Pretty Good BGP: Protecting BGP by Cautiously Selecting Routes

Josh Karlin
University of New Mexico

Stephanie Forrest
University of New Mexico
Santa Fe Institute

Jennifer Rexford
Princeton University

Abstract

The Border Gateway Protocol (BGP), the Internet's interdomain routing protocol, is vulnerable to a number of damaging attacks. Proposed solutions either (i) rely on a public-key infrastructure and accurate routing registries or (ii) detect attacks only after they have spread throughout the network. However, BGP routers could avoid selecting and propagating malicious routes if they were cautious about adopting new reachability information. We describe an enhancement to BGP, Pretty Good BGP (PGBGP), that slows the dissemination of malicious routes, providing network operators time to respond before the problem escalates into a large-scale Internet attack. Results show that realistic deployments of PGBGP could provide 99% of Autonomous Systems with 24 hours to investigate and repair malicious routes without affecting prefix reachability. The results also show that without PGBGP, 40% of ASs cannot avoid using malicious 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.

1 Introduction

The Border Gateway Protocol [1] has been the Internet's de-facto interdomain routing protocol for the last decade. It is a trusting and therefore vulnerable protocol that does not ensure the validity of the route announcements passed between BGP-speaking routers. Malicious networks (Autonomous Systems) can exploit BGP by announcing false routes in order to reroute traffic to an incorrect destination. For instance, on May 7th 2005, an AS falsely claimed to originate Google's prefix, which contained IP addresses for www.google.com [2]. For roughly one hour parts of the Internet could not reach Google's search engine as traffic was misdirected to the attacking AS. This is just one example of the kind of disruption that such attacks can cause. Rather than discarding packets, the adversary could snoop the contents of the packets or direct them to a different Web server to return alternative content or steal sensitive user information, such as financial data or passwords.

Some solutions have been proposed to increase BGP's security, for example, refs. [3, 4, 5, 6, 7]. These rely on global routing information maintained by a central authority. The authority would authenticate the AS that originates the BGP route for a destination prefix. It would also ensure that the AS-path attribute in the advertised route is a feasible path on the AS-level topology. However, ASs have been reluctant to reveal their business relationships, and existing registries, such as ARIN, RIPE, and APNIC [8, 9, 10], are incomplete and often inaccurate [11].

A second category of proposals relies on anomaly detection [12, 13, 14] to identify attacks early in their propagation and limit damage. This promising approach does not require changing the BGP protocol and can be deployed incrementally. However, to to be effective, an anomaly detector must be coupled with an effective response. Except for Whisper [13], which requires ubiquitous deployment to detect inconsistent routes, the BGP anomaly detectors do not actively stop the progression of attacks, simply alerting a human operator who may not be able to respond quickly enough (e.g., to prevent identity theft).

In this setting, false negatives are more problematic than false positives, because there are often multiple routes available to any destination. An AS that selects a malicious route places its customers in jeopardy, even if for a short period of time (a false negative). However, if a suspicious route is erroneously discarded in favor of a trusted route (a false positive), little damage results in the short run. Thus, we advocate avoiding suspicious routes when credible alternatives exist until a secondary process can confirm their authenticity. This approach would prevent a malicious route from harming any of the AS's cus-

tomers. If the route is not identified as malicious within some period of time, the suspected route could be released back into the network, causing no long-term loss of reachability. This represents a fairly conservative approach to anomaly detection, which seems appropriate given the vulnerabilities of BGP and the rather weak defense mechanisms available in practice.

In this paper we present Pretty Good BGP, a system that responds to BGP attacks by delaying their propagation. We illustrate PGBGP's effectiveness by studying its behavior on two of the most dangerous BGP exploits: prefix hijacks and sub-prefix hijacks. PGBGP is the first BGP security proposal to address the sub-prefix hijack problem. Because no protocol changes would be necessary to implement PGBGP, it is incrementally deployable via software updates and provides protective benefits for each AS that adopts it, even without widespread deployment in the rest of the Internet.

Our simulations show that on average over 97% of ASs can be temporarily protected from prefix hijack attempts, even if PGBGP is deployed on only the 62 most highly connected ASs (only 0.3% of all ASs). For the same deployment, on average over 85% of ASs can be protected from sub-prefix hijack attempts. If deployed on an additional set of randomly selected ASs across the network, PGBGP can prevent over 99% of the network from using hijack routes. An illegitimate route could be fixed within the time that it is suppressed, and the vast majority of the network would be unharmed. We show that without PGBGP, an average of nearly 50% of the network immediately reroutes to a malicious AS, and only 60% of the ASs are able to route around the attack once it has been detected. Finally, the potential impact of false positives is shown to be minimal, as only 0.1% of BGP announcements are anomalous.

The remainder of the paper is organized as follows. Section 2 discusses the challenges of detecting malicious BGP routes, and Section 3 describes how PGBGP addresses these challenge. In Section 4, we describe a simulator for evaluating PGBGP. Section 5 reports simulation results that assess PGBGP's effectiveness under various deployment scenarios. Section 6 discusses the implementation overhead of PGBGP and options for incremental deployment. Section 7 reviews related work, and Section 8 presents our conclusions and directions for future research.

