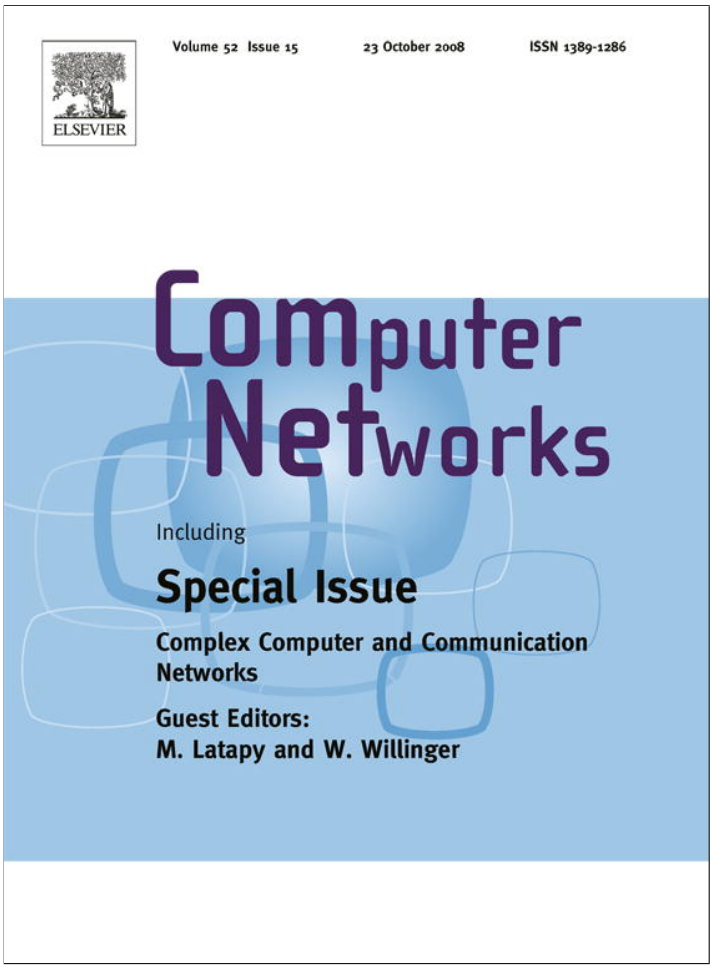

Computer Networks 52 (2008) 2908–2923

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Autonomous security for autonomous systems

Josh Karlin [a,*], Stephanie Forrest [a,b], Jennifer Rexford [c]

[a] Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, United States
[b] Santa Fe Institute, Santa Fe, NM 87501, United States
[c] Department of Computer Science, Princeton University, Princeton, NJ 08540, United States

ARTICLE INFO

Article history:
Available online 25 June 2008

Keywords:
BGP
Interdomain
Anomaly
Security

ABSTRACT

The Internet's interdomain routing protocol, BGP, supports a complex network of Autonomous Systems which is vulnerable to a number of potentially crippling attacks. Several promising cryptography-based solutions have been proposed, but their adoption has been hindered by the need for community consensus, cooperation in a public key infrastructure (PKI), and a common security protocol. Rather than force centralized control in a distributed network, this paper examines distributed security methods that are amenable to incremental deployment. Typically, such methods are less comprehensive and not provably secure. The paper describes a distributed anomaly detection and response system that provides comparable security to cryptographic methods and has a more plausible adoption path. Specifically, the paper makes the following contributions: (1) it describes pretty good BGP (PGBGP), whose security is comparable (but not identical) to secure origin BGP; (2) it gives theoretical proofs on the effectiveness of PGBGP; (3) it reports simulation experiments on a snapshot of the Internet topology annotated with the business relationships between neighboring networks; (4) it quantifies the impact that known exploits could have on the Internet; and (5) it determines the minimum number of ASes that would have to adopt a distributed security solution to provide global protection against these exploits. Taken together these results explore the boundary between what can be achieved with provably secure centralized security mechanisms for BGP and more distributed approaches that respect the autonomous nature of the Internet.

1. Introduction

The routing protocol that connects large IP networks to form a single Internet is the border gateway protocol (BGP). This critical networking infrastructure has significant security vulnerabilities that are well documented but have remained unresolved for many years [1,2]. BGP's vulnerabilities arise from its trust model. By design BGP trusts each individual network (Autonomous System or AS) of the Internet to announce only legitimate and accurate routing information. BGP's trust model allows networks to operate independently of external authority, thus enabling a truly distributed network of self-interested, often competitive, autonomous nodes. The current success of the Internet and widespread adoption of BGP are evidence that this level of trust is not entirely unwarranted.

In spite of BGP's success, inadvertent mistakes and malicious attacks can cause serious problems. A simple typographical error, either inadvertent or intentional, when configuring a router can cause an AS to "hijack" traffic intended for other networks. For example, an AS might announce to its neighbors that it is the destination (origin AS) for a block of IP addresses (prefix) that it does not own, and this false information would then propagate to other networks. We refer to these as *origin AS* attacks. Such attacks are common, and many examples have been documented [3–8]. Hijacks can range in severity from misrouting a small segment of address space, as happened

* Corresponding author.
E-mail address: karlinjf@cs.unm.edu (J. Karlin).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.06.012

in the 2005 Panix incident [8], to diverting a significant fraction of all Internet traffic to a small ISP for several hours [6,9].

Other vulnerabilities exist within an attribute of BGP update messages, known as the AS path. The AS path is a list of ASes that an update message is purported to have traversed. It is used both for loop detection and route selection. Each AS that propagates a route is supposed to add its own AS number to the path and leave the remainder of the path intact. However, a router can be configured to manipulate the path. Such paths are considered *invalid paths*. For example, an AS could append the legitimate origin AS to a forged path to make it look as if the legitimate origin AS originated the path. We show that invalid paths are less harmful than origin AS attacks, but can steal a non-negligible amount of traffic and should be protected against.

Existing proposals for protecting BGP from hijacking and other attacks fall into two broad categories, cryptographic protection and anomaly detection. Cryptographic approaches involve an authenticated registry that maps IP prefixes to their proper origin ASes. The registry would be secured and distributed using a public key infrastructure (PKI). This approach requires global cooperation among the ASes to build and actively maintain the registries. To date, efforts to create such registries [10–12] have suffered from inaccuracy [2] and lack of trust by the operational community [13]. Other impediments include both the need to change the basic BGP protocol and the requirement that all ASes along a path participate in the cryptographic check in order for updates to be verifiable. Despite several credible proposals, cryptographic solutions have not yet been widely deployed.

Given the difficulty of introducing a centralized security solution for BGP [14], it is worth asking how much security an individual AS (node) can achieve without relying on other networks to deploy the same method. This question could be asked of all distributed networks. An ideal security enhancement would be able to both detect and suppress the propagation of origin AS and invalid path attacks. It would require little cooperation from other ASes, minimal (if any) changes to the underlying routers, and it would be simple (and cheap) to adopt.

In Section 4 we describe pretty good BGP (PGBGP), a system that meets all of the above requirements. It includes material from our original system [4] for defending against origin AS attacks as well as recent enhancements [15] to handle invalid paths. PGBGP is comprised of an anomaly detection algorithm that uses the past history of valid routes to discover origin AS attacks and invalid paths. The detection algorithm is combined with a soft-response mechanism that slows the propagation of anomalous routes to human time-scale by temporarily "depreferencing" unfamiliar routes. This gives human operators time to repair the invalid (bogus) routes before they can spread and hijack traffic. Finally, we have designed and deployed an alert system, known as the internet alert registry (IAR). The IAR disseminates notifications of anomalous routes to operators of both the historically trusted and untrusted ASes of each anomalous route, rather than the local operator, because they are in a position to validate and fix an invalid route. In this paper, we modify PGBGP to address invalid paths by detecting new links (edges) between ASes. This paper also gives a formal analysis of PGBGP's correctness and studies the impact of attacks that introduce invalid paths.

This paper makes contributions directly to BGP security research and more generally to the field of complex networks. Specific to BGP, it describes the first distributed approach to BGP security that is comparable (but not identical) in strength to cryptographic approaches. Next, we prove that pretty good BGP can detect and stop the propagation of invalid paths and invalid origin ASes. Third, we simulate the Internet's vulnerability to many of the known BGP exploits when the victim and adversary are randomly placed. By understanding the severity of each type of exploit, researchers can focus upon the most significant problems. Finally, we study possible adoption paths for BGP security enhancements, including those other than PGBGP. Specifically, we show through simulation that a small deployment of the invalid path extensions to pretty good BGP on the 125 largest ASes (0.5% of all ASes) would be sufficient to minimize the global effect (reaching only 0.07–2% of all ASes depending on the type of attack) of randomized BGP attacks. We further show that existing cryptographic solutions would require at least as large a deployment to be effective.

