle attacks are always initiated by a malicious agent. Man-in-the-middle attacks occur when an agent does not claim to originate another AS's prefix but instead announces itself as part of an invalid path to the origin in order to gain access to traffic it should not receive. Man-in-the-middle attacks are the least common[1] of the three forms of attack, so in this paper we concentrate on both classes of hijacking attacks.

1) Prefix Hijacks: Prefix hijacking is surprisingly difficult to prevent. Ideally, every AS would apply filters to the routes received from neighboring ASs and discard BGP routes for unexpected prefixes. However, cases such as the Panix attack show that even vigilant ASs cannot maintain up-to-date filters to their neighbors, let alone for routes that originate several AS hops away. Ultimately, even security-conscious operators cannot adequately protect their ASs today.

Prefix hijacking can also be difficult to detect. Ideally, a prefix would have a single origin AS for its entire lifetime, causing a route announcement with a different origin AS to be clear indication of attack. However, prefixes may change ownership. For example, some companies and universities prefer to have their provider announce prefixes into BGP on their behalf. If the institution switches providers, a new AS would then start announcing the prefix. In addition, a small fraction of prefixes have more than one legitimate originating AS [20]. For example, an institution might have multiple providers that each announce the prefix. Thus, not all new origins for a prefix necessarily imply a prefix-hijack attempt.

2) Sub-prefix Hijacks: In a conventional prefix-hijacking attack, some ASs direct traffic toward the adversary while others continue to forward packets to the legitimate destination as the hijacked route is potentially one option among many. However, a small modification makes the attack more dangerous. When a data packet arrives on an incoming link, the router looks in its forwarding table for the entry with the longest matching prefix. By announcing more specific prefixes (*sub-prefixes*), the adversary can trick nearly every AS into using the bogus route. For example, the adversary could announce BGP routes for two sub-prefixes, each covering half of the address space of the original prefix. Routers throughout the Internet would select a best BGP route for each prefix—the original prefix and the two sub-prefixes. Yet, these routers would forward data packets based on the longest matching prefix—that is, the sub-prefix announced by the adversary.

Route filtering could help prevent such attacks by discarding BGP announcements for small address blocks. However, the network operators in one AS cannot easily determine what prefix lengths are reasonable to expect for each part of the IP address space. Operators typically take a conservative approach by allowing announcements for prefixes corresponding to 256 addresses or more (i.e., a prefix with a mask length of 24 bits or less), rather than run the risk of misrouting legitimate traffic. Even when detected, sub-prefix hijacks are hard to avoid. For example, suppose a network operator detects a sub-prefix hijack and configures a route filter to discard the offending route. Although that AS's routers would then forward data packets based on the original prefix, other ASs in the path to the legitimate destination might still be forwarding packets based on the bogus sub-prefix. These ASs would essentially *deflect* the packets to the adversary.

III. Pretty Good BGP (PGBGP)

The basic idea behind PGBGP is simple, namely, that unfamiliar routes should be treated cautiously when forwarding data traffic. This conservative approach to new route information takes advantage of natural redundancy in the network (more than one route for most data packets to reach their destination), and it mitigates the effect of temporary problems caused by configuration errors. Adopting potentially bogus routes slowly also creates time for secondary processes to check their validity. In the following, we discuss how PGBGP determines if a new route should be treated suspiciously (Identifying Anomalous Routes) and how PGBGP routes around bogus routes (Avoiding Suspicious Routes).

A. Identifying Anomalous Routes

PGBGP uses a window of historical data to determine whether or not a route is trusted. Thus, we say that the window of routes recently advertised or in the router's tables constitutes our definition of *normal*. Here we give some details about what information is used to construct normal, how it is built, and how long the data are trusted.

The most disruptive routes are those that can mislead routers into sending data to the wrong destination. PGBGP is therefore concerned with the originating AS of each route update an AS receives. Route origins can be obtained from update messages by selecting the last AS from the AS Path list. [2] PGBGP also uses the following information: the time that each update is received, the prefix associated with the update, and a snapshot of each edge router's RIB (table of known routes) in the AS.

A router's RIB and history of updates are used to create a history of known origins for each prefix. This history is what PGBGP uses to define normal behavior. On initialization, there is no concept of normal, and therefore all incoming updates are accepted. This process continues for h days (the *history period*). After this initial training phase, new routes that would alter the state of normal behavior are quarantined if possible. The quarantine lasts for s days (the *suspicious period*), and after that time the update is accepted by PGBGP if it still exists in the routing table. This prevents short-term anomalous behavior from corrupting the definition of normal. Finally, stale data should be eliminated from the history. PGBGP

[1]A malicious agent must gain access to the router in order to perform a man-in-the-middle attack.

[2]If the route is aggregated with an AS set, which is rare, the originating AS is considered to be the last AS before the AS set.