2 Challenges of Detecting BGP Attacks

In this section, we briefly review the BGP protocol and discuss some of its vulnerabilities, to set the stage for PGBGP. We then discuss the use of anomaly detection for detecting BGP attacks, focusing on the use of BGP update messages.

2.1 Border Gateway Protocol (BGP)

Internet routing operates at the level of IP address blocks, or *prefixes*. Regional Internet Registries (RIR), 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, and growing, with prefixes appearing and disappearing over time.

ASs exchange information about how to reach destination prefixes using the Border Gateway Protocol (BGP). A BGP-speaking 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. 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.

If a router learns multiple routes for a 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]. Over the years, additional steps have been added to the decision process to give operators greater flexibility and control over their networks. Generally, a router prefers routes that conform to the policies of the local network operator. Next, the router prefers routes with the shortest AS path. 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 [15, 16]. 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. 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.

2.2 BGP Vulnerabilities

Many BGP vulnerabilities arise from the lack of reliable information about prefix ownership and the ease with which malicious parties can introduce BGP announcements for prefixes they do not own. In a prefix hijack, an adversary configures a router to announce a destination prefix that it does not own. The idea being that traffic destined for the legitimate AS will be diverted to the attacking AS. The adversary can drop the hijacked traffic, causing a denial-of-service known as a *black hole*. For example, on December 24, 2004, a time in which many operators were on holiday, thousands of prefixes were hijacked by AS 9121 (TTnet) [17], leading to widespread disruptions in Internet connectivity.

Instead of dropping the traffic, the adversary can snoop the packets before directing them to the legitimate host. In the worst case, the adversary could impersonate the services of the legitimate host, such as a government or financial Web site, to publish misinformation or steal sensitive user data. Even a short-lived attack can inflict significant damage, such as identity theft from a large number of users. In fact, short-lived attacks are an effective way to avoid arousing the suspicion of users and operators. Short-lived prefix hijacks also arise due to configuration mistakes, where a network operator inadvertently configures a router to announce the wrong prefix (e.g., due to a typographical error).

2.2.1 Prefix Hijacks are Hard to Prevent and Detect

Prefix hijacking is surprisingly difficult to prevent. Ideally, every AS would apply filters to the routes learned from neighboring ASs, to discard BGP routes for unexpected prefixes. Although an AS directly connected to the adversary could easily filter announcements for unexpected prefixes, best common practices for route filtering are not deployed ubiquitously. But, even vigilant ASs cannot easily apply filters to BGP routes that originate several AS hops away, because the AS would not know what origin AS to expect for each prefix. Also, the overhead of applying large prefix-filtering rules can overwhelm today's routers [17], forcing operators to make difficult trade-offs between security and robustness when configuring route filters. Ultimately, even security-conscious operators cannot completely protect their ASs.

Prefix hijacking is sometimes difficult to detect, too. Ideally, a prefix would have a single origin AS for its entire lifetime, making a route announcement with a different origin AS a clear indication of an attack. However, prefixes may change ownership. For example, some companies and universities prefer to have their upstream provider announce their prefixes into BGP on their behalf. If the institution switches providers, a new AS would start announcing the prefix. In addition, a small fraction of prefixes have more than one legitimate originating AS [18]. For example, an institution might have multiple upstream providers that each announce the prefix into BGP. Thus, not all new origins for a prefix necessarily imply a prefix-hijack attempt.

2.2.2 Sub-prefix Hijacks are Especially Difficult

In a conventional prefix-hijacking attack, some ASs direct traffic toward the adversary while others continue to forward packets to the legitimate destination. However, a small modification can make the attack even 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 malicious 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—a sub-prefix announced by the adversary.

Ideally, route filtering would 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 the conservative approach of 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 blackholing legitimate traffic. Even if they could be detected, sub-prefix hijacks

are hard to avoid, once detected. 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 malicious sub-prefix. These ASs would essentially *deflect* the packets onto a path toward the adversary anyway.

Finally, not all new sub-prefixes are triggered by malicious attacks or configuration errors. Prefixes are often legitimately subdivided into smaller blocks when one AS delegates address space to another. In addition, a legitimate AS might start advertising sub-prefixes of a larger address block to exert fine-grain control over the incoming traffic (e.g., for effective load balancing over multiple incoming links). A sub-prefix might also be announced when a customer connects to a new provider. For example, consider a customer that owns a small portion of its provider's address block. If the customer has no other providers, other ASs can reach the destinations through the provider's larger address block, obviating the need to announce the more-specific prefix. However, if the customer decides to enlist a second provider, both providers need to start announcing the sub-prefix to ensure that the customer receives traffic through both connections. Hence, sub-prefix announcements sometimes have legitimate causes, even when they seem suspicious.

2.3 Challenges of BGP Anomaly Detection

The previous subsection showed that it is difficult to determine which announcements are legitimate. That is, the problem of classifying a route announcement as legitimate or malicious is to some extent ambiguous. Consequently, we must rely on methods that can evaluate announcements in the context of the network's history and current state. One way to do this is with anomaly detection, in which the normal behavior of a process is characterized by a model, and deviations from the model are called anomalies.