BGP networks exhibit many of the hallmarks of complex networks. First, they are truly distributed, because each node in the network is an autonomously administered network (AS). Nodes often compete with one another, either economically (as in the case of Sprint and AT&T) or politically (as in the case of China and the US), and thus they may have incentives not to reveal information about themselves (e.g., their routing tables) or to cooperate with other nodes. The network is redundant in the sense that there are multiple routes between most source/destination pairs, and the redundancy is introduced through the BGP protocol's local propagation algorithm. We exploit this redundancy in PGBGP's response mechanism. Information about the network topology is not available directly, but resides implicitly in the routing tables of each AS, which are not necessarily publicly available. Thus, there is no centralized routing information or repository of the network topology. A second hallmark of complex networks is dynamics. In BGP networks, change is routine and continual as new links and ASes are added, links go up and down, and ASes are withdrawn. The topology changes more quickly than inferences can be made about overall topology, so it is impossible with current technology to know the actual network graph. Because there is no global view of the network, changes in topology are spread through routing announcements. The announcements spread from neighbor to neighbor, similar to the way disease spreads in epidemiological networks. BGP networks also blend economic with technological constraints. Because customer/provider and peer/peer contractual agreements affect which routes are propagated, BGP networks are policy networks as well as routing networks. Finally, there is opportunity and incentive for malicious agents in the network. As other vulnerabilities in the network are addressed through computer security improve-

ments, the open trust model of BGP is likely to become an inviting target for malicious agents, whether politically or economically motivated. As in other ecological arms races, the appearance of malicious agents will likely lead to inadvertent evolution as novel attacks are countered by new defenses, leading to yet more novel forms of attack. In this paper, we focus on the security of complex networks such as BGP, exploiting network redundancy (multiple routes between most source/destination pairs) and trading off evolvability (how quickly new routes are incorporated into the network) to provide more autonomous security for the autonomous systems of the Internet.

2. Background

This section reviews the BGP protocol and discusses two important classes of BGP attacks, origin AS attacks and invalid paths.

2.1. The border gateway protocol

The border gateway protocol (BGP) [16] is the standard interdomain routing protocol used today. It ensures that each participating network (autonomous system) has a route for reaching every block of IP addresses, known as prefixes. It is a path-vector protocol in which information about network topology is spread only by local contact. Each AS informs its neighbors about its best available route to each destination via update messages. If a new prefix is added to the network, the owner of the prefix announces it from her AS with an *announcement* update message. Similarly, if a prefix is retracted or a BGP session is dropped, ASes retract routes by propagating a *withdrawal* update.

If an AS knows about multiple routes to the same prefix, its router chooses the route that the operator has assigned highest preference to. If multiple routes tie for the highest preference, then a series of tiebreaker rules are applied (such as shortest path length) to select a single route to announce to its neighbors. Generally, all routes learned from a given neighbor are assigned the same preference, which is based upon the cost of directing traffic through that neighbor, rather than performance. For example, the cost of using a route can be determined by the contractual relationship between an AS and its neighbor.

There are three types of AS relationships [17]: customer–provider, peer–peer, and sibling–sibling. In a customer–provider relationship, the customer pays the provider for access to the rest of the Internet. This is economically beneficial to the provider, and the provider has an incentive to assign highest preference to routes learned from its customers. Customers meanwhile have an incentive to avoid provider routes when possible, by assigning them low preference. Peer–peer relationships occur when two ASes agree to share routes and carry traffic for each other's customers at no charge. Because peer–peer routes are free, they are given higher preference than provider routes, but lower than customer routes. Sibling–sibling relationships are used to share all routes between two ASes, as if they were both providers for one another. Unless stated otherwise, sibling ASes are considered a single AS for the remainder of the paper.

ASes are often prevented by contractual agreements from forwarding (exporting) their best routes to all of their neighbors [17]. Routes that are exported in violation of contractual stipulations are considered policy violations, and are one type of invalid path. According to Gao [17], an AS should export routes learned from its peers and providers only to its customers. Routes learned from customers should be exported to all neighbors. Therefore, an AS should not export a route learned from a provider or peer to another provider or peer. An AS that does so is considered to be a *policy violator* and the resulting AS path is a *policy violating path*. Table 1 lists each of the export rules in common practice for future reference.

Even accidental policy violations can cause serious network problems. Providers give highest preference to customer routes. When a customer AS C violates policy and exports a route learned from one provider or peer to another, P, then P will likely choose C's route. Since P will continue to export the route to its providers, a number of large provider ASes would be likely to route through C because it is a customer route. The increased traffic to C might overload C's capacity, preventing the traffic from reaching its destination.

2.2. Origin AS attacks

There are two main classes of origin AS attacks: prefix hijacks and sub-prefix hijacks. Because BGP does not validate the origin AS of an update message, a BGP router can announce any prefix, even those it does not own, which is known as a prefix hijack. For example, a university could announce that it owns a prefix that actually belongs to a financial institution, such as a bank. Those ASes that selected the university's route would send their data to the wrong destination. The university could then use the data however it pleased: it could discard it (known as a black hole); it could read the data and then forward it on to the intended destination [18]; or, it could impersonate the bank's services to gain passwords (such as a website login page).

Because an AS can announce any prefix, a network can accidentally or maliciously announce a subnet of another network's prefix rather than the whole prefix. This is known as a sub-prefix hijack. For instance, an AS could announce 12.0.0.0/9 which is a subnet of AT&T's 12.0.0.0/8. This is a serious form of attack because at packet forwarding time routers will forward traffic to the smallest matching subnet.

An adversarial AS could also announce a larger network, or supernet, of its victim's prefix. Although it has been

Table 1
Standard route export rules

Route learned from	Should export route to
Provider	All customers
Peer	All customers
Customer	All neighbors
Local	All neighbors

Routes learned from providers are propagated to customers while local routes and those learned from customers are propagated to all neighbors.

shown that such hijacks could be used for sending spam from unused address space [19], it could not be used to divert traffic away from proper destinations because routers always forward packets to the smallest matching prefix, and in this paper we do not consider such attacks.

There are many examples of actual origin AS attacks, including the infamous 1997 incident in which a single ISP sub-prefix hijacked the first class-C subnet of every announced prefix causing reachability problems for a large number of networks. On November 30th, 2006 AS 4761 announced at least 4000 prefixes that it did not own [7], including specific prefixes owned by organizations such as banks, universities, and large corporations. More recently, on February 24th 2008, AS 17557 (Pakistan Telecom) sub-prefix hijacked YouTube's (http://www.youtube.com/) website [3]. It is generally thought that most of these attacks were accidental, but they still cause damage and they occur routinely.

It is worth noting that origin AS attacks could be stopped by using only methods available to BGP today. BGP provides programmable filters, in which operators can program their routers to discard routes that violate certain conditions. Filters are used by some providers to ensure that their customers announce routes only for prefixes that they own. If all providers did this, the BGP network would be safe from origin AS attacks. However, many networks do not filter effectively, forcing neighboring ASes to infer the validity of routes that originate from many hops away, an impossible task without an accurate registry. Even careful network operators make mistakes, allowing their customers to announce prefixes they do not own. For example, AS 2914 is well known to run carefully configured filters for its customers, but it was one of the ASes that allowed its customer (AS 4761) to announce Panix's prefix in the well publicized hijack [8].

2.3. Invalid paths

BGP does not verify the path that an update claims to have taken. The path might not have been traversed by the update, or the path might violate a network's contractual policy, or it might not exist. The BGP protocol states that before propagating an update, each AS must prepend its own AS number to the path and leave the remainder of it untouched. An adversary could disobey the protocol and edit the path before propagating it, perhaps to shorten it or append a legitimate origin AS to a path to bypass origin AS verifiers.

A consensus does not exist on what aspects of an AS path should be validated. In this paper, we define an *invalid path* as an AS path in which an edge (pair of consecutive ASes in an AS path) in the path is spoofed, the path violates a contractual policy, or at least one AS in the path has a spoofed AS number. This extends the definition introduced in [20] to include policy violations.

There are many exploits that use invalid AS paths, and each existing proposal for enhancing BGP security considers only a subset of them. For instance, secure BGP (SBGP) [21] does not verify if a path violates policy. On the other hand, secure origin BGP (soBGP) [22] verifies policy but verifies only that the AS path is comprised of known edges, not that the update actually traversed the path. Secure origin BGP is therefore vulnerable to AS number (ASN) spoofs and shortest valid path attacks (described below).

Most security proposals (including SBGP and soBGP) validate the origin AS and the update's path separately. Assuming that the origin AS detector is accurate, then only invalid paths that appear to be originated by the correct ASN could bypass the hijack detector. The most important examples of known BGP exploits that use invalid paths are listed below:

1. *Shortest spoofed path*
   To avoid prefix hijack detection, an AS could erase the entire path between itself and the origin AS before propagating a route. This leaves the apparent (spoofed) edge (Adversary, Origin) at the end of the path. This is also the shortest path possible between the Adversary and the Origin, increasing its chances of being selected by upstream ASes.
2. *Shortest valid path*
   To perform a hijack but avoid having any spoofed edges in the path, an adversarial AS might erase the existing path and prepend the shortest valid path of actual edges between itself and the origin AS.
3. *Redistribution attack*
   If a BGP router is not correctly configured, it could accidentally export routes learned from providers or peers to other providers or peers, causing a policy violation. This is fairly common as many BGP routers export all learned routes to all neighbors by default. The reason that accidental policy violation attacks are harmful is that the providers (and the provider's providers) that the customer might export the route to would be likely to select the customer route for the destination, but the customer might not be able to cope with such a large amount of traffic.
4. *ASN spoof*
   Instead of prepending its own AS number before exporting the route, an adversarial AS might prepend another AS's number instead. An AS could use this to forge a prefix hijack. For instance, if an adversarial AS A wanted to steal the victim AS's (V) prefix P, then AS A could originate an announcement for P using V's AS number. This is a difficult attack to perform because A's peers would likely discard routes that do not have the correct next-hop AS number. Therefore, AS A must either convince its neighbors that it is indeed AS V, convince its neighbors to collude with it, or compromise its neighbor's routers.

Examples of these attacks are given in Fig. 1, and a short description of each type of attack is listed in Table 2.