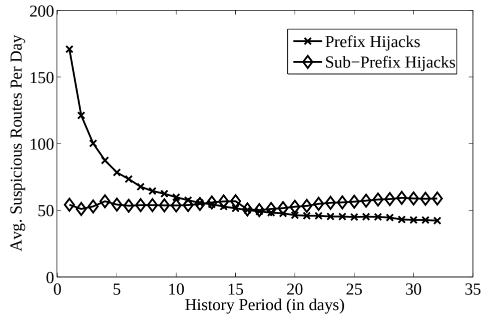

Fig. 1. Average number of announcements (per day) classified as suspicious using a suspicious period of 1 day and a variety of history periods (h).

removes known origins for a prefix if it has not appeared in the router's RIB in the last h, days. Likewise, if a prefix has not appeared in the router's RIB in the last h days, the entire prefix is removed from the history.

Incoming route updates are compared against the history of origins to determine whether or not they are suspicious. With this approach, hijack attempts are easy to detect, because they always originate a prefix at a new origin AS. PGBGP scans incoming updates for prefixes that have been seen recently (within the history period) but were not originated at the advertised location. Such route updates are labeled suspicious unless one of the trusted (recently seen) origins of the prefix are on the route's AS path. If the route is not a potential prefix hijack, it is either normal or a sub-prefix hijack attempt. Sub-prefix hijacks must announce a *new* prefix that is contained within another, recently seen, prefix in order to disrupt routing decisions. The prefix of a route update can be compared to recently seen prefixes to determine if it is a sub-prefix of a known prefix. If it is, then PGBGP labels it suspicious if the AS path does not traverse one of the larger prefix's origins.

The *suspicious period* s and *history period* h are PGBGP's only parameters. They correspond to the time an anomalous route is avoided before being accepted (s) and the time that an origin is viewed as "recently seen" (h). Parameter s should be long enough for network operators to detect and resolve problems before they spread, but no longer than necessary. If s is too long, false positives will be slow to self-correct. A previous study of BGP misconfiguration showed that roughly 45% of new origins and prefixes exist for less than 24 hours [13]. These are temporary routes such as route leaks and hijack attempts. Because 24 hours is also a reasonable length of time for an operator to analyze and fix a routing problem, we use this value for s.

Parameter h cannot be too short, or many valid origin ASs will be treated as suspicious following a brief outage. On the other hand, h should not be longer than necessary for two reasons. First, a long history period might allow a repeated prefix-hijack attack to become trusted. This would occur if an undetected malicious origin AS remained in the history buffer after the first attack. And, h determines the initial training time for a router coming online (unless it is bootstrapped with recent history information).

To determine a reasonable value for h, we ran the PGBGP algorithm on RouteViews BGP update data from Equinix for the months of November through January (inclusive) of 2005,2006 with $s = 24\ \texttt{hours}$. Only one of Equnix's many streams, that of AS 2914, was analyzed for this experiment. The average number of incoming announcements (per day) that are labeled anomalous are displayed in Figure 1 for each evaluated history period (for both suspicious new origins and sub-prefixes). The figure shows that as h increases the number of suspicious routes decreases on average for suspected prefix hijacks and gently increases for suspected sub-prefix hijacks. The reason that the average number of suspicious sub-prefix routes increases is that sub-prefixes are only considered suspicious if any recently seen prefix contains it. The larger the value of h, the more likely a prefix will have been seen within that period that contains it. For prefix hijacks, the figure shows a large initial drop in the average number of suspicious routes. This suggests that some prefixes have multiple origins that were not seen in the update stream for a few days at a time. The figure also shows marginal reductions in the rate of suspicious routes after ten days and therefore we have (somewhat arbitrarily) chosen $h = 10$.

B. Avoiding Bogus Routes

A PGBGP-enabled router would avoid selecting anomalous routes whenever possible. If the router had alternative routes for the prefix, the router would select the best of the trusted routes. False positives, while possible, cause the router to select a potentially less desirable route (temporarily). If no alternative route existed, the router would select the suspicious route. This behavior is accomplished by giving suspicious routes the lowest possible preference during the suspicious period. In this way a suspicious route will only be selected when no alternatives exist.

Preventing a sub-prefix hijack is more complicated because the router does not have any normal routes available for the sub-prefix. PGBGP approaches this problem by forwarding packets as before, using the BGP route for the larger address block (super-prefix). The suspicious routes are not immediately entered into the routing table but instead quarantined until the suspicious period has passed. Extra consideration must be taken in selecting the route for the larger address block now that a sub-prefix has been announced. A downstream AS that chose a malicious route would *deflect* the data packets along the wrong path anyway.Hence, when possible, the super-prefix route that is selected should lead to a neighbor that has not announced the suspicious sub-prefix.

An interesting question is how the announcement of a new prefix that is not contained in a larger address block should be handled. In this case, the new announcement provides a route to an address block that was either previously unreachable or is specified more specifically by prefixes in the table. If the announced addresses were previously unreachable then the route cannot be hijacking traffic destined to another, legitimate AS. PGBGP accepts the new announcement and installs the new prefix in the forwarding table. A super-prefix announcement is not a hijack either. Super-prefixes will not be preferred over sub-prefixes at packet forwarding time and cannot hijack traffic. Therefore, super-prefixes are accepted by PGBGP as well.

