Pretty Good BGP: Protecting BGP by Cautiously Selecting Routes

Paper 49, 14 pages

Abstract

The Internet's interdomain routing protocol, BGP, is vulnerable to a number of damaging attacks primarily due to operator misconfiguration. Proposed solutions with strong guarantees require a public-key infrastructure, accurate routing registries, and changes to BGP. Until such a large proposal is adopted, networks will remain vulnerable to false information injected into BGP. However, BGP routers could avoid selecting and propagating these routes if they were cautious about adopting new reachability information. We describe a protocol-preserving enhancement to BGP, Pretty Good BGP (PGBGP), that slows the dissemination of disruptive routes, providing network operators time to respond before the problem escalates into a large-scale Internet attack. Simulation results show that realistic deployments of PGBGP could provide 99% of Autonomous Systems with 24 hours to investigate and repair misconfigured routes without affecting prefix reachability. We also show that without PGBGP, 40% of ASs cannot avoid selecting disruptive 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 (BGP) [1] has been the Internet's de-facto interdomain routing protocol for the last decade. During this time several exploits have been discovered and documented [2]. Most of these vulnerabilities can be avoided by adopting good administrative practices such as authenticating peering connections with neighbors and giving routing-protocol traffic the highest priority. Unfortunately, the protocol's most troublesome weakness is also the hardest to resolve: BGP does not have any means to verify the routing information—by default, routers trust the contents of all routing messages.

Malicious Autonomous Systems (ASs) can exploit this property of BGP by announcing false (bogus) routes in order to reroute traffic to an incorrect destination. Generally such bogus announcements are the result of typographical errors or improper filtering, not malicious agents. For example, a simple typographical error can cause a human operator to enter the wrong IP address block (prefix), causing an AS to originate routing information for a prefix it does not own.

Such simple mistakes have caused damage for nearly the last 10 years. A classic example is the incident in 1997 where a small ISP (AS 7007) originated the first class-C subnet of every IP prefix [3, 4]. This created reachability problems for every network, and it simultaneously crashed routers around the world due to the increase in prefix state information. Administrative practices have since improved, but even today's well-managed ASs, such as Verio (AS 2914), cannot always protect themselves. For example, on January 22, 2006, Con Edison (AS 25706) originated many prefixes it did not own, causing outages for several networks such as Panix (AS 2033) [5]. Verio accepted these false routes and passed them on to others due to stale information in its routing registry.

Several solutions have been proposed to increase BGP's security [6, 7, 8, 9, 10]; some, such as sBGP, soBGP, and psBGP, even offer strong guarantees. However, these proposals require full, or at least very large scale, deployment and further cooperation from each AS; sBGP and soBGP require global routing information to be maintained by a central authority. The authority would authenticate the AS that originates the BGP route for a prefix. The authority 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 [11, 12, 13], are incomplete and out of date [14], making them unlikely to provide a solid underpinning for a secure interdomain routing protocol for the Internet.

Strong security guarantees are ideal for such critical infrastructure. However, nearly ten years have elapsed without significant progress in operational deployment. Until a fully secure solution is readily deployable, alternatives must be sought to keep the infrastructure robust against routing disruptions. In response, a second category of proposals has appeared that rely on anomaly detection or out-of-band services [15, 16, 17, 18] to identify attacks early in their propagation. This promising approach can be deployed incrementally since it does not require changing the BGP protocol. However, to be effective, an anomaly detector must be coupled with an effective response. Except for Whisper [16], which requires ubiquitous deployment to detect inconsistent routes, the BGP anomaly detectors do not actively stop the progression of attacks. Instead, they simply alert a human operator who may not be able to respond quickly enough (e.g., to prevent identity theft or router overload).

In this paper we present Pretty Good BGP, a system that responds to BGP misconfigurations and some classes of ma-

licious attacks by delaying their propagation. In contrast to previous work on anomaly detection, PGBGP's *automated response* to suspicious BGP announcements prevents the propagation of bogus routing information. In PGBGP, routers identify suspicious routes by consulting a table of trusted routing information learned from the recent history of BGP update messages. We evaluate PGBGP's effectiveness by studying its behavior on two of the most common BGP exploits—prefix hijacks and sub-prefix hijacks—using a sliding history window to construct a list of trusted (prefix, origin AS) pairs from the BGP update stream. PGBGP is the first BGP security proposal to address the sub-prefix hijack problem. Because our design does not require any changes to the BGP protocol, PGBGP is incrementally deployable via software updates.

PGBGP would confer significant benefits to early adopters, even without widespread deployment. Our simulations show that on average over 97% of ASs could be temporarily protected from prefix-hijack attempts, even if PGBGP were to be deployed on only the 62 most highly connected ASs (only 0.3% of all ASs) in the core of the Internet. If deployed on an additional set of randomly selected ASs across the network, PGBGP could prevent over 99% of the networks from using hijacked routes. An illegitimate route could be fixed within the time that it is suppressed, and then the vast majority of the network would be unharmed. We show that without PGBGP, an average of nearly 50% of the ASs would immediately reroute to a malicious AS, and only 60% of the ASs would be able to route around it once the malicious route is detected. Finally, the potential impact of false positives is shown to be minimal, as only 0.1% of BGP announcements are anomalous.