3. Related work

Improving BGP security has been an active area of research in recent years. Some security proposals use cryptographic signatures to authenticate BGP update messages [21–24], while others rely on anomaly detection [25,26,20,27,28] or introduce verification services [29]. In

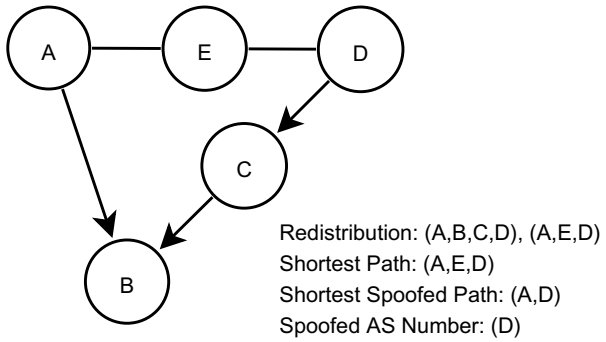

Fig. 1. Examples of invalid paths. Autonomous System A modifies its AS path when exporting routes to gain access to D's traffic. The paths listed in the legend are those that A could send to its neighbors for each type of invalid path attack. Arrows point to customers from providers, and undirected edges represent peer–peer relationships.

Table 2
Invalid path exploits

Exploit name	Category	Procedure
Shortest spoofed path	Spoofed edge	Erase AS path between adversary and origin AS before export
Shortest path	Policy violation	Replace AS path with shortest path of existing edges to origin AS
Redistribution	Policy violation	Export route learned from one provider or peer to another
Spoofed AS number	Invalid AS number	Erase AS path and prepend victim's AS number

this section, we briefly discuss the work most relevant to pretty good BGP.

Secure BGP (SBGP) [21] was the first attempt at a comprehensive approach to BGP security. SBGP requires a public key infrastructure (PKI) to ensure prefix and ASN ownership. Each announced route is signed by the originating AS to ensure that the router represents the ASN that it claims, and that the ASN owns the prefix in question. Recipient ASes that propagate the route also sign the path, to verify that the advertised path is the path actually traversed by the update.

Secure origin BGP (soBGP) [22] takes a more decentralized, cryptographic approach. Secure origin BGP uses a distributed PKI that contains prefix ownership and policy objects. The policy objects are used to declare that two ASes are neighbors and the prefixes that should be propagated over the edge. Secure origin BGP verifies an update by first ensuring that the origin AS of the path is correct for the prefix using a PKI lookup, and then it verifies that the AS path is plausible given the sequence of edges for the prefix and the policy objects.

A third cryptographic approach known as psBGP [23] employs a centralized PKI to distribute AS number certificates, and a web-of-trust PKI to designate prefix ownership. Its security guarantees are similar to both soBGP and SBGP. All three cryptographic approaches have clearly defined security properties when ubiquitously deployed. Unfortunately, it is unclear how they should handle paths that cannot be fully verified using the PKI when only some of the ASes have adopted the solution.

Kruegel et al.'s [20] topology detector verifies edges using out-of-band information. This detector queries WHOIS registries to determine the geographic location of networks and ensures that both ends of an edge have physical presence in the same area. This is most useful for for small networks that are not distributed over a large geographical area.

Whisper [25] and MOAS lists [30] (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. Anomalous routes are detected when the community attribute of routes for the same prefix does not match. Unfortunately, in ASes that have not deployed the protocol enhancements, the routers are likely to strip the community tag as it propagates.

The PGBGP edge detection algorithm was originally described in [15]. A similar detection algorithm was proposed independently in [31] but did not include an automated response.

Wang et al. [32] developed a BGP anomaly detector for use with root/gTLD domain server routes. They suggest filtering out all but the most durable (and verified) routes to these addresses. This is feasible for two reasons. First, root/gTLD routes have been shown to be stable, in fact most popular prefixes are [33]. Second, it is possible to lose reachability to some root/gTLD prefixes without disrupting DNS services because alternate root/gTLD addresses exist.

4. Pretty good BGP

This section describes pretty good BGP (PGBGP), an anomaly detection and response system. Pretty good BGP combines a conservative anomaly detector with a soft-response to ensure that as many attacks are detected and suppressed as possible without degrading routing behavior. New origin ASes and new *directed* edges are considered anomalous. In response, PGBGP takes advantage of the AS network's natural path redundancy and temporarily lowers the local preference of anomalous routes, encouraging known trusted paths to be used instead while anomalous routes are vetted. This automatically mitigates the effect of short-term attacks and misconfigurations. To help suppress longer attacks, we describe a notification system known as the internet alert registry (IAR) that informs the operators involved with and affected by anomalous routes of the problems so that they can be fixed quickly.

4.1. Anomaly detection

The PGBGP detection mechanism is simple. Recent routing information is used to construct a database of normal (trusted) network characteristics. New network characteristics that deviate from the trusted database are treated as anomalous. Routes which contain anomalous characteristics are considered *anomalous routes*. To maintain a database of normal network characteristics over time, new routing information is added after a short delay while old information that is not active is removed.

There are two route characteristics that the normal database N monitors. First, a mapping between prefixes and their origin ASes (prefix pairs) is used to detect origin AS attacks [4]. Second, an inferred AS network topology is used to detect invalid paths [15,31].

These characteristics can be extracted from BGP announcement updates. The prefix pair of an update u is ($u.prefix$, $u.originAS$) where $u.prefix$ is the prefix that the update addresses and $u.originAS$ is the last AS in the AS path.[1] To construct a partial network topology, directed edges in the network are inferred from the update's AS path, $u.as\_path$. Given an AS path $u.as\_path = (v_1, v_2, \ldots, v_n)$, the directed edges of the path are $\{(v_i, v_{i+1}) | 1 \leq i < n\}$. PGBGP monitors directed edges as opposed to undirected edges because in many cases one direction can be seen in valid paths by a single observer, but not the other. For instance, a Tier 1 AS (an AS with no providers) may see an edge from an AS C to its provider P (P,C), but should never see the reverse edge (C,P) according to Table 1.

Initially, a PGBGP router's normal database, N, is empty.[2] The prefix pair and edges are extracted from each received update for h days and added to N. After h days, new prefix pairs or edges (and the routes that contain them) are considered anomalous for twenty-four hours. Anomalous network characteristics found within the router's tables (RIB) after h days will be added to N. To remove stale information, all trusted network characteristics not present in any of the RIB's paths within the last h days are removed from N.

We consider two special cases. First, if an update has a new origin AS for a prefix in N, but one of the normal origin ASes is present in the AS path before the origin AS, then the route's prefix pair is not considered an origin AS anomaly. This is because forwarded data along the path will reach one of the normal ASes before it reaches the anomalous origin AS. The second special case pertains to routes with prefixes that are not in N. If the new prefix is not a sub-prefix of any existing prefixes in N, then the new origin is not considered anomalous as it cannot be used to steal data.

Finally, in Section 6, our experiments show that the value of h should be different for edges and prefix pairs to reduce the number of false positives. Therefore, we introduce two window lengths for the normal database, h_{prefix} and h_{edge}. Given the experimental results, we suggest values of ten days for h_{prefix} and sixty days (two months) for h_{edge}.

4.2. Response

An ideal response mechanism would effectively hinder the propagation of bogus routes without interfering with normal network operation in the case of a false positive. Pretty good BGP is the only anomaly detection algorithm to incorporate such a response. We achieve this by decreasing the likelihood of an anomalous route being used and propagated, without precluding it.

When presented with multiple routes for a given prefix, the BGP selection mechanism applies a standard set of tie-break rules to select a single best route. The first rule selects the routes of the highest local preference. By lowering the local preference of anomalous routes to zero, we can suppress their use so long as a trusted route for the prefix exists.[3] After providing enough time (twenty-four hours) for operators to fix (withdraw or filter) the route, if it is indeed bogus, the route is restored to its normal local preference.

This soft-response does not affect network reachability. If only anomalous routes exist for a prefix, then they will be used. We show in Section 6 that most anomalous routes are short-lived, and suppressing them has little impact on network operation. In fact, these routes are likely the result of churn during BGP convergence, and are best avoided.

Unfortunately, our soft-response mechanism cannot be applied to sub-prefix hijacking anomalies. This is because the sub-prefix hijack necessarily introduces a new prefix, and all routes for this prefix will be destined to the hijacker's AS. Instead, PGBGP delays all routes which contain new sub-prefix hijack anomalies from entering the router's tables for twenty-four hours. In the meantime, traffic destined to addresses in the sub-prefix will continue to be forwarded toward its super-net's origin AS. If the super-net is withdrawn during this period, then the anomalous sub-prefix routes will be used.

The sub-prefix hijack response could cause reachability problems in the following unlikely scenarios: If a customer AS C uses a sub-prefix of it's provider P's space, but temporarily loses its connection to P, it might try to announce the sub-prefix over a backup-provider link. Since the sub-prefix is not typically announced (perhaps it is aggregated by P), it may be viewed as a sub-prefix hijack, and data will continue to be forwarded to P. If P has no means of reaching C, then the data would be discarded. This scenario is unlikely as typically a customer with multiple upstream providers would announce the more specific prefix through both providers at all times (with a padded path on the backup route to discourage its use). A second scenario in which reachability could be lost is if a customer AS changes providers but keeps the old provider's sub-prefix (this is discouraged by many ASes). So long as the customer maintains connections to both the old and new provider for at least one day (which is typical) the new sub-prefix (which was not previously announced with the old provider) will be accepted as normal before the old provider is dropped.

4.3. Intuition behind PGBGP

Here we describe the instances in which PGBGP can successfully identify origin AS and invalid path attacks. We assume that the normal database, N, is clean. That is, the database does not contain incorrect network characteristics from invalid paths. This assumption simplifies our explanation. When deployed, it is expected that the normal database might initially be corrupted, but would gradually become more reliable as anomalous routes were detected and fixed.