We have shown that it is possible to avoid suspicious routes. However, any modification to the decision process needs to consider the possible effects on BGP convergence. Although BGP is not guaranteed to converge for all combinations of routing policies [21], ASs typically select and export routes based on their business relationships. If every AS prefers customer-learned routes, BGP convergence can be provably guaranteed [18]. As long as local preference remains the first step in the decision process, the guidelines in [18] are still being followed and convergence is assured. However, ranking all anomalous routes lower than other routes seems to violate these guidelines. For example, an AS would prefer a non-suspicious route learned from a peer over an suspicious route learned from a customer. Fortunately, this does not cause a problem. Removing the suspicious route from consideration is conceptually the same as having the customer decide not to announce the route to the AS in the first place. The convergence guarantee in [18] holds when ASs apply more conservative export policies than their business relationships normally suggest.

IV. The PGBGP Simulator

We have developed a high-level BGP simulator for evaluating route selection and propagation on large topologies. The software, available for download under the GPL license [22], simulates BGP and PGBGP routing decisions on an AS topology with routing policies based on the business relationships. In this section, we describe the AS-level topology, the decision process and route propagation, and how the simulator is configured for the experiments in Section V.

A. AS Topology and Relationships

Large ASs are often spread over vast geographical areas and have many BGP-speaking routers. Because we are concerned only with AS-level behavior, each AS's network is represented as a single node in the graph. In spite of this simplification, determining the AS-level topology of the Internet is a difficult problem. Much of the topology can be inferred from the BGP routing announcements themselves. For example, suppose that an AS A announces the paths (A,C,D,E) and (A,C,D,T,Y) for two different prefixes. These paths imply the existence of several edges in the AS-level topology, namely (A,C), (C,D), (D,E), (D,T), and (T,Y). The AS paths also provide a glimpse into the business relationships between ASs. For example, the path (A,C,D,E) implies that AS A is permitted to transit traffic through AS C to AS D. As such, we can infer that AS A and AS D cannot both be providers or peers of AS C. Each path implies a set of constraints on the relationships between ASs. By combining these constraints across a large number of paths, inference algorithms can classify the relationship between each pair of adjacent ASs as customer-provider or peer-peer [23].

Based on the topology and AS relationships, we identified a set of ASs that are likely at the top of the AS "hierarchy," the core ASs. These ASs connect to each other via peer-peer links and provide transit service to large customer bases. We label an AS as core if it has peer-peer relationships with fifteen or more neighbors. For our experiments, we used the AS topology and business relationships described in [24], which were inferred from BGP data collected primarily from RouteViews [25]. The topology has 18,943 ASs with an average of four AS-AS links each. The work in [24] introduced the concept of a sibling relationship, which we approximate as a peer-peer relationship. The network has 62 core ASs according to our definition. Although inferring AS topology and business relationships is by no means perfect, we believe that the inferred graph is representative of the connectivity and hierarchical structure present in today's Internet.

B. Route Selection and Propagation

The simulator models how each AS selects and propagates a best route for a prefix. Following conventional business practices, an AS exports its best route to a peer or provider only if the route was learned from a customer; in contrast, an AS always exports its best route to its customers. For each AS, the simulator models a decision process with three main steps. First, the routes with highest local preference are selected; highest preference is given to routes announced by customers, then peers, and finally providers. Next, routes with the shortest AS paths are chosen. If multiple routes remain, the route learned from the neighbor with the lowest AS number is arbitrarily chosen as the tie-breaker. The simulator does not model other steps in the decision process, which relate to details of intra-AS topology and routing. When PGBGP is enabled, suspicious routes are ranked lower than trusted routes before the local-preference step.

The simulator propagates routes by visiting the originator's neighbors in breadth-first order. Upon reception of the new route, the neighbors run the decision process and propagate the route to their neighbors if it is selected as the best route. Cycles are avoided by ignoring routes that contain the receiving AS in the path. The propagation process continues until all of the ASs' best routes have stabilized. Every experiment terminated successfully, consistent with the observation in Section III-B that the routing system should converge.

Our experiments determine which ASs would select a bogus route, and how PGBGP limits and delays the propagation of the route across the AS topology. Studying the propagation of the bogus route does not require any simulation of network dynamics such as topology changes, route-flap damping, or configuration changes. Instead, the simulator repeats the computation of the ASs' routing decisions once every s steps. First, the simulator computes the routing decisions for each AS with only the legitimate AS originating the prefix. Then, the simulator introduces a malicious AS that also originates the prefix, and recomputes the routing decisions. Because some ASs may suppress the bogus route for s steps, we then evaluate what happens when these ASs stop suppressing the route. The

Variable	Values
History period (h)	number of days (3)
Suspicious period (s)	number of days (1)
Deployment type	random or (core + random)
Attack type	prefix or sub-prefix hijack
Runs	positive integer (500)

TABLE I

SIMULATOR PARAMETERS (AND DEFAULT VALUES)

process repeats until no ASs change their decisions. Since the AS-level diameter of the Internet is small, no experiment required more than six steps to complete.