In behavior-based anomaly-detection systems, examples of normal behavior are presented to the system in a training phase and a model of normal behavior is constructed from these examples. In some cases, examples of known attacks (labeled data) are also presented during training to simplify the learning problem. However, in many situations, the space of possible attacks is not understood well enough to use this simplification. Formally, the anomaly-detection problem can be viewed as a one-class online learning problem in non-stationary environments. The learning is "one class" if the system is presented only with examples of normal behavior during training; it is "online" if the learning must occur while the system is operating and making routing decisions, and it is "non-stationary" if the learned concepts can change through time. For BGP, all three of these conditions hold, complicating the detection problem.

Over time, the detector needs to incorporate new information, so that it is not making decisions based solely on old data. This is because over time prefixes change ownership and location, prefixes are subdivided, and previously unallocated prefixes are announced—the nonstationary environment. Without incorporating new data, the detector would have fewer and fewer legitimate routes available to it. The anomaly detector also needs to eliminate old routes if they are no longer active. This consideration addresses scalability as well as security. Preserving a long history of old routes is potentially memory intensive, and in the event that a hijacked route is erroneously accepted (a false negative), the system needs some mechanism of recovery.

A final complication is that unlabeled attack data may occur in the training data. In the BGP domain, this arises because some of the announcements used during training may in fact be attacks.

We incorporated these considerations into a simple learning and response rule for PGBGP—accept all new routes after they have survived an initial probationary period. In addition, routes that have not shown recent activity are removed from the history. We define recent activity as a route that appears in an update message or resides in a router's table within a window of time that we call the *history period*. PGBGP is an anomaly detector in that it treats the window of recently active routes as *normal* and everything else as anomalous. PGBGP learns new behavior by incorporating new routes into the normal definition after a probationary period, called the *suspicious period*. As most bad routes do not persist for very long, PGBGP can tolerate attacks in the continuous stream of training data, as they will usually disappear before being incorporated into the definition of normal. Finally, PGBGP implicitly responds to anomalies by avoiding the suspicious routes.

3 Pretty Good BGP (PGBGP)

The basic idea of PGBGP is to delay the adoption of new routes for forwarding data traffic. We argue that PGBGP allows time for a secondary process to check if the new route announcement is valid, while protecting the network in the meantime. Although PGBGP is applicable to various kinds of BGP attacks, we initially focus on how to identify and prevent prefix and sub-prefix hijacks by delaying their propagation. We also discuss the effects of false positives (i.e., when a valid announcement is mistakenly classified as suspicious) on the network.

3.1 Identifying Hijack Attempts

PGBGP detects prefix hijacks by monitoring the origin ASs in BGP announcements for each prefix over time. If an announcement has an origin AS that has not been seen recently for said prefix, PGBGP treats the route as anomalous. We define "recent" as any origin that has been announced or resided in the router's BGP table for that prefix within the last h days where h is history period of the anomaly detector. Sub-prefix hijacks are discovered by monitoring new prefixes. New prefixes that are completely contained by recently seen prefixes are potentially sub-prefix hijack attempts, and PGBGP treats them as anomalous. The router continues to treat the origin AS as anomalous for this prefix for a period of s days, where s is the length of the suspicious period of the anomaly detector. If the route is known to be good, the router can be reconfigured sooner to treat the route as normal.

When an anomalous route is discovered, the router generates an alarm, triggering a secondary process to check the validity of the new route. This could be either network operators or an automated network-management system. In some cases, the neighboring AS may advertise a different route or withdraw the anomalous route after a brief period of time, obviating the need for an explicit response. This could happen if the new route were caused by a configuration error or a short-lived attack. By delaying adoption of the route, an AS would protect its customers from the short-lived problem, even if other ASs adopted it. This provides a tangible benefit for early adopters of PGBGP and an incentives for other ASs, lest they lose customers to the early adopters.

In other cases, the anomalous route may persist, requiring further investigation. Groups of ASs who trust each other might work together, out of band from BGP, to determine whether the new route appears suspicious, perhaps using an overlay service like the one proposed in [19]. Trust relationships can be based on personal or business relationships, or prior knowledge that an AS follows best common practices for securing its infrastructure, applies protective route filters, or runs PGBGP. As part of the investigation, a network-management system could launch active probes to learn how data traffic would be handled by neighbors announcing the suspicious route. These probes might reveal unexpectedly long propagation delays or unusual hops in the IP-level path, confirming a problem. The probes could also send application-level messages to the end hosts via the suspicious path, allowing comparisons to the content available via the other paths.

3.2 Avoiding Prefix Hijacks

A PGBGP-enabled router avoids selecting anomalous routes whenever possible. If the router has alternative routes for the prefix, the router selects the best of the already known routes. False positives, while possible, only favor the less-preferred route temporarily. If no alternative route exists, the router could either select the suspicious route (in the hope of avoiding a black hole) or not select any route (rather than direct data traffic to a risky route). The first approach favors the known route until the very last step in the decision process, whereas the second approach simply disregards all suspicious routes. This choice is a policy decision that should be left to network operators, although we envision the first alternative being preferred in order to avoid losing reachability when false positives occur.