Because N is clean, any update with a prefix hijack u, must include a prefix pair that does not exist in N. The same

[1] If the route has an AS set, which is rare, the origin AS is the last AS before the AS set.

[2] Subsequent reboots could restore the database from disk.

[3] This response is best applied to ASes of high degree, as they are likely to have stable alternate paths to select from.

reasoning can be applied to sub-prefix hijacks so long as the super-net exists within N when u is received. If the super-net has not been announced within h_{prefix} time, then the sub-prefix hijack will fail to be detected. To reduce anomalies, PGBGP does not consider new origins as anomalous if a trusted AS is along the path. This exception reduces the number of anomalies by about 16%, but makes PGBGP's origin AS detector vulnerable to shortest valid path attacks, and may be omitted in future work to improve security.

Next, we enumerate PGBGP's ability to detect all three classes of invalid paths:

1. *Spoofed edges.* Since N is clean, any update with a spoofed edge will contain a directed edge that does not exist within N.
2. *Policy violations.* Here we show that any update u with a policy violation in $u.as\_path = (v_1, v_2, \ldots, v_n)$ contains a directed edge not in v_1's normal database N unless v_1 is a transitive customer of the closest policy violator in the path to v_1, AS v_v. We define a *transitive customer* of an AS a as the union of set $\{a\}$ with all of a's customers and their customers, ad infinitum. We show that edge (v_v, v_{v+1}) cannot be a member of v_1's normal database N as it could not observe the directed edge in a policy valid path.*Proof by contradiction*: Let v_v be the closest policy violator to v_1 in path $u.as\_path$. We assume that v_1 is not v_v's transitive customer. Let path $Z = (v_1, \ldots, v_v, v_{v+1}, \ldots)$ be a valid path received by AS v_1. We know that v_{v+1} is v_v's provider or peer by definition of a policy violator. According to Table 1, routes learned from providers or peers can only be propagated to transitive customers. Since v_1 is not v_v's transitive customer, and v_v learned the route from its peer or provider (v_{v+1}), path Z is a policy violating path.
3. *ASN spoofing.* Spoofed AS numbers are especially difficult to detect with local information. However, unless the spoofer can compromise or collude with one of the victim's neighboring ASes, a spoofed edge will be introduced into the path. Let v represent the victim's AS number and let n represent any of the adversary's neighbors AS numbers. Then all ASN spoofed paths will include directed edge (n, v), which is a spoofed edge not in the recipients normal database unless the victim is AS n's neighbor.

To summarize, PGBGP can detect all prefix hijacks and sub-prefix hijacks (unless the super-net has been withdrawn), spoofed paths, policy violations for all but customers of the violating AS, and many instances of spoofed ASNs.

4.4. Algorithm overhead

To be useful, PGBGP should run on today's routing hardware. PGBGP's processing requirements are very low. The normal database can be integrated into the routing software's RIB so that no additional lookups are needed for origin AS detection. The normal database of edges has to be stored in a separate data structure, such as a hash table. Insertions into the table (which can be expensive) are rare since new edges are added to the AS graph relatively infrequently.

PGBGP's memory requirements are also low. Each edge or prefix pair requires only three integers of storage. These are the time the object was last observed, the number of instances of the object in the router's RIB, and the object's depreference time. Also, each anomalous route must enter a 24-hour queue to ensure that its preference is readjusted. Each queue entry requires a pointer to the affected route and a depreference time.

There are not many edges or prefix pairs to store in the normal database. It has been shown that the distribution of AS degree (number of neighbors) follows a power law, where most ASes have few neighbors (an average of six) [34,35]. Also, the number of origin ASes per prefix is believed to be a low constant value. With 250,000 prefixes in use at the time of writing, there are likely less than 500,000 prefix pairs. The size of the normal database should be less than ten megabytes which is reasonable for today's routers.

4.5. Alert notification

Pretty good BGP is capable of mitigating short-term attacks and misconfigurations autonomously. An additional mechanism is needed for longer attacks. The internet alert registry (IAR) [36] is a distributed notification system that can inform networks when their traffic is being stolen and when they may be stealing traffic from other networks. Once informed, the networks can confirm the validity of an anomalous route and take action to fix it if necessary. So long as the anomaly is withdrawn within twenty-four hours, it will not enter a PGBGP router's normal database.

The IAR is an opt-in service that runs the PGBGP algorithm on public feeds of BGP updates from Réseaux IP Européens (RIPE) [11] and RouteViews, [37] and distributes e-mail alerts to the ASes affected by each anomaly. Although it is currently hosted in one location, the IAR can be run from many locations, with different feeds, to increase robustness.

Rather than receive notification of all anomalies for their AS, operators could optionally download the IAR Tracker utility from the IAR website to filter out false positives. The IAR tracker periodically compares the operator's BGP configuration against an RSS feed of alerts posted on the IAR's website. When an alert is discovered that disagrees with the configuration, a notification e-mail is sent.

The IAR is not the only service of this type [38,39]. Any alert system able to notify ASes about origin AS attacks or invalid paths could be substituted or added to the IAR. What is important is that operators have the opportunity to discover and fix attacks before they can propagate.

In the case that an adversarial AS is unwilling to withdraw its bogus routes, the problem could be taken to public forums [40–44]. The network operator forums are very active, and the operators are primarily concerned with network reachability. If an adversarial AS is shown to be "misbehaving," other networks might filter out the adversary's anomalous routes or even de-peer it from their network. For instance, during the YouTube hijack [3], many ISPs responded to the hijack by filtering out Pakistan Telecom's sub-prefix, effectively stopping the hijack within two hours.

5. Incremental adoption

Any new version of BGP software is likely to be adopted incrementally. The experiments in this section quantify the effectiveness of PGBGP when only a subset of ASes adopt it. We compare PGBGP's response to that of an ideal BGP security solution, one that recognizes and discards all bogus routes with 100% accuracy.

5.1. Experimental set-up

To simulate PGBGP's defenses against attack, we created the BGP simulator (BSIM) [45]. BSIM is a route propagation simulator freely available under the GPL license. It takes as input a user-specified topology (including inferred relationships) and simulates the propagation of route announcements across the network according to the export rules defined in Section 2.[4] Ties between routes are first decided by relationship type, then path length, and finally by the neighbor's AS number, similar to the BGP decision process [16].

For our simulations we used the topology and relationships provided by the AS Relationships Dataset [46] built on the 2nd of February 2007. The inferred topology describes 48,986 edges between 24,267 ASes. We chose to simulate BGP propagation over an inferred graph of the Internet's AS network over synthetic networks because existing synthetic models do not accurately capture both the Internet's structure and BGP routing policies. The complete AS topology is unknown. It is believed that the inferred topology contains a significant fraction of customer–provider links, but many peer–peer links are likely missing from the inferred topologies. It is possible, that such peer–peer links would lessen the impact that the Tier 1 ASes have on BGP routing as predicted by our experiments.

We extended the BSIM framework to support both PGBGP and the idealized *perfect detector*. The perfect detector is a "black box" that discards all invalid routes, never making a mistake. It is therefore the best security mechanism that an AS could deploy. Each simulated router can run as a normal BGP router, a PGBGP router, or a router with perfect detection. Finally, we added all of the attack scenarios described in Table 2 as well as prefix and sub-prefix hijacks into BSIM.

Within BSIM, an attack is simulated in two steps: initialization and attack. To initialize the network, each router's BGP routing table is cleared and its protection status is assigned as either none, PGBGP, or perfect. The adversarial and victim ASes are distinct and chosen uniformly at random from the network. Then, the victim AS announces its address blocks to prime the history-based registry of each PGBGP-enabled router. For the second step, at time h (h_{origin} or h_{edge} depending on the attack type) the adversarial AS announces an invalid (bogus) route to steal the victim's traffic. After the routers have selected their best routes, ASes that select a path that includes the attacking AS are counted as having been hijacked. For simplicity, we consider all routes that include the adversary's routers after the attack to be bogus, even if the adversary's router was used before the attack to reach the destination.

Our experiments report attack effectiveness – the fraction of ASes that erroneously select a route through the attacker – for varying levels of PGBGP deployment. In these experiments we systematically deploy PGBGP in ASes in order of decreasing node degree, starting with the AS of highest degree. This is because it would be easier to convince a small number of large ASes (even though they have thousands of BGP routers) to adopt a new protection method than 25,000 smaller ASes. The large service providers are known to upgrade their routing software frequently (PGBGP can be installed as a software upgrade) and follow the latest trends and best common practices.

5.2. Unprotected networks

We simulated all four attacks described in Section 2 as well as prefix and sub-prefix origin AS attacks on an unprotected BGP network. The routers do not perform ingress filtering, and they do not have any security mechanism deployed. This provides an upper bound on how damaging each attack type could be. The results are shown in Fig. 2, where the x-axis shows the type of attack, and the y-axis is the fraction of ASes that selected a route through the adversarial AS.

As the figure shows, sub-prefix hijacks pose the most significant threat. This is expected because a new sub-prefix propagates to every AS and is always selected because it is the only available route for the prefix. Prefix hijacks are also a serious threat. On average, prefix hijacks convince roughly half of the ASes to misroute their traffic.

Assuming some form of origin AS protection, adversaries would then have to use invalid path attacks to steal data. Of the invalid path attacks, it is surprising that policy violation attacks (shortest path and redistribution, as summarized in Table 2) fare very poorly. Because a customer AS could have many providers, which in turn have have many large providers, and each of these providers prefers routes from customers, it seemed likely that such attacks would be significant. Instead, on average, the adversary in each attack convinced only 4% of the network to route through it. This is likely due to the extreme lengths of the adversary's paths. We had expected that the shortest fake edge attack would not be as successful as a prefix hijack. This is because the fake edge attack has a longer path and is less likely to be selected. Finally, ASN spoofing is equivalent in impact to a prefix hijack. This is an expected outcome as the paths are of equal length and unprotected networks do not verify AS numbers.