C. Experimental Configuration

The simulator has several configurable parameters, as summarized in Table I. These include h and s, which are set to 3 days and 1 day, respectively. There are also two deployment options. A *random* deployment enables PGBGP on a random set of nodes, modeling a situation where all ASs are equally likely to deploy the enhanced protocol. The *core + random* deployment enables PGBGP on the 62 core nodes (i.e., the ASs with fifteen or more peers) and a random chosen subset of the remaining nodes, modeling a likely scenario in which a small number of large service providers deploy the enhanced protocol, along with a random set of other ASs.

We can simulate both prefix and sub-prefix hijacks. In the first case, a randomly chosen AS originates the prefix and, on the next simulated day, a randomly chosen attacking AS originates the same prefix. Sub-prefix hijacks are simulated identically except that the attacking AS announces a sub-prefix of the legitimate AS's prefix. Each "Run" simulates a single attack instance for the given parameter settings. Each set of runs is evaluated with different fractions of ASs deploying PGBGP, ranging from 0 to 100% in increments of 10%. For each deployment scenario, attack type, and fraction of AS deployment, we simulated 500 attacks.

For all the experiments, we randomly selected the origin AS of the bogus route. This might be a reasonable assumption for prefix hijacks caused by unintentional configuration mistakes. However, some intentional, malicious attacks would be difficult for PGBGP, or any other solution, to stop. For example, suppose the adversary controls an AS that lies on all paths to the legitimate origin AS—i.e., if the adversary is the provider for the legitimate origin AS. (Admittedly, such an attack seems unlikely because a provider would not have an incentive to disrupt reachability to its own customers, but this situation might happen due to an insider attack.) In future work, we plan to evaluate the effects of targeted attacks such as these, in which the adversary chooses the most damaging possible attack location.

V. LARGE-SCALE EVALUATION

This section reports simulation results on PGBGP's effectiveness. First, we show that PGBGP can protect most ASs from prefix hijack attacks, even when only a small fraction of ASs deploy the enhanced protocol. Then, we show that defending against sub-prefix hijacks requires a larger-scale deployment. Next, we illustrate that PGBGP's automated response helps ensure ASs learn a viable alternative to the bogus route. Last, we demonstrate that false positives will self-correct over time; all legitimate routes eventually propagate throughout the network.

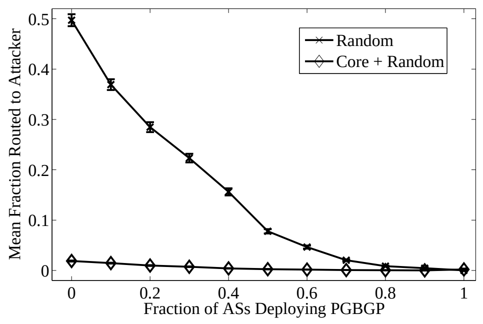

Fig. 2. Both Deployments, Prefix Hijack, Day One

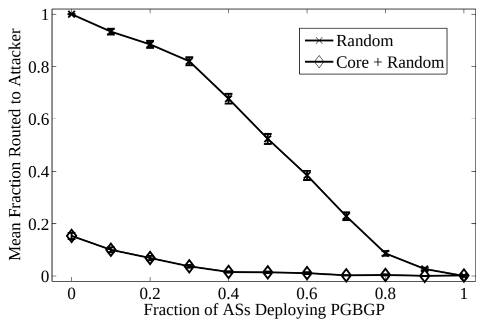

Fig. 3. Both Deployments, Sub-Prefix Hijack, Day One

A. Stopping Prefix Hijacks

First, we study PGBGP's ability to detect and avoid prefix-hijack attempts immediately after the adversary originates the route announcement. Figure 2 plots the average fraction of ASs that select a route to the bogus origin AS, as a function of the fraction of ASs that have deployed PGBGP. The error bars represent the standard error of the mean. The top curve plots the results for a random deployment of PGBGP. With zero deployment, which represents BGP today, half of the ASs select a route to the bogus AS, on average. With a complete deployment of PGBGP, more than 99% of the ASs are protected during the initial outbreak of an attack. (Even with complete deployment, a few ASs may learn only the bogus route. For example, the adversary's single-homed customers would learn only the bogus route. In the extreme case where the adversary is the sole provider for the legitimate origin AS, no other ASs could learn the legitimate route.) Although incremental deployment of PGBGP offers incremental gains, achieving substantial gains still requires a fairly large number of randomly chosen ASs to enable PGBGP.

An AS that deploys PGBGP provides protection for all neighbors that learn the AS's best route. As such, deploying PGBGP on the small number of core ASs offers substantial benefits, as shown in the bottom curve in Figure 2. Running PGBGP just on these 62 ASs (and 0% of the remaining ASs) ensures that, on average, less than 2.5% of the ASs in the Internet select a route to the bogus origin AS. Comparing with the top curve shows that a completely random deployment would require *three-fourths* of the ASs to run PGBGP to offer the same degree of protection. Along with the base deployment on the 62 core ASs, running PGBGP on a randomly chosen set of additional ASs offers even larger gains. The results for the "core+random" scenario are very important, because convincing a small number of large service providers to run PGBGP is much easier than convincing ten thousand smaller ASs to do so. Large service providers upgrade their router software much more frequently and are more aware of the latest trends and best common practices.