As an example of how PGBGP would respond to a real attack, consider the previously mentioned attack on Google. On May 7th of 2005, at 14:37:56 UTC, the prefix 64.233.161.0/24 (which at the time contained IP addresses for www.google.com) was originated by AS 174 for one hour. To understand the impact, we analyzed the RouteViews BGP update feeds [20] provided by Equinix, a large service provider, during that period. From the routes announced by the Equinix peers, three ASs (2914, 6730, and 13645) chose the route to the hijacking AS 174. The other three (3257, 16559, and 1239) chose the route to the legitimate AS. An AS running PGBGP could have avoided using the malicious route, as AS 174 was a new origin for the prefix and the other had been stable for months. So long as at least one of Equinix's peers provided a route to the legitimate origin, PGBGP would have chosen correctly. Because the attack lasted for one hour, even a relatively small suspicious period would have thwarted the attack.

3.3 Avoiding Sub-Prefix Hijacks

In the case of prefix hijacks, anomalous routes are avoided by selecting an alternative route. Thwarting a sub-prefix hijack is more complicated, however, because the router does not have any normal routes available for the sub-prefix. There are two possible approaches to this problem. One approach would be to avoid selecting any route for the sub-prefix by relying on the route for the larger address block to forward the data packets. This would allow the AS to continue forwarding packets normally, using the BGP route for the larger address block. This approach is flawed, however, because a downstream AS that chose a malicious route would deflect the data packets toward the adversary. Hence, we should expect that PGBGP would have to be widely deployed for this approach to be effective. Instead, a PGBGP-enabled

router could intentionally discard packets matching the sub-prefix (e.g., by installing a "null" route in the forwarding table), rather than risk having a downstream AS deflect data packets toward a malicious adversary. As before, the choice between these two options is a policy decision left to the network operator, although the first approach would prevent an honest AS from accidentally creating a denial-of-service.

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 previously unreachable, and thus, cannot be hijacking traffic destined to another, legitimate AS. PGBGP accepts the new announcement and installs the new prefix in the forwarding table. This scenario might arise when an AS announces a BGP route for a private address block, a reserved address block, or a block that has not yet been allocated by the Regional Internet Registries. In practice, the routers in a well-run AS would be configured with route filters that discard such "bogon" routes [21]. The BGP route announcement would then be discarded before the PGBGP rules could be applied. After an Internet registry allocates a new address block, the vigilant operator modifies the route filters to allow announcements for the new block. When the first announcement arrived, a PGBGP-enabled node would accept the new route and use it to forward packets[1].

Legitimate new sub-prefixes will sometime be treated as suspicious by PGBGP—a false positive. In this case, the router will forward packets based on the larger address block it used before the sub-prefix was announced. In many cases, two valid announcements (for the larger and smaller address blocks) would have the same origin AS or traverse the same downstream AS. This would occur, for instance, if a service provider delegated a portion of its address block to a customer AS. In this scenario, forwarding based on the larger address block would be completely appropriate and likely have no effect on the flow of traffic. However, if the customer connects to multiple providers and is announcing the sub-prefix to control the flow of inbound traffic, the situation is more complicated. Here, the PGBGP-enabled router could be temporarily disregarding the wishes of the origin AS by sending data traffic along a different (albeit still valid) path. Once the sub-prefix announcement was deemed to be legitimate, traffic would flow as the origin AS intended.

The only situation that would compromise reachability arises if the origin AS switched providers, while still retaining the IP address block allocated by its old provider—a practice sometimes explicitly disallowed by the business agreement between customer and provider. In this case, forwarding packets based on the larger address block would be a mistake that could lead to a temporary black hole. In practice, when an AS switches providers, the AS typically connects to both providers during a transition period to avoid an abrupt loss of connectivity (e.g., if the old provider disconnects the customer before the new connection starts). The common practice of maintaining the old connection for a brief period would also give the PGBGP-enabled ASs time to learn about the new route and determine that it was valid.

In summary, PGBGP prevents sub-prefixes covered by other BGP prefixes from entering the decision process until the suspicious period has passed. Once this period has passed, all of the associated routes are accepted into the decision process, and traffic to valid sub-prefixes flows freely over the same paths that they would in BGP today.

3.4 Decision Process and Convergence

To maximize protection from malicious routes, an AS should always prefer safe (non-suspicious) routes, when available. That is, preference for non-suspicious routes should be the first step in the decision process, ahead of local preference and AS-path length. This introduces an interesting economic trade-off for the AS. Local preference is typically based on the business relationship with the neighboring AS, with the highest preference reserved for customer-learned routes and the lowest for provider-learned routes. Selecting a safe route learned from a provider over a new route learned from a customer goes against the AS's immediate economic incentive to gain revenue by directing as much traffic as possible through downstream customers. Some network operators, as a matter of policy, might prefer to keep local preference as the first step in the decision process, applying the PGBGP heuristic as a a second step.

Although the preference-first policy might be appealing financially, it could substantially reduce the effectiveness of PGBGP. For example, consider a scenario with ubiquitous deployment of PGBGP, but where every AS applies the PGBGP heuristic as the second step in the decision process. Then, the provider of the malicious AS would select the malicious route, unless the legitimate route was learned from one of its other customers. In turn, that AS's provider would pick the malicious route, unless the legitimate route was learned from one of its customers. As a result, large portions of the Internet might still direct traffic to the malicious AS. This scenario would also hamper PGBGP in avoiding sub-prefix hijacks. When using local preference as the first step, an AS would always select the suspicious route to a sub-prefix, rather than forwarding traffic based on a safe route for the larger address block.