5.3. PGBGP's effectiveness

Here we study PGBGP's effectiveness at stopping attacks with partial deployment. Fig. 3 compares PGBGP to the perfect detector for the different attack types. In each panel, the x-axis shows the number of ASes (out of 24,267 total) running PGBGP (or the perfect detector), in order of decreasing node degree, and the y-axis shows

[4] BSIM also respects sibling relationships, which are included in the CAIDA data set.

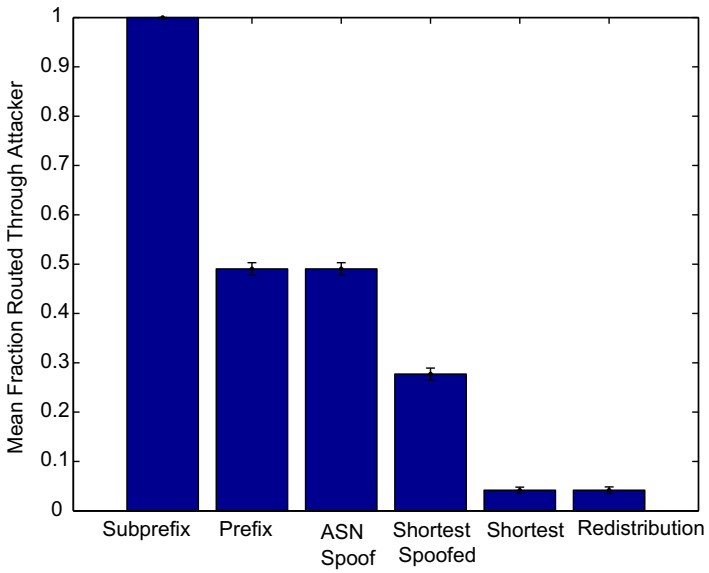

Fig. 2. Effectiveness of each synthetic attack against a network of ASes without PGBGP deployed and no ingress filtering using the standard export rules. The *x*-axis describes the form of attack simulated while the *y*-axis represents the fraction of ASes that routed through the adversarial AS after 500 simulated attacks. Error bars represent standard error of the mean.

the fraction of ASes that choose routes that pass through the adversarial AS. Although the PGBGP automated response depreferences routes while the perfect detector actually discards them, our results show that their is a negligible difference when used on large ASes with many alternate routes. This suggests that PGBGP's softer depreferencing mechanism could be as effective as discarding routes outright (which soBGP and SBGP do), while retaining the ability to tolerate false positives.

For most attack scenarios, running PGBGP on only 125 (0.5%) of all ASes would suffice to protect the entire Internet from both invalid path and origin AS attacks. The same number is required for the perfect detector.

5.4. Propagation of anomalous routes

If an anomalous route is not withdrawn within time *s*, it is accepted by the PGBGP routers and propagated to the next level of ASes. We show in Fig. 4 how anomalous routes spread as a function of time for the sub-prefix hijack. Other attacks have similar results (data not shown). The bottom line represents the initial response of the network to an attack. After time *s*, the route is accepted as normal and propagated further, shown by the second line from the bottom. This process is repeated for a total of four iterations. Our simulations suggest that it could take three time periods (3s) on average for the route to propagate fully if 125 large ASes were running PGBGP.

Fig. 4 represents a worst case propagation scenario. Many false positives are overcome quickly and propagated to the rest of the network. For instance, if an AS changed providers but kept its prefix, its (prefix, origin AS) would change and be considered anomalous by PGBGP. However, PGBGP will select this route if only anomalous routes are available. Similarly, new edges (e.g., backup links) which become available due to link failure would not be hindered if no alternative existed.

6. Analysis of PGBGP anomalies

As with any anomaly detection method, some legitimate network characteristics will be labeled as anomalous (false positives). Because of PGBGP's soft-response, reachability is typically not affected, however. This section describes an experiment in which we ran PGBGP on four months of public BGP update feeds and discovered that most anomalous network characteristics are quickly withdrawn. We predict from this experiment that depreferencing short-term routes would have little negative impact in practice, as most affected routes are misconfigured, non-optimal routes discovered during convergence, or attacks. Next, we estimate how many new network characteristics would likely be experienced by routers on a daily basis, and show how to tune the parameter *h* to reduce this value. Finally, we evaluate the number of alert notifications ASes would likely receive from the IAR, and find that, on average, the number is low (0.03 alerts per day).

6.1. Experimental set-up

The routers of each AS have a unique perspective on the Internet's routes. Predicting PGBGP's behavior on a particular AS is difficult without access to feeds of its BGP update messages. Instead, we ran PGBGP's detection algorithm against four months of publicly available BGP updates to estimate how many new network characteristics might be labeled as anomalous per day based upon the size of the router (interpreted as the number of update streams) and history length, *h*.

The BGP update streams were collected from the RouteViews [37] project at the University of Oregon. RouteViews collects BGP update messages from many routers scattered around the world, including backbone

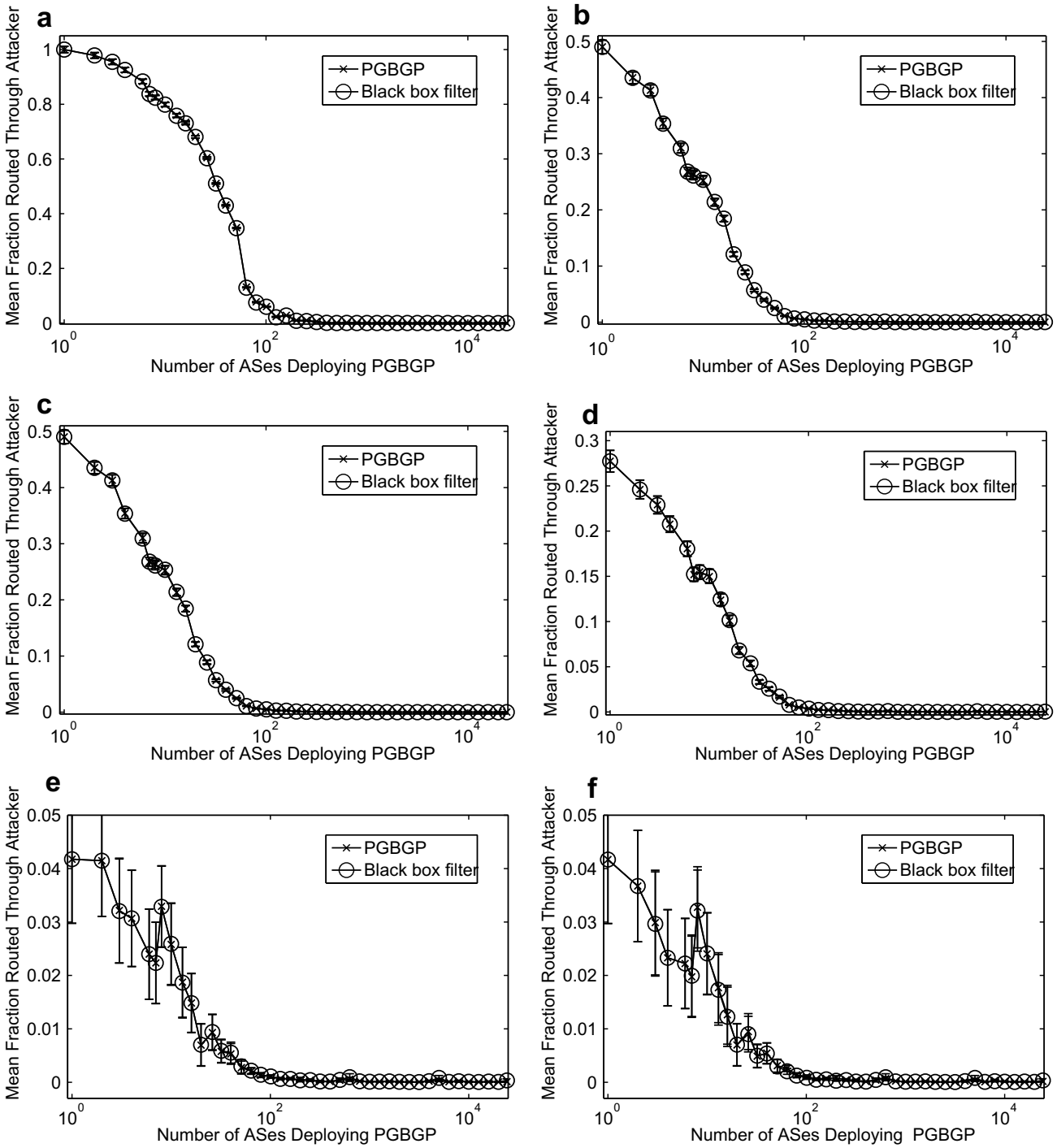

Fig. 3. Effectiveness of each synthetic attack against networks protected by PGBGP and the perfect detector. The results of the two detectors are nearly identical. The x-axis is log-scaled (and shifted up by one to show $x = 0$) and represents the number of ASes that have deployed the PGBGP (or the perfect detector). The y-axis is linearly scaled and represents the number of ASes that selected a route that included the adversary's AS. Error bars show the standard error of the mean over five hundred runs. (a) Sub-prefix hijack, (b) prefix hijack, (c) ASN spoof, (d) spoofed edge, (e) prepended shortest path, (f) redistribution attack.

routers in large ASes. The data set consists of all BGP updates from September 1st 2006 through December 31st 2006 inclusive from the RouteViews2 server, which includes over 40 BGP sessions.