B. Stopping Sub-Prefix Hijacks

The results for sub-prefix hijacks are similar, although a wider PGBGP deployment is required to achieve the same gains, as shown in Figure 3. With zero deployment of PGBGP, which represents BGP today, every AS directs traffic to the bogus AS, because the routers forward packets based on the longest prefix match. The incremental benefits of deploying PGBGP on a random set of ASs is not as significant for sub-prefix attacks until around 40% of ASs run the enhanced protocol, compared with the top curve in Figure 2. The incremental gains are smaller because ASs along the path to the legitimate origin AS may deflect the data packet toward the adversary. Successfully avoiding the adversary sometimes depends on these intermediate ASs running PGBGP as well.

Fortunately, the "core+random" deployment fares much better because the large service providers do not choose the bogus routes, and thus do not advertise any route for the sub-prefix to their many customers. The bottom curve in Figure 3 shows that deploying PGBGP on the 62 core ASs, along with 20% of the remaining ASs, protects 94% of ASs from the sub-prefix attack. In fact, the results are nearly as good as the "core+random" results for the prefix-hijack case in Figure 2. As an added benefit, ASs that never learn the sub-prefix (e.g., because their providers classified it as suspicious) do not waste space on the routers for storing the routes. This helps protect smaller customer ASs with low-end routers from the excessive overhead introduced by short-lived route leaks caused by configuration errors.

C. Importance of a Collective Response

In addition to avoiding bogus routes, a PGBGP-enabled AS plays an important role in ensuring that other ASs learn viable alternative routes. As a point of comparison, suppose that no ASs run PGBGP, but that an AS has a separate anomaly-detection system that determines that a particular route is malicious. When a bogus route is detected, would the AS have a legitimate alternative? When all ASs are running conventional BGP, half of the ASs select a route to the bogus AS, as shown earlier in the top curve of Figure 2. Do most of these ASs have an alternate route that uses the legitimate

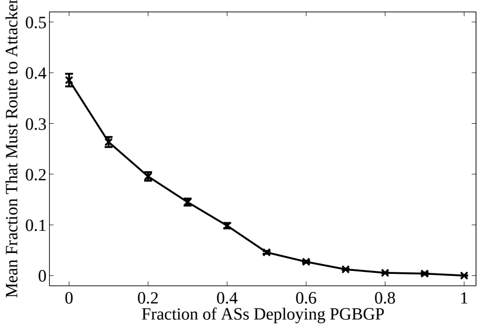

Fig. 4. Random Deployment, Prefix Hijack, Cannot Avoid

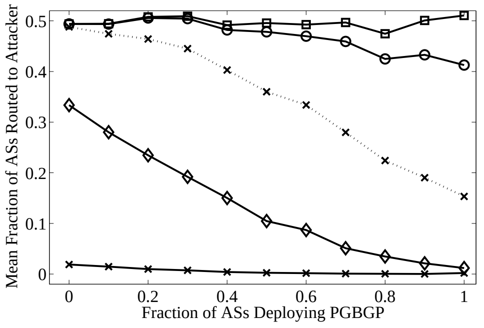

Fig. 5. Core + Random Deployment, Prefix Hijack, 5 Days

AS, should they independently realize that the other AS is advertising a bogus route?

The general answer is "no," as shown in Figure 4. For this graph, we compute the fraction of ASs that learn no routes to the legitimate origin AS. When no ASs deploy PGBGP, nearly 40% of the ASs fail to learn a route that could avoid the bogus route's origin AS; that is, nearly four-fifths of the ASs that pick the malicious route do so because they have no alternative. Even if these ASs had a separate anomaly-detection system, they would be unable to protect themselves retroactively from the prefix-hijack attack. As more ASs deploy PGBGP, many of these ASs choose legitimate routes and, in turn, help ensure more ASs have a viable alternative.

D. Attack Propagation

For the simulation parameters, operators have a 24-hour period to detect and resolve attacks before the routers automatically accept the anomalous routes as normal. If a bogus route has not been diagnosed and blocked, some of these ASs would select the route and propagate it to additional ASs, enabling the second wave of an attack. If the route is legitimate (i.e., a false positive), a broader set of ASs will start learning about the valid route. By analyzing how these routes propagate, we can understand both how quickly an undetected bogus route spreads and how quickly a false positive corrects itself.