In spite of the short-term financial benefit of a

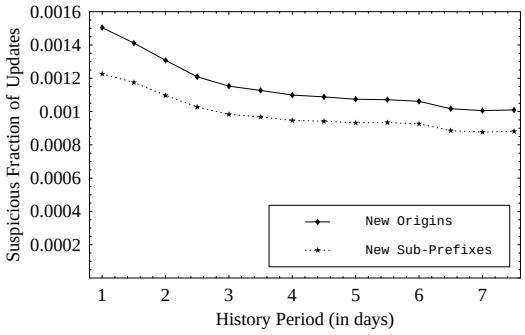

Figure 1: Fraction of announcements classified as suspicious using different history periods (h).

preference-first policy, it might make longer-term business sense to be cautious. First, the AS would not violate its normal preference rules very often or for very long. Only a small fraction of BGP routes would be classified as anomalous and for a short period of time. False positives could be handled even more quickly if the secondary process for validating the route were successful. Second, protection against malicious routes is a valuable security service for the AS's customers; customers might use security as a criteria for choosing an AS. Third, an AS would rarely view a route learned from its customer as anomalous. A well-run AS would have good information about valid prefixes for its own customers, and could apply route filters to discard routes for unexpected prefixes. In practice, we envision that anomalous routes would be acquired primarily from peers and providers.

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 [22], ASs typically select and export routes based on their business relationships. If every AS prefers customer-learned routes, BGP convergence can be provably guaranteed [15]. As long as local preference remains the first step in the decision process, the guidelines in [15] 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 [15] holds when ASs apply more conservative export policies than their business relationships normally suggest.

3.5 Tuning PGBGP's Timer Parameters

Two important parameters, the suspicious period (s) and the history period (h), affect the behavior of PGBGP. These parameters correspond to the time an anomalous route is avoided before being accepted (s) and the time that an origin AS 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 [11]. 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 anomalies following a brief outage. On the other hand, h should not not be longer than necessary for three reasons. First, the implementation overhead grows with the size of h, because the router must store information about the past (prefix, origin AS) pairs. Second, a long history period might allow a repeated prefix-hijack attack to become trusted, if an undetected malicious origin AS remained in the history buffer. Third, h will effectively set the initial training time for a router coming online unless it is bootstrapped with history information from other routers in the same AS.

To set h, we ran the PGBGP algorithm on RouteViews BGP update data from Equinix for the months of May through July of 2005 with a 24-hour suspicious period. The average fraction of incoming announcements that are labeled anomalous for either presenting a new origin or sub-prefix are displayed in Figure 1 for each evaluated history period. The figure suggests that a correlation between the rate of introduced sub-prefixes and new origins exists which is expected since any origin for a new prefix will be suspicious since it is also new. The figure shows only marginal reductions in the rate of anomalies after three days. We speculate that three days is enough time for network operators to fix equipment failures that cause a legitimate (prefix, origin AS) pair to disappear from view. Particularly for small ASs, such as universities or small companies, disruptions in connectivity might reasonably persist for a couple of days before repair. Thus, we selected three days as the value for the history period.

4 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 li-

cense [23], 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 5.

4.1 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 [24].

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 [25], which were inferred from BGP routing data collected primarily from RouteViews [20]. The topology has 18,943 ASs with an average of four AS-AS links each. The work in [25] 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.

4.2 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, anomalous routes are removed from consideration either before or after the local-preference step, depending upon the configuration of the simulator.

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 3.4 that the routing system should converge.

Our experiments determine which ASs would select a malicious route, and how PGBGP limits and delays the propagation of the route across the AS topology. Studying the propagation of the malicious 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 malicious route for s steps, we then evaluate what happens when these ASs stop suppressing the route. The 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.

4.3 Experimental Configuration

The simulator has several configurable parameters, as summarized in Table 1. 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 PG-

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

Table 1: Simulator parameters (and default values)

BGP 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. The simulator also has the option of removing anomalous routes from consideration either before or after the local-preference step in the BGP decision process.

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.

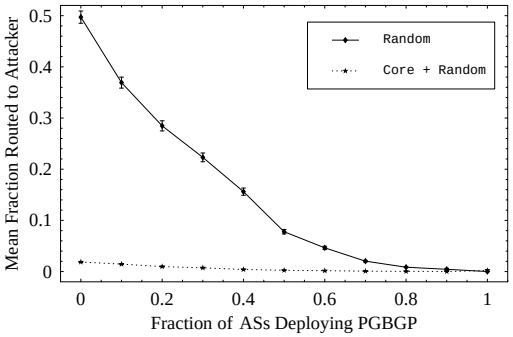

Figure 2: Both Deployments, Prefix Hijack, Day One

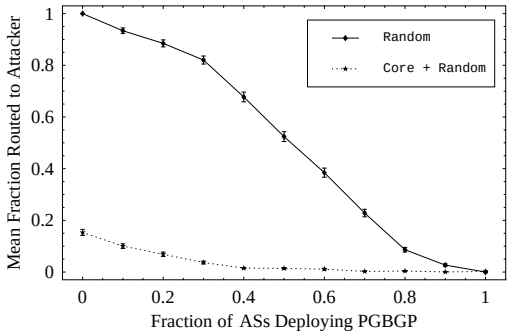

Figure 3: Both Deployments, Sub-Prefix Hijack, Day One

5 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 wider-scale deployment of PGBGP. Next, we illustrate that PGBGP's automated response helps ensure ASs learn a viable alternative to the malicious route. Then, we demonstrate that false positives will self-correct over time; all legitimate routes eventually propagate throughout the network. Last, we show that PGBGP is most effective if the decision process eliminates anomalous routes in the first step. The section ends with a summary and discussion of future directions.