We measured the rate at which anomalies were discovered over the four-month period with varying h values and number of router feeds (neighbors). Each anomaly corresponds to a single alert from the internet alert registry. To simulate BGP routers of different size (1–10 external neighbors), we selected individual feeds (from unique ASes) from the data in decreasing order of size. The size of a feed is determined by the number of updates propagated by the peer in the data set. The first h days were used to initialize the normal database N, and the remaining days were used to monitor for anomalies.

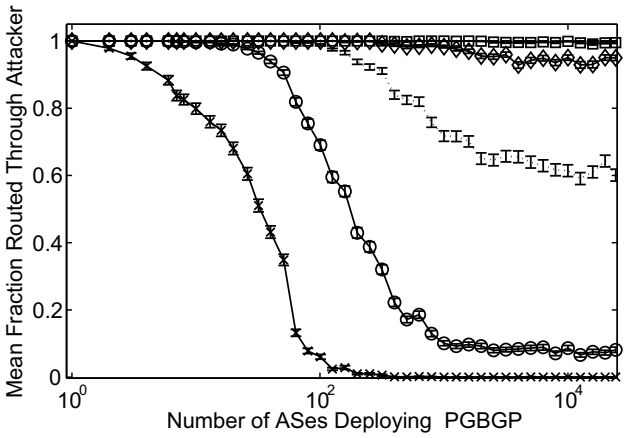

Fig. 4. Effectiveness of each synthetic attack against a network of ASes over time. The x-axis is log-scaled (and shifted up by one to show $x = 0$) and represents the number of ASes that have deployed the security solution. The y-axis is linearly scaled and represents the number of ASes that have selected a route that includes the adversary's AS after convergence. The bottom plot represents the time of the attack and each subsequent plot above it represents an increase of s simulated time. Error bars show the standard error of the mean.

6.2. Most anomalies disappear quickly

On the largest BGP feed, we recorded the time at which each new network characteristic was first observed, and the time that it was last observed during the twenty-four hour depreference period. Anomalies that were withdrawn before the depreference period ended are likely due to misconfigurations, short-term attacks, or convergence churn.

Fig. 5 shows the results of this experiment. Panel a of the figure shows that new network characteristics either disappear from the RIB quickly (within one hour) or remain the full twenty four hours. Nearly 50% of new edges are withdrawn from a router's RIB within one hour of being identified as anomalous. By the twenty-four hour mark, panel b shows that roughly 70% of the anomalies have disappeared. New prefix pairs that could be prefix hijacks behave similarly. This suggests that the observed anomalies are highly correlated with attacks or misconfigurations. Interestingly, most (60%) new sub-prefixes remain in the RIB for at least twenty-four hours. We speculate that new edges and prefix origins often occur from convergence churn, whereas sub-prefixes usually do not.

6.3. Number of anomalies

This sub-section discusses the number of anomalies a router is likely to experience over time, given connectivity (number of neighbors measured by the number of streams), and different values for h. PGBGP has three tunable parameters, s, h_{prefix}, h_{edge}. We set s to twenty-four hours to allow operators time to respond to alerts. The history window h determines how recently an origin or edge must have been observed to be considered normal. The values of h_{prefix} and h_{edge} were chosen to minimize the number of anomalies and keep the history window relatively small (so the database is current).

Fig. 6 shows the number of new prefix pairs (possible prefix hijacks) compared to the number of BGP streams and the value of h_{prefix}. Larger values of h_{prefix} decrease the number of anomalies slightly. Adding streams does not significantly increase the number of anomalies, except for the tenth stream, which introduced a significant number of anomalies. This is because that stream included 4,035 prefix hijacks by AS 4761 on November 30th of 2006 [7]. These hijacks include prefixes owned by eBay, the Bank of New York, Cisco, Princeton University, and the University of New Mexico.

The number of new (prefix, origin AS) pairs attributed to sub-prefix hijacks is shown in Fig. 7. Unlike with prefix

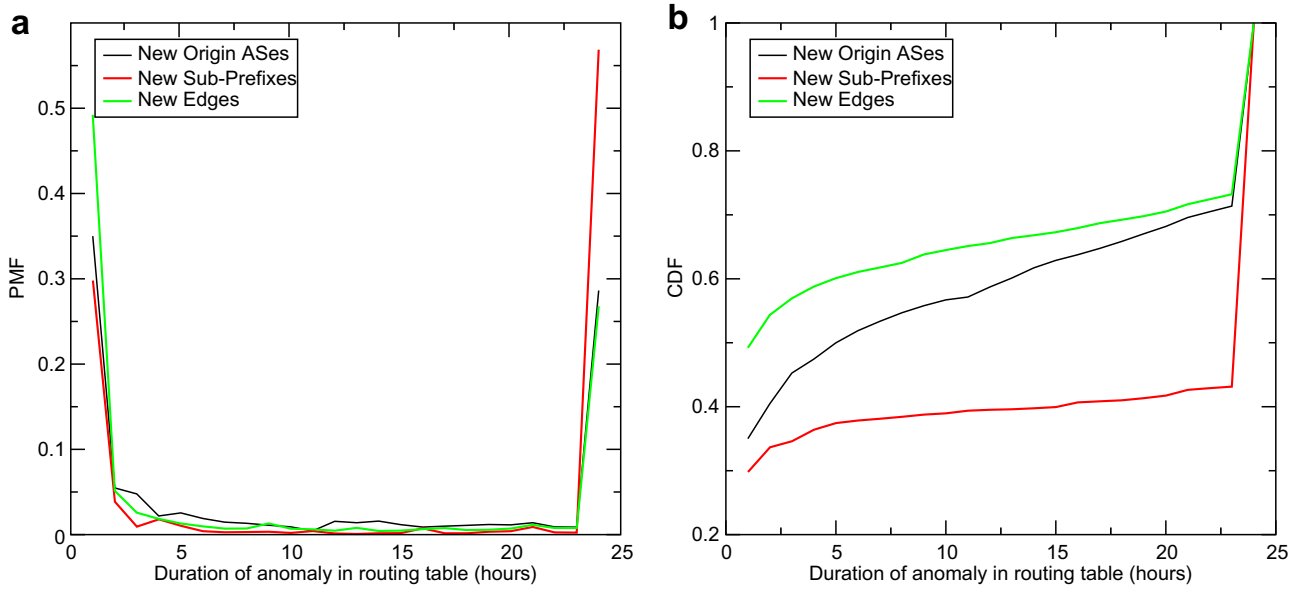

Fig. 5. Length of stay in the RIB for anomalies. Anomalies that exist within the RIB at twenty-four hours are added to the normal database and considered trusted. Panel (a) shows the probability mass function while panel (b) shows the cumulative distribution function. Only the largest feed was used for this experiment.

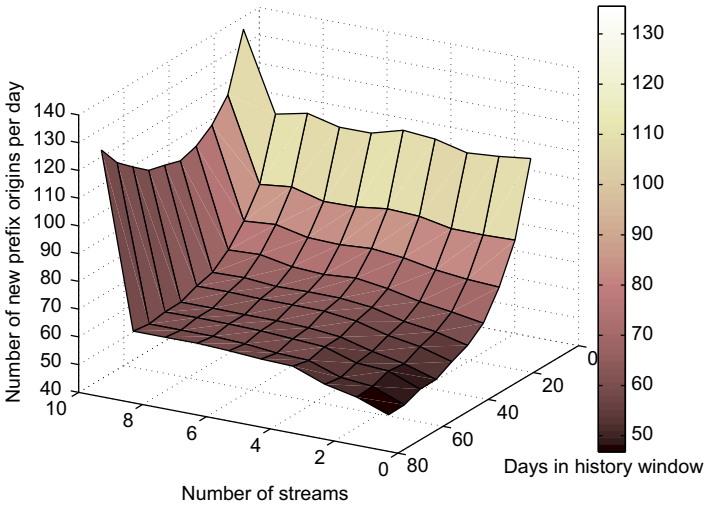

Fig. 6. The number of prefix hijack anomalies, or alerts, that PGBGP observed during the four month time period. The initial h days were used to initialize the normal database. The figure represents a parameter sweep of the number of BGP streams and the duration of the history period h.

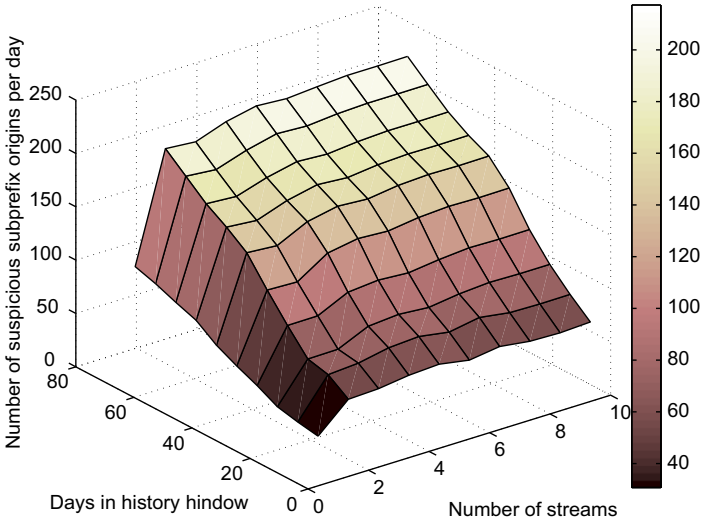

Fig. 7. The number of sub-prefix hijack anomalies, or alerts, that PGBGP observed during the four month time period. The initial h days were used to initialize the normal database. The figure represents a parameter sweep of the number of BGP streams and the duration of the history window h.

hijacks, increasing h_{prefix} increases the number of sub-prefix anomalies. Given Figs. 6 and 7, we chose ten days for h_{prefix} to keep the history short and minimize the total number of anomalies. We chose a single value for hv as opposed to one for prefix hijacks and another for sub-prefix hijacks for simplicity. To further reduce the number of alerts, these values could be set independently. The number of sub-prefix alerts would also be reduced if we filtered out all routes more specific than/24 and less specific than/8. Many BGP routers adhere to this practice to decrease their table size. Our experiments included these routes because they generally should not be propagated (e.g. route leaks), and are interesting to study.