Figures 5 and 6 show how the routes propagate under a "core+random" deployment for both prefix and sub-prefix hijacks, respectively. Each graph has five curves, corresponding to five days. The bottom curves (with diamonds) represents

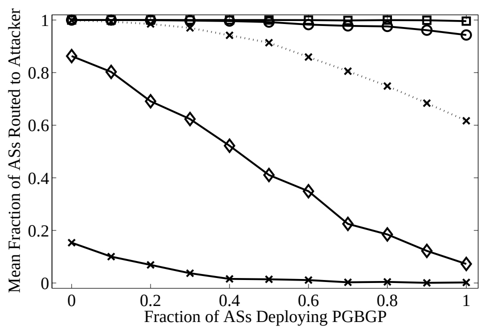

Fig. 6. Core + Random Deployment, Sub-Prefix Hijack, 5 Days

the first day, corresponding to the bottom curves in Figures 2 and 3, respectively. On each subsequent day, the protective effect decreases, as each day's curve is higher than the one before. With a ubiquitous deployment of PGBGP (the most effective protection), five days is sufficient for a nearly complete propagation of the previously suspicious route, because most pairs of ASs are connected by paths with five hops or less. By then, half of ASs would select the prefix and nearly 100% would use the sub-prefix, as with BGP today.

These graphs illustrate the trade-off between protecting against bogus routes (real attacks) and self-correcting for false positives (legitimate new routes). Ultimately, the trade-off can be managed by manipulating the duration of the suspicious period. In addition, once a secondary response system concludes that a suspicious route is valid, the routers in an AS could be configured to start treating the route as a legitimate immediately, rather than wait for the timeout.

VI. Implementation and Deployment

PGBGP does not require any changes to the BGP protocol, allowing one AS to deploy the enhanced protocol when other ASs have not. An implementation of PGBGP has two main components: constructing the set of recently-seen (prefix, origin AS) pairs and applying a modified decision process to select the best route for each destination prefix. We see three main options for realizing these two functions:

Implementing both functions on the routers: Implementing PGBGP on the routers requires extending how the routing software processes incoming BGP update messages. Upon receiving a BGP announcement, the router would need to compare the origin AS with the recently-seen ASs for this destination prefix to determine if the route is suspicious. Suspicious prefixes would be assigned a lower local-preference value, and suspicious sub-prefixes would be suppressed, to ensure that the router uses trusted routes where possible. In addition, the router would need to update the set of trusted (prefix, origin AS) pairs as new update messages arrive.

Separating the functions between routers and servers: The edge routers can be configured to forward all externally-learned BGP update messages to a server. The server can analyze the data to construct the set of trusted (prefix, origin AS) pairs, and periodically upload the information to the routers. When a new BGP update message arrives, the router can consult the set of trusted (prefix, origin AS) pairs to classify the route and apply the PGBGP decision process. This approach allows the set of (prefix, origin AS) pairs to reflect the BGP routes seen by all routers in the network, and reduces the load on the routers. The router can continue to process BGP update messages and select routes in real time, without waiting for the latest upload from the server.

Implementing both functions on separate servers: The server could take complete responsibility for implementing the PGBGP algorithm. As in the previous solution, the edge routers are configured to forward all externally-learned routes to the server. In addition to constructing the set of trusted (prefix, origin AS) pairs, the server applies the PGBGP decision process and sends each router a single best route for each prefix. This would be possible today by implementing PGBGP on the Routing Control Platform (RCP) described in [26, 27]. This approach obviates the need for *any* changes to routers, but places a larger burden on the server to be fast and reliable.

All three approaches are viable in practice. In addition, the overhead for analyzing the BGP updates is not significant. We implemented the analysis algorithm to generate the results in Section III. Our prototype analyzed three months worth of BGP update data (from May to July of 2005) from AS 2914's reflector stream to Equinix in 46 minutes on a 1.8 GHz Opteron with a maximum memory usage of 100 MB for a suspicious period of 1 day and history period of 3 days. Conducting the same analysis for all 40 peers of the RouteViews2 view requires 400 MB memory and 18 hours. This should be fast enough to handle any AS's update streams in real time. This is consistent with the previous work on the RCP [27] that shows that a high-end PC has sufficient CPU and memory resources to process all BGP update messages from the edge routers of a large ISP in real time.

VII. The Internet Alert Registry

Once a suspicious route has been identified by PGBGP, it can be difficult to determine if it is a true or false positive. It is impractical to expect network operators to verify all suspicious routes manually, because of volume and ambiguity.

The operators in the best position to determine the legitimacy of a particular route are often the ones who are most interested in those routes. For example, the legitimate origin AS of a hijack is best aware of the legitimate origins of the hijacked prefix. Also, the operator of the AS from which the attack originates knows which prefixes it should announce and can most quickly repair a misconfiguration. If these two operators are informed of each suspicious route that PGBGP finds, the operating overhead could be minimized and routes could be verified by the most knowledgeable parties.

Here we describe the Internet Alert Registry, a prototype service for notifying origin ASs of bogus routes. IAR is an opt-in service in which operators submit their e-mail address and the AS numbers that they wish to monitor. For instance, an operator of AT&T might register only AS 7018. Thereafter, the operator will be notified by e-mail of any attack in which

AS 7018 is either the victim or instigator. When an alert is received, the operator can attempt to resolve the situation with the other party.

The service does not need to be adopted by all operators in order to succeed. ASs that do not receive alerts will not receive the benefits of the IAR, while those that do can take an active role in protecting themselves.