5.1 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 malicious 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 malicious 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 malicious route. For example, the adversary's single-homed customers would learn only the malicious 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 of-

fers 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 malicious 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.

5.2 Stopping Sub-Prefix Hijacks

The results for sub-prefix hijacks are similar, although a larger PGBGP deployment fraction 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 malicious 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 malicious 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.

5.3 Importance of a Collective Response

In addition to avoiding malicious route, 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 malicious 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 malicious AS, as shown earlier in the top curve of Figure 2. Do most of these ASs have an alternate route that uses the legitimate AS, should they independently realize that the other AS is malicious?

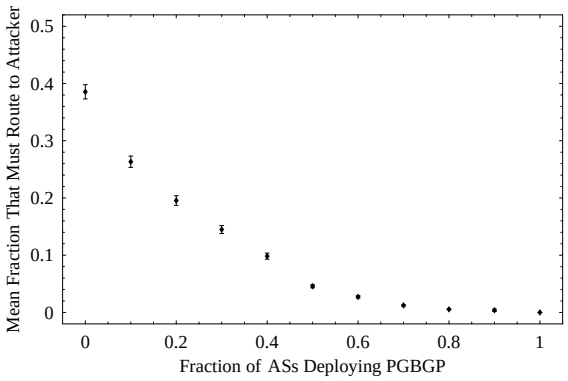

Figure 4: Random Deployment, Prefix Hijack, Cannot Avoid

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 malicious 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.

5.4 Attack Propagation

For the simulation parameters, network operators have a 24-hour period to detect and resolve attacks before the routers automatically accept the anomalous routes as normal. If a malicious 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 quickly these routes propagate, we can understand both how quickly an undetected malicious route spreads and how quickly a false positive corrects itself.

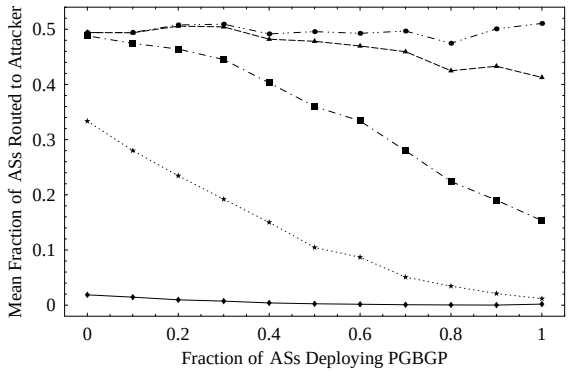

Figure 5: Core + Random Deployment, Prefix Hijack, 5 Days

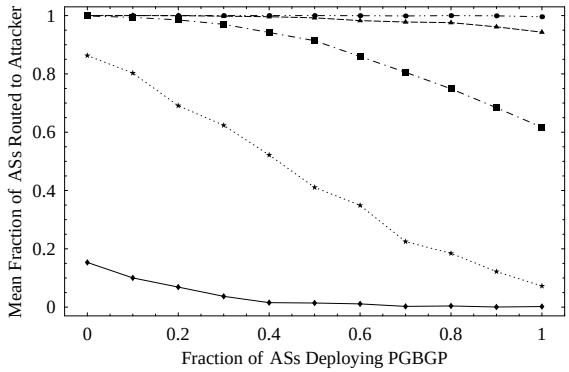

Figure 6: Core + Random Deployment, Sub-Prefix Hijack, 5 Days

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 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 malicious routes (real attacks) and self-correcting for false positives (legitimate new routes). As the figures show, we hamper the spread of new attacks and accommodate the introduction of legitimate 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 relying on the automatic timeout to release the route.

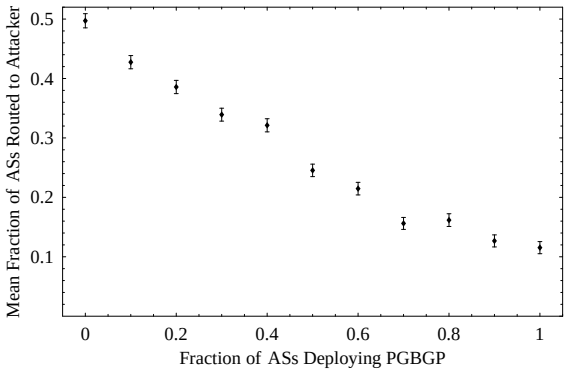

Figure 7: Random Deployment, Prefix Hijack, Operator Preference First

5.5 Prioritizing Local Preference

Section 3.4 discussed what might happen if ASs applied their local preference rules as the first step of the decision process. Figure 7 illustrates the negative consequences of this policy for the random deployment scenario under prefix hijacks. When none of the ASs run PGBGP, half of the ASs pick a malicious route, consistent with the top curve in Figure 2. However, as an increasing fraction of nodes adopt PGBGP, its benefits are sharply reduced compared to Figure 2. In fact, with local preference as the first step in the decision process, an average of 10% of ASs would pick a malicious route *even with ubiquitous deployment of PGBGP*. As discussed earlier (Section 3.4), the adversary's provider would pick the malicious route unless it had a legitimate route from one of its other customers. In turn, this AS's customers and providers would likely pick the malicious route as well.