Fig. 8 shows the number of anomalous edges observed per day compared to h_{edge} and the number of neighbors. As the number of neighbors increases, the number of anomalies due to new edges decreases. This is probably because, over time, the router is exposed to more legitimate edges as routes change. If PGBGP were adopted first by the largest ASes with the most neighbors, this would be beneficial. Similarly, as the length of h_{edge} increases, the number of anomalies due to new edges decreases. This analysis suggests that h_{edge} should be set to 60 days (roughly two months).

In future experiments, once the internet alert registry has attained additional feeds and data, the values of h could be adjusted. It would also be interesting to use adaptive algorithms to determine appropriate values of h for each router.

With parameters of $h_{\text{prefix}} = 10d$, $h_{\text{edge}} = 60d$, and one stream, there are an average of roughly 340 anomalies per day, of which 240 are short-term and one hundred are long-term. If the IAR sent one e-mail per anomaly to each victim and adversary AS, then the average AS would have received 0.02 alerts per day with a standard deviation of 0.18. Large ASes, such as the "Tier 1" providers (AS numbers 1668, 7018, 3549, 3356, 701, 2914, 209, 3561, and 1239) would have only received 4.24 alerts per day (with a standard deviation of 2.33).

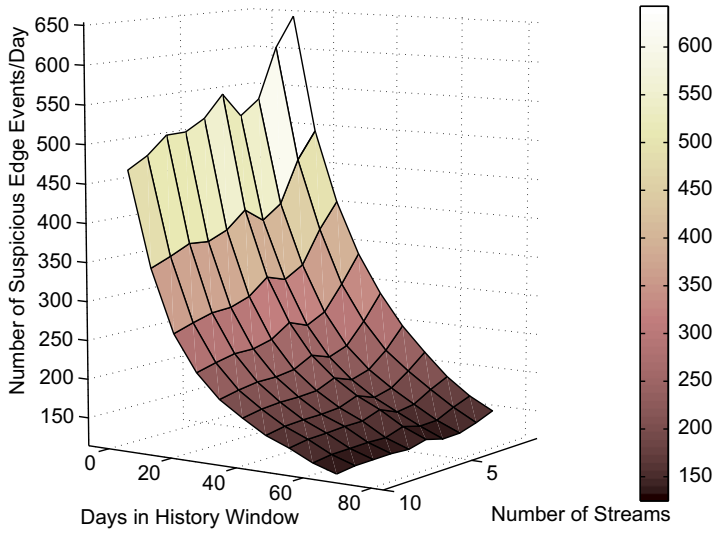

Fig. 8. The number of edge anomalies, or alerts, that PGBGP observed during the four month time period. The initial h days were used to initialize the normal database. The figure represents a parameter sweep of the number of BGP streams and the duration of the history period h.

7. Comparison to other BGP security approaches

If deployed ubiquitously with an accurate PKI, SBGP and soBGP could provide more comprehensive security than PGBGP. Table 3 shows the strengths and weaknesses of each protocol. Considering the data shown in Fig. 2, it appears that SBGP's weakness detecting policy violations is not a great concern because policy violations affect only about 5% of the network, whereas spoofed AS numbers are significantly more harmful on average. However, this analysis does not account for the relative frequency of each type of exploit. Policy violations are likely more common as most routers are configured by default to propagate all learned routes to all neighbors. This means that routes learned from providers or peers, by default, will be propagated to other providers and peers. On the other hand, ASN spoofing requires routers on both ends of a connection to be misconfigured.

As discussed earlier, an effective security mechanism for distributed networks must also have a plausible path for adoption (Table 4). One aspect of this issue is what security is provided if the mechanism is deployed on only some nodes in the network. This is problematic for methods like SBGP that require each AS to sign route updates as they propagate. When only a fraction of nodes deploy SBGP, then AS paths will have holes in their signature chains, making them unverifiable. Participating ASes would be able to sign for the origin AS of the path, and even verify some edges, but there is no guarantee that the extra signature attributes would not be stripped by malicious or non-participating ASes. Similarly, secure origin BGP cannot verify routes unless every AS in the path properly updates the soBGP registry. On the other hand, PGBGP could detect short-term misconfigurations and attacks for all prefixes and edges, for any AS that deployed it. Preventing long-term attacks requires IAR alert monitoring and response only by the networks that are concerned about their own security.

Table 3
Comparison of BGP security protocols when ubiquitously deployed

	SBGP	soBGP	PGBGP
Invalid origin AS	Yes	Yes	Yes
Policy violations	No	Yes	Partial
Spoofed AS numbers	Yes	Partial	Partial
Spoofed edges	Yes	Yes	Yes

Table 4
Comparison of BGP security protocols when partially deployed

	SBGP	soBGP	PGBGP
Invalid origin AS	Partial	Partial	Yes
Policy violations	No	Partial	Partial
Spoofed AS numbers	Partial	Partial	Partial
Spoofed edges	Partial	Partial	Yes

Although PGBGP effectively prevents the propagation of short-term attacks, stopping long-term attacks requires operator intervention, as PGBGP eventually returns anomalous routes to normal preference. Secure Origin BGP and Secure BGP do not have this limitation, as they discard bogus routes immediately (although it is unclear when this would happen if the protocol were only partially deployed). Instead, pretty good BGP relies implicitly upon the community of operators to punish non-compliant (misbehaving) networks via filters and possible de-peerings. We believe that this is a reasonable assumption as most networks ultimately wish to achieve maximum reachability to legitimate destinations for their customers.

Finally, we consider the feasibility of deploying each security mechanism in the absence of a global authority that can dictate its adoption. In a distributed system of self-interested ASes, a new mechanism will be adopted only if there is an incentive for each individual AS to do so. This issue can be framed by asking the question: What advantage, if any, is conferred on early adopters? In the case of SBGP there would be little incentive for individual ASes, as many ASes must agree to deploy it before it can provide substantial security benefits. Since the infrastructure costs would likely be non-negligible, it might even be financially advantageous to be the last adopter of SBGP. Similarly, soB-

GP would require community consensus to maintain a reliable and distributed PKI. Because pretty good BGP was designed for incremental adoption, it is no surprise that it has several advantages on this dimension. It would likely be cheaper to deploy (as it does not require a change to BGP, simply a change to the preference rules), it would have dramatic effect even if deployed on only 100 ASes, its mechanism is simpler than the SBGP and soBGP (no consensus or PKI required), and it provides more advantages for early adopters (protection from all short-term attacks).

To summarize, SBGP and soBGP would provide the most comprehensive security if ubiquitously deployed, but they each provide different types of security. PGBGP would provide more protection than SBGP or soBGP if partially deployed, and the incentive for early adopters provides a believable adoption path. If ubiquitously deployed, PGBGP would provide security comparable to soBGP with the addition of policy violations. Finally, PGBGP is the simplest system to adopt. Combination schemes are also possible. For instance, soBGP or SBGP could be used to offer cryptographic protection for signed updates, with PGBGP serving as the default for cryptographically unverifiable routes.

8. Limitations of PGBGP

Our proposed enhancements to PGBGP would provide a safer but not perfectly secure environment for the BGP network. In this section we describe all of the PGBGP vulnerabilities that we are aware of.

8.1. Insecure data plane

Like most BGP security mechanisms, PGBGP only protects the routing control messages (control plane), and does not verify that the traffic actually traverses the announced route (data plane). Hu et al. have begun studying data plane route verification [28,47] by measuring destination characteristics such as the destination host OS, IP identifier probing, and TCP timestamps. Such techniques could be used to reduce the number of false positives in PGBGP.

8.2. Corrupted data

PGBGP implicitly relies upon attentive operators to monitor alerts from the IAR to prevent invalid data from entering PGBGP normal databases. All operators may not exhibit this level of vigilance, and their networks will be less safe. We showed in Section 6 showed that there are very few alerts to any individual operator and the alerts are trivial to receive. If the adversarial AS were contacted during the depreference period but failed to correct the problem, it would remain up to the adversary's providers and the operational community to prevent the bogus routes from propagating.

8.3. Adversary location

Under some circumstances, an anomalous route could spread unhindered by PGBGP, although alerts would still be distributed. For instance, if the adversary were the victim's sole provider, then the victim's routes would not be able to propagate. However, ASes with many connections are less susceptible to this vulnerability. In future work we intend to explore this area further.

8.4. Mixed relationships

If two ASes have both a customer–provider and a provider–customer relationship, PGBGP could miss a policy violation involving that edge. For instance, in North America AS A might be AS B's provider, but in Europe AS A could be B's customer. Both directed edges (A,B) and (B,A) could regularly be seen by other ASes, that are not customers of A and B. PGBGP would be unable to detect policy violations involving those edges. Generally such a relationship mixture is rare, customer–provider and peer–peer mixtures are more common and PGBGP can detect policy violations that include them.

8.5. Potential DoS

PGBGP is vulnerable to denial-of-service attacks. For example, an adversary could introduce many new edges or (prefix, origin AS) pairs with false route updates that the normal database would have to keep track of. As shown in Section 4.4, the amount of history data required for each edge or pair is small, so such an attack would have to be significant (and noticeable due to all of the anomalies). This might be remedied by discarding route updates with excessively long AS paths and limiting the rate of updates for each prefix.