We implemented a proof-of-concept IAR [28] in the form of a website. It displays all suspicious routes found within the last 24 hours and provides search functionality for archival purposes. It currently monitors all BGP update streams from RIPE RRCs $\{0, 1, 2, 3, 5, 11, 12, 13\}$ [3] for bogus routes by using the pseudo-PGBGP router developed for use in Section III-A. E-mail registration is also available on the website in the fashion previously described. Preliminary Results show that for the inclusive period from May 1st - May 10th of 2006, the mean number of alerts for Tier-1 ASs (7018, 3356, 701, 1239) per day was only 1.1 with a standard deviation of 0.5.

VIII. Limitations of PGBGP

PGBGP is an extremely simple approach to a complex problem. In this section, we discuss some of the complexities, how PGBGP addresses them, and some cases where extending PGBGP may be warranted.

A. False Positives

A prefix hijack is identified by the announcement of multiple simultaneous origins for a single prefix. There are two cases when this can legitimately occur:

Provider Change: PGBGP will accept the new provider's route by default once the old provider withdraws it.

Previously Unseen Auxiliary Provider: If both the old and the new routes are advertised simultaneously, the old one will be used until the suspicious period has elapsed. In the event of a backup route due to failure of the primary provider, the backup route will be used by default. Note: If a trusted origin for the prefix is on the AS path to the new origin, PGBGP will not treat the route as potentially bogus.

A sub-prefix hijack is identified when a prefix that is wholly contained within a prefix recently seen and owned by another AS is announced. In some scenarios, an AS could legitimately announce sub-prefixes that it owns. PGBGP will not interfere in this scenario because the known super-prefix origin is on the AS Path to the sub-prefix. However, PGBGP will treat the following legitimate situations as suspicious:

Provider Change: In rare circumstances an ISP will transfer a block of its old provider's address space to a new provider. In order to change providers rapidly under PGBGP, the following protocol could be followed: (1) The old provider announces the sub-prefix; (2) Routes with the sub-prefix will therefore be treated as a prefix hijack (instead of a sub-prefix hijack), because of (1); (3) The new provider announces the prefix, and the old provider withdraws it; (4) The new provider's route is then used by default.

Previously Unseen Auxiliary Provider: If an AS legitimately announces IP space allocated by one provider to another, PGBGP will delay its propagation until the suspicious period has passed. During the delay period traffic would continue to flow based on the larger address block. If the route is found to be bogus the delay would have prevented an attack. Otherwise, after the suspicious period has passed, the saved routes would be entered into the routing table and used normally. In order to ensure that a backup of this type is not delayed, operators could employ the common practice of regularly announcing the backup route with a prepended AS path.

B. An Intelligent Adversary

We are aware of three ways in which PGBGP's security could be compromised. First, an adversary could force a prefix's routes to be withdrawn via a denial-of-service attack and subsequent announcement of its own route for the same prefix. In this scenario, PGBGP would select the illegitimate route because no alternative route for the prefix would exist. This is no different than what happens with the current BGP. This case is addressed by sBGP, and PGBGP could potentially address it as well through the IAR mechanism. A second vulnerability is created by PGBGP's delay mechanism. If a bogus route were to pass through the delay phase unnoticed, it would eventually propagate as occurs with BGP today. This form of attack is well addressed by the IAR and would succeed only if operators neglected their IAR notifications. Finally, a sophisticated attacker could compromise a router and announce a very short fake route that passes through her AS but ends at a legitimate origin. This is known a man-in-the-middle attack. Although this case is not covered by our current PGBGP design, we could use PGBGP principles to cover this kind of attack as well, for example, by treating routes with anomalous edges as suspicious. This is an important avenue for future work, even though man-in-the-middle attacks are still uncommon.

IX. Related Work

In addition to the centralized approaches discussed earlier, there are several other proposals for improving BGP security.

Whisper [15] and MOAS lists [14] (lists of legitimate origins for a prefix), detect suspicious routes by monitoring the BGP messages exchanged between routers. Both proposals use the BGP community attribute to convey extra information along with the update. Unfortunately, in ASs that have not deployed the protocol enhancements, the routers are likely to strip the community tag.

Kruegel *et al.* [16] proposes a solution that detects prefix-hijack attempts and false updates based on geographical information obtained from a central registry, such as the Whois database. Although Whois data are often incomplete and out-of-date, they argue that the geographic locations of ASs do not change frequently. Although their prefix-hijack detector bears

[3] In future iterations, registered users will also be able to forward alerts that their own servers have discovered to the IAR. This will become necessary as PGBGP's deployment increases to ensure maximum visibility of bogus routes.

some similarity to PGBGP's, it relies on precomputed prefix-ownership lists and does not detect sub-prefix hijacks. Their detector passively responds to attacks by alerting the operator to the problem, while still allowing the attack to propagate. In contrast, PGBGP has an automated response that prevents the dissemination of bogus routes.