It is worth noting that the change in ordering in the decision process does not affect PGBGP's ability to avoid sub-prefix hijacks. For sub-prefix hijacks, the malicious route corresponds to a unique prefix, so the comparison based on local preference does not eliminate any legitimate routes from consideration.

5.6 Summary and Discussion

Our experiments show that PGBGP is effective at protecting the network from prefix and sub-prefix hijack attacks, especially when the small number of core ASs

run the enhanced protocol. With PGBGP deployed in the 62 core ASs and 30% of the remaining ASs, around 99% of the ASs can avoid prefix attacks and 95% can avoid sub-prefix hijacks, compared to 50% and 0% respectively with conventional BGP. In addition to avoiding malicious routes, a PGBGP-enabled AS also helps ensure that other ASs (including its customers) learn at least one legitimate route. As time progresses, an anomalous route is allowed to propagate through the network, unless the route disappears on its own or a secondary process verifies that the route is malicious; because of the small diameter of the Internet, a new route would finish propagating within $5 \times s$, that is, within five days using our parameters.

For all the experiments, we randomly selected the malicious AS. This might be a reasonable assumption for prefix hijacks caused by unintentional configuration mistakes, or for adversaries who must locate a vulnerable AS (e.g., one that does not use route filtering) to launch a hijacking attack. However, some 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 arise 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. We also plan to study the effectiveness of PGBGP in conjunction with selective route filtering. We hope to show that combining route filtering with PGBGP would enable a well-run AS to protect itself, despite the presence of other ASs that are not as careful.

Although hijacking attacks are among the most serious threats, they are not the only way for an adversary to introduce false information into BGP. In future work, we plan to evaluate PGBGP's ability to block other kinds of attacks. For example, an adversary might add or remove AS hops in the AS-path attribute to make a route look more or less attractive. If AS A could reach AS D by the path (A,B,C,D) but instead announced (A,B,D) it would have falsely made its route more attractive to its neighbors. Such an attack could be recognized by keeping track of all recently seen routes for each prefix and treating all routes with new AS-path subsequences as anomalous. Preliminary results show that 15% of announced routes contain new AS-paths when the history period is set to 3 days. We also plan to study the effectiveness of ASs cooperating to determine the legitimacy of a route.

6 Implementation and Deployment

Implementing PGBGP involves adding a single step to the decision process running in software on the routers. No changes to the BGP protocol, message format, or the underlying router hardware are required, and one AS could deploy PGBGP when others have not. This significantly lowers the barrier to deploying PGBGP in practice.

PGBGP would be a small extension to the BGP software running on today's routers. First, the router would need to maintain a history buffer that stores recently seen (prefix, origin AS) pairs that no longer appear in active routes. When a BGP update message withdraws or overwrites the last active route with this origin AS, the router would record the (prefix, origin AS) pair along with a timestamp in a hash table. A background process could delete entries from the hash table once the history period has expired. The history buffer would not need to store any other BGP route attributes, such as the AS path or community values. In contrast, the BGP routing table (already stored on the router) must store complete information for each active route from each neighboring AS. In addition, we expect that the router would not have to store history information for many inactive (prefix, origin AS) pairs because legitimate prefixes rarely have serious outages or change originating ASs in practice. Thus, we expect the history buffer to be an extremely small addition to the storage requirements on the router.

When a route announcement is received, the router would determine if the last hop in the AS path matches the origin AS of any existing (normal) route for that prefix, or matches an entry in the history buffer. If not, the new route would be marked as anomalous, and a timestamp stored with the entry in the routing table. A background process could update the classification after s time periods. Finally, implementing PGBGP would require an additional step in the BGP decision process that compares routes based on their classification. This simple binary comparison could be applied either before, or after, the determination of local-preference values, depending on the router configuration. This extra step should run quickly, compared to the remaining dozen or so steps in the BGP decision process.

Changes to the decision process require vendor adoption of PGBGP, through an update of the software running on their routers. Alternatively, an AS could move the entire responsibility for BGP path selection to a separate software platform, as advocated in [26]. The Routing Control Platform (RCP) would receive BGP update messages from neighboring ASs and select a single best route for each BGP prefix on behalf of each router. The RCP uses the existing BGP protocol to send the best route for each prefix to each router, without requiring

any changes to the software on the legacy routers. The RCP could implement a new decision process such as that of PGBGP. By seeing BGP routes announced by all neighbors, the RCP would be in a good position to identify anomalous routes and store historical data. The RCP would also be a natural place to implement extensions to PGBGP that allow trusted ASs to cooperate in detecting malicious routes.