9. Conclusions

The Internet's IP routing infrastructure has a number of critical security vulnerabilities, which arise in large part because it is a complex network. It is comprised of self-interested, often competitive, autonomous nodes; it is dynamic; and there is incentive for malicious behavior. Any network with these properties is likely to suffer similar vulnerabilities as the BGP network and to face similar difficulties in designing and encouraging the adoption of appropriate defenses. The basic approach outlined in this paper (adopt new routes cautiously) is potentially applicable to other settings.

In the case of BGP, existing cryptographic solutions have not been deployed because they require a public key infrastructure, community consensus, and changes to the BGP protocol. Although anomaly detection schemes are easier to deploy, they have traditionally been unable to offer the same level of protection. In this paper, we showed that simple anomaly detection, coupled with an automated response, can offer protection comparable to that provided by the cryptographic solutions.

We showed through simulation that pretty good BGP could largely eliminate the effects (reaching only 0.07–2% of all ASes depending on the type of attack) of origin AS and invalid path attacks if deployed on the largest 0.5% of ASes. We also showed that PGBGP is nearly as effective at

stopping attacks as an idealized security solution. Finally, we showed that PGBGP is incrementally deployable because it does not require global cooperation or changes to the BGP protocol. In addition, PGBGP has low overhead, generating an average of only 0.02 alerts per day per AS, and could be readily included in a routing software upgrade.

Acknowledgements

The authors thank Deepak Kapur and Yannis Avramopoulos for their help with the manuscript. JK and SF gratefully acknowledge the support of the National Science Foundation (Grants CCF-0621900 and CCR-0331580) and the Santa Fe Institute. JR gratefully acknowledges the support of HSARPA (Grant 1756303).

References

[1] S. Murphy, BGP Security Vulnerabilities Analysis, RFC 4272, January 2006.
[2] R. Mahajan, D. Wetherall, T. Anderson, Understanding BGP misconfiguration, in: Proceedings of ACM SIGCOMM, 2002, pp. 3–16.
[3] Renesys Blog, Pakistan Hijacks YouTube, <http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml>.
[4] J. Karlin, S. Forrest, J. Rexford, Pretty good BGP: Improving BGP by cautiously adopting routes, in: Proceedings of IEEE International Conference on Network Protocols, November 2006.
[5] P. Boothe, J. Hiebert, R. Bush, How prevalent is prefix hijacking on the Internet? NANOG 36 <http://www.nanog.org/mtg-0602/boothe.html>, February 2006.
[6] S.A. Misel, Wow, AS7007! <http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html>, April 1997.
[7] Internet Alert Registry forums, <http://cs.unm.edu/karlinjf/IAR/phpBB2/viewtopic.php?t=30>, November 2006.
[8] Renesys Blog, Con-Ed Steals the'Net, <http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml>.
[9] V.J. Bono, 7007 explanation and apology, <http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html>, April 1997.
[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] W. Leibzon, Question on 7.0.0.0/8, <http://www.merit.edu/mail.archives/nanog/msg05883.html>, April 2007.
[14] H. Chan, D. Dash, A. Perrig, H. Zhang, Modeling adoptability of secure BGP protocol, in: SIGCOMM New York, NY, USA, ACM, 2006, pp. 279–290.
[15] J. Karlin, S. Forrest, J. Rexford, Protecting BGP from Invalid Paths, University of New Mexico TR-CS-2007-12, 2007.
[16] Y. Rekhter, T. Li, S. Hares, A border gateway protocol 4 (BGP-4), RFC 4271 (Draft Standard), Jan. 2006. Available: <http://www.ietf.org/rfc/rfc4271.txt>.
[17] L. Gao, On inferring autonomous system relationships in the Internet, IEEE/ACM Transactions on Networking 9 (6) (2001).
[18] H. Ballani, P. Francis, X. Zhang, A study of prefix hijacking and interception in the Internet, in: SIGCOMM, New York, NY, USA, ACM, 2007, pp. 265–276.
[19] A. Ramachandran, N. Feamster, Understanding the network-level behavior of spammers, in: Proceedings of ACM SIGCOMM, New York, NY, USA, 2006, pp. 291–302.
[20] C. Kruegel, D. Mutz, W. Robertson, FredrikValeur, Topology-based detection of anomalous BGP messages, in: Proceedings of Symposium on Recent Advances in Intrusion Detection, vol. 2820, September 2003, pp. 17–35.
[21] S. Kent, C. Lynn, K. Seo, Secure border gateway protocol, IEEE Journal on Selected Areas in Communications 18 (4) (2000) 582–592.
[22] J. Ng, Extensions to BGP to support secure origin BGP (soBGP), expired internet draft draft-ng-sobgp-bgp-extensions-02, April 2004.
[23] T. Wan, E. Kranakis, P. van Oorschot, Pretty secure BGP, psBGP, in: Proceedings of Network and Distributed System Security, 2005.
[24] Y.-C. Hu, D. McGrew, A. Perrig, B. Weis, D. Wendlandt, (r)Evolutionary bootstrapping of a global PKI for securing BGP, in: Hot Topics in Networks Workshop, November 2006.
[25] L. Subramanian, V. Roth, I. Stoica, S. Shenker, R. Katz, Listen and whisper: security mechanisms for BGP, in: Proceedings of Networked Systems Design and Implementation, March 2004.
[26] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S.F. Wu, L. Zhang, Detection of invalid routing announcement in the internet, in: Proceedings of Dependable Systems and Networks, 2002.
[27] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, L. Zhang, Protecting BGP routes to top level DNS servers, IEEE Transactions on Parallel and Distributed Systems 14 (9) (2003) 851–860.
[28] X. Hu, Z.M. Mao, Accurate real-time identification of IP prefix hijacking, in: Proceedings of IEEE Security and Privacy, 2007.
[29] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, A. Rubin, Working around BGP: An incremental approach to improving security and accuracy of interdomain routing, in: Proceedings of Network and Distributed Systems Security, February 2003.
[30] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S.F. Wu, L. Zhang, An analysis of BGP multiple origin AS (MOAS) conflicts, in: Proceedings of Internet Measurement Workshop, November 2001.
[31] J. Qiu, L. Gao, S. Ranjan, A. Nucci, Detecting bogus BGP route information: going beyond prefix hijacking, SecureComm, 2007.
[32] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S.F. Wu, L. Zhang, Observation and analysis of BGP behavior under stress, in: Proceedings of 2nd ACM SIGCOMM Workshop on Internet Measurement, 2002, pp. 183–195.
[33] J. Rexford, J. Wang, Z. Xiao, Y. Zhang, BGP routing stability of popular destinations, in: Proceedings of Internet Measurement Workshop, 2002.
[34] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the internet topology, Computer Communication Review 29 (1999) 251–262.
[35] P. Holme, J. Karlin, S. Forrest, Radial structure of the internet, Proceedings of the Royal Society A 463 (2007) 1231–1246.
[36] Internet Alert Registry, <http://cs.unm.edu/karlinjf/IAR/>.
[37] RouteViews, <http://www.routeviews.org/>.
[38] Renesys routing intelligence, <http://www.renesys.com/products_services/routing_intelligence/>.
[39] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, L. Zhang, PHAS: A prefix hijack alert system, in: Proceedings of USENIX Security Symposium, 2006.
[40] North American Network Operators Group, <http://www.nanog.org>.
[41] African Network Operators Group, <http://www.afnog.org>.
[42] Australian Network Operators Group, <http://www.ausnog.net>.
[43] Japan Network Operators Group, <http://www.janog.gr.jp>.
[44] Pacific Network Operators Group, <http://www.pacnog.org>.
[45] J. Karlin, S. Forrest, J. Rexford, BGP simulator (BSIM), <http://cs.unm.edu/karlinjf/pgbgp/>.
[46] The CAIDA AS relationships dataset, <http://www.caida.org/data/active/as-relationships/>, February 2007.
[47] C. Zheng, L. Ji, D. Pei, J. Wang, P. Francis, A light-weight distributed scheme for detecting ip prefix hijacks in real-time, in: Proceedings of ACM SIGCOMM, 2007, pp. 277–288.

Josh Karlin is a PhD student in the Computer Science Department at the University of New Mexico in Albuquerque. He is a member of the Adaptive Computation Laboratory under the advisement of Dr. Stephanie Forrest. His research interests include adaptive systems, computer networks, and computer security.

Stephanie Forrest is Professor and Chairman of the Computer Science Department at the University of New Mexico in Albuquerque. She is a Senior Member of IEEE, a member of the NSF Network Science and Engineering Council, and an External Professor of the Santa Fe Institute. Her research area is adaptive systems, including genetic algorithms, computational immunology, biological modeling, and computer security. She received the PhD in Computer and Communication Sciences from the University of Michigan (1985). Before joining UNM in 1990 she worked for Teknowledge Inc. and was a Director's Fellow at the Center for Nonlinear Studies, Los Alamos National Laboratory.

Jennifer Rexford is a Professor in the Computer Science department at Princeton University. From 1996–2004, she was a member of the Network Management and Performance department at AT&T Labs – Research. She is co-author of the book "Web Protocols and Practice" (Addison-Wesley, May 2001). She served as the chair of ACM SIGCOMM from 2003 to 2007, and currently serves on the CRA Board of Directors, the ACM Council, and the GENI Science Council. She received her BSE degree in electrical engineering from Princeton University in 1991, and her MSE and PhD degrees in computer science and electrical engineering from the University of Michigan in 1993 and 1996, respectively. She was the winner of ACM's Grace Murray Hopper Award for outstanding young computer professional of the year for 2004.