Wang *et al.* [29] developed a BGP anomaly detector for use with top-level domain server routes. In order to prevent TLD route hijacks, they suggest filtering out all but the most durable (and verified) routes. This is feasible for two reasons. First, TLD routes have been shown to be stable, in fact most popular prefixes are [30]. Second, it is possible to lose reachability to some TLDs without disrupting DNS services because redundancy is built into the system. As these assumptions cannot be made for all prefixes a more conservative mechanism such as PGBGP is required.

X. Conclusions

BGP is vulnerable to bogus routes because the contents of route announcements cannot be easily verified. After nearly ten years, none of the proposed strong solutions have been widely deployed. This paper introduced a simple, incrementally deployable modification to the BGP decision process, called PG-BGP, which can mitigate BGP's most critical vulnerabilities. The basic principle behind PGBGP is that routers should be cautious about adopting a route with new information, such as an unfamiliar origin AS. By avoiding new routes when possible, many attacks can be blocked for long enough to correct the attacks before they cause widespread damage.

We evaluated the performance of PGBGP on two important classes of attack—prefix and sub-prefix hijacks. We show that PGBGP is highly effective at blocking the spread of hijacked routes, even with relatively small-scale deployments. PGBGP can protect 97% of ASs from malicious prefix routes and 85% from bogus sub-prefix routes when deployed only on the 62 core ASs in our study network. If PGBGP were deployed on all ASs, both numbers would exceed 99%. In contrast, today's BGP makes half of ASs vulnerable to a prefix hijack, and 100% vulnerable to a sub-prefix hijack.

These results are significant for several reasons. First, we have shown that delaying the acceptance of new routes is a safe and effective method for reducing the spread of bogus routes to a human time scale. Second, we have proposed and demonstrated an effective method of validating the correctness of suspicious routes. Third, it is incrementally deployable: (1) PGBGP is compatible with the current BGP protocol, requiring changes only to a router's decision rules; (2) Individual ASs have an incentive to adopt PGBGP, as it provides immediate benefit even if other ASs have not deployed it. PGBGP is highly effective, even if only the core ASs adopt it.

References

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

[2] S. Murphy, "BGP Security Vulnerabilities Analysis," RFC 4272, Jan. 2006.

[3] S. A. Misel, "Wow, AS7007!" Apr. 1997, http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html.

[4] V. J. Bono, "7007 explanation and apology," Apr. 1997, http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html.

[5] Renesys Blog, "Con-Ed Steals the 'Net," http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml.

[6] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000.

[7] J. Ng, "Extensions to BGP to support secure origin BGP (soBGP)," *Internet Draft draft-ng-sobgp-bgp-extensions-02*, April 2004.

[8] T. Wan, E. Kranakis, and P. van Oorschot, "Pretty secure BGP, psBGP," in *Proc. Network and Distributed System Security*, 2005.

[9] B. Smith and J. Garcia-Luna-Aceves, "Securing the border gateway routing protocol," in *Proc. Global Internet*, November 1996.

[10] American Registry for Internet Numbers, http://www.arin.net.

[11] RIPE, http://www.ripe.net/.

[12] Asia Pacific Network Information Centre, http://www.apnic.net.

[13] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proc. ACM SIGCOMM*, 2002, pp. 3–16.

[14] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Detection of invalid routing announcement in the Internet," in *Proc. Dependable Systems and Networks*, 2002.

[15] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and Whisper: Security mechanisms for BGP," in *Proc. Networked Systems Design and Implementation*, March 2004.

[16] C. Kruegel, D. Mutz, W. Robertson, and FredrikValeur, "Topology-based detection of anomalous BGP messages," in *Proc. Syposium on Recent Advances in Intrusion Detection*, vol. 2820, September 2003, pp. 17–35.

[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 of interdomain routing," in *Proc. Network and Distributed Systems Security*, February 2003.

[18] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. on Networking*, vol. 9, no. 6, pp. 681–692, December 2001.

[19] M. Caesar and J. Rexford, "BGP policies in ISP networks," *IEEE Network Magazine*, October 2005.

[20] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of BGP multiple origin AS (MOAS) conflicts," in *Proc. Internet Measurement Workshop*, Nov. 2001.

[21] T. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. on Networking*, vol. 10, no. 1, pp. 232–243, April 2002.

[22] J. Karlin, S. Forrest, and J. Rexford, "PGBGP simulator," http://cs.unm.edu/~karlinjf/pgbgp/.

[23] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Trans. on Networking*, vol. 9, no. 6, December 2001.

[24] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, "AS Relationships: Inference and Validation," *ArXiv Computer Science e-prints*, Apr. 2006.

[25] RouteViews, http://www.routeviews.org/.

[26] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The case for separating routing from routers," in *Proc. Future Directions in Network Architecture*, Aug. 2004.

[27] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and Implementation of a Routing Control Platform," in *Proc. USENIX/ACM Symposium on Networked Systems Design and Implementation*, May 2005, pp. 15–28.

[28] Internet Alert Registry, http://cs.unm.edu/~karlinjf/IAR/.

[29] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, and L. Zhang, "Protecting BGP routes to top level DNS servers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 9, pp. 851–860, 2003.

[30] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP routing stability of popular destinations," in *Proc. Internet Measurement Workshop*, 2002.