7 Related Work

Many proposed BGP security solutions, such as sBGP [4] and soBGP [5], depend on central authorities to maintain an accurate registry of prefix ownership and to provide keys and signatures. However, such registries have remained elusive. Alternative solutions, such as Whisper [13] and MOAS lists [12] (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. Although the MOAS list monitor alerts the operator only upon detection of a malicious route, Whisper prevents suspected routes from being used. However, Whisper's "penalty-based route selection" policy only circumvents ASs that are suspicious for multiple prefixes, and the solution relies on ubiquitous deployment.

Kruegel *et al.* [14] 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 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 malicious routes.

The way that PGBGP responds to new routing information is similar to route-flap damping [27] and age-based tie-breaking [1]. First, route-flap damping temporarily excludes unstable routes from the BGP decision process. However, route-flap damping suppresses routes for an unstable (prefix, neighbor) pair, rather than considering the attributes of the route (such as the origin AS). Second, age-based tie-breaking is a step later in the BGP decision process on some routers. When two routes are equally good, age-based tie-breaking prefers an older route over a recent one. Age-based tie-breaking only considers when the routes were learned, not the past history or the route attributes. As with route-flap damping, the goal is to improve stability, rather than security.

PGBGP has some similarities to rate limiting mechanisms that have been proposed for other security problems. Virus throttling [28], for example, throttled back abnormally high rates of outgoing connection attempts to ensure that Internet viruses propagated slowly. Slowing the propagation of a malicious route is similar to slowing the propagation of viruses, although our mechanism is quite different.

The PGBGP design differs from these earlier systems, however, in that it does not actually delay packet delivery. PGBGP could also be viewed as a form of temporary quarantine [29], in which new routes are temporarily quarantined from the rest of the network.

8 Conclusions

BGP is vulnerable to malicious attacks and configuration errors because the contents of route announcements cannot be easily verified. This paper introduced an incrementally deployable modification to the BGP decision process, called PGBGP, 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. We implemented this simple heuristic by imposing a 24-hour period during which new routes are given lower priority in the decision process. By avoiding new routes, 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. Our results 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 malicious sub-prefix routes when deployed only on the 62 core ASs in our study network. If PGBGP were deployed on all ASs, protection would be greater than 99% in both cases. 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 showed that delaying the acceptance of new routes is a safe and effective method for slowing the propagation of malicious routes to a human time scale. An important feature of our method is that false positives self-correct within five days, so that legitimate changes in the network are automatically incorporated. A second feature of our approach is that 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 PG-

BGP, as it provides immediate benefit even if other ASs have not deployed it. Finally, PGBGP is highly effective, even if only the core ASs adopt it.

9 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. ANIR-9986555, CCR-0331580, CCR-0311686 and EIA-0324845 and the Defense Advanced Research Projects Agency Grant No. F30602-02-1-0146. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

[1] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol (BGP-4)," *Internet Draft (work in progress). Available at http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-26.txt*, October 2004.

[2] T. Wan, "Analysis of bgp prefix origins during googles 7-may-2005 outage," *Manuscript, Available from Authors*.

[3] 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.

[4] 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.

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

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

[7] S. Murphy, O. Gudmundsson, R. Mundy, and B. Wellington, "Retrofitting security into Internet infrastructure protocols," in *Proc. DARPA Information Survivability Conference and Exposition*, vol. 01, pp. 3–17, 1999.

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

[9] RIPE. http://www.ripe.net/.

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

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

[12] 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.

[13] 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.

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

[15] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. on Networking*, vol. 9, pp. 681–692, December 2001.

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

[17] T. Underwood, "The anatomy of a leak: AS9121," *NANOG*, May 2005.

[18] 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 Workshops*, Nov. 2001.

[19] H. Yu, J. Rexford, and E. Felten, "A distributed reputation approach to cooperative Internet routing protection," in *Proc. Workshop on Secure Network Protocols*, November 2005.

[20] RouteViews. http://www.routeviews.org/.

[21] N. Feamster, J. Jung, and H. Balakrishnan, "An empirical study of 'bogon' route advertisements," *ACM SIGCOMM Computer Communications Review*, January 2005.

[22] T. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. on Networking*, vol. 10, pp. 232–243, April 2002.

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

[24] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Trans. on Networking*, vol. 9, December 2001.

[25] X. Dimitropoulos, D. Krioukov, M. Fomenkova, B. Huffaker, K. Claffy, and G. Riley, "Inferring AS relationships: Peering links resurrected." In submission.

[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] C. Villamizar, R. Chandra, and R. Govindan, "BGP route flap damping," 1998. RFC 2439.

[28] M. M. Williamson, "Throttling viruses: Restricting propagation to defeat malicious mobile code," in *Proc. ACSAC Security Conference*, 2002.

[29] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet quarantine: Requirements for containing self-propagating code," in *INFOCOM*, pp. 285–294, April 2003.

Notes

[1] An adversary could conceivably track new RIR address allocations, in the hope of announcing a route for the prefix *before* the owning AS does. Yet, the adversary would not be able to attract any real traffic to these IP addresses without also subverting some other system, such as the Domain Name System, to cause end hosts to send packets to these new addresses.