In the remainder of the paper, we discuss the challenges of detecting malicious BGP routes (Section 2) and present PGBGP (Section 3). 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 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 (RIRs), such as ARIN, RIPE, and APNIC, allocate IP prefixes to institutions such as Internet Service Providers. These institutions may, in turn, subdivide the address blocks and delegate these smaller blocks to other ASs, such as their customers. Ideally, the RIRs would be notified when changes occur, such as an AS delegating portions of its address space to other institutions, two institutions combining their address space after a merger or acquisition, or an institution splitting its address space after a company break-up. However, the registries are notoriously out-of-date and incomplete. Ultimately, BGP update messages and the BGP routing tables themselves are the best indicator of the active prefixes and the ASs responsible for them. BGP tables today contain around 170,000 active prefixes, 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 router learns how to reach external destination prefixes via BGP sessions with routers in neighboring ASs. BGP has two kinds of update messages—announcements and withdrawals. Upon receiving an announcement for a destination prefix, the router overwrites the old route (if any) from the neighbor with the new information. Announcements contain information such as the destination prefix, the announcer's IP address, and the AS path the route will take. As the route announcement propagates, each AS adds its own unique AS number to the AS path. The router responds to a withdrawal message by deleting the previously announced route from its routing table and propagating the withdrawal to its neighbors. BGP routing changes can occur for many reasons, such as equipment failures, software crashes, policy changes, or malicious attacks. Inferring the cause directly from the BGP update messages is a fundamentally difficult, if not impossible, problem.

A router with multiple neighbors would likely learn multiple routes for each prefix. A single "best" route is chosen by applying the BGP *decision process*. The decision process is a non-standard sequence of about a dozen rules that compare one route to another [1]. 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 [19, 20]. The most common relationships are customer-provider and peer-peer. In a customer-provider relationship, the provider ensures that its customer can communicate with the rest of the Internet

by exporting its best route for each prefix, and by exporting the customer's prefixes to other neighboring ASs. In contrast, the customer does not propagate routes learned from one provider to another as it pays for transit to its providers. In a peer-peer relationship, two ASs connect solely to transfer traffic between their respective customers. An AS announces only the routes learned from its customers to its peers. These business relationships drive local preferences, which in turn influence the decision process. Typically, an AS prefers customer-learned routes over peer-learned routes, and peer-learned routes over provider-learned routes.

2.2 BGP Vulnerabilities

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

2.2.1 Prefix Hijacks

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

Prefix hijacking can also be difficult to detect. Ideally, a prefix would have a single origin AS for its entire lifetime, causing a route announcement with a different origin AS to be clear indication of attack. However, prefixes may change ownership. For example, some companies and universities prefer to have their provider announce prefixes into BGP on their behalf. If the institution switches providers, a new AS would then start announcing the prefix. In addition, a small fraction of prefixes have more than one legitimate originating AS [21]. For example, an institution might have multiple providers that each announce the prefix into BGP. Thus, not all new origins for a prefix necessarily imply a prefix-hijack attempt.

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

2.2.2 Sub-prefix Hijacks

In a conventional prefix-hijacking attack, some ASs direct traffic toward the adversary while others continue to forward packets to the legitimate destination as the hijacked route is potentially one option among many. However, a small modification makes the attack more dangerous. When a data packet arrives on an incoming link, the router looks in its forwarding table for the entry with the longest matching prefix. By announcing more specific prefixes (*sub-prefixes*), the adversary can trick nearly every AS into using the 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—that is, the sub-prefix announced by the adversary.

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

Finally, not all new sub-prefixes are introduced 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 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 a single provider, 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 will announce the sub-prefix to ensure that the customer receives traffic through both con-

nections. 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 when announcements are legitimate. 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 (suspicious routes).

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.

Prefixes are non-stationary and consequently the detector needs to incorporate new information, so that it is not making decisions based solely on old data. 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—delay the adoption of suspicious routes. Suspicious routes are those that do not reflect ownership information learned from recent BGP update messages. PGBGP learns new behavior by incorporating suspicious routes into the normal definition after a probationary period, called the *suspicious period*. As many bad routes persist for a short time, [14] PGBGP's training data is mostly clean. Finally, PGBGP implicitly responds to anomalies by actively avoiding suspicious routes.

3 Pretty Good BGP (PGBGP)

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

3.1 Identifying Anomalous Routes

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

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

A router's RIB and history of updates are used to create a history of known origins for each prefix. This history is what PGBGP uses to define normal behavior. On initialization, there is no concept of normal, and therefore all incoming updates are accepted. This process continues for h days (the *history period*). After this initial training phase, new routes that would alter the state of normal behavior are quarantined if possible. The quarantine lasts for s days (the *suspicious period*), and after that time the update is accepted by PGBGP. This prevents short-term anomalous behavior from corrupting the definition of normal. Finally, stale data should be eliminated from the history. PGBGP removes known origins for a prefix if it has not appeared in the router's RIB in the last h, days. Likewise, if a prefix has not appeared in the router's RIB in the last h days, the entire prefix is removed from the history.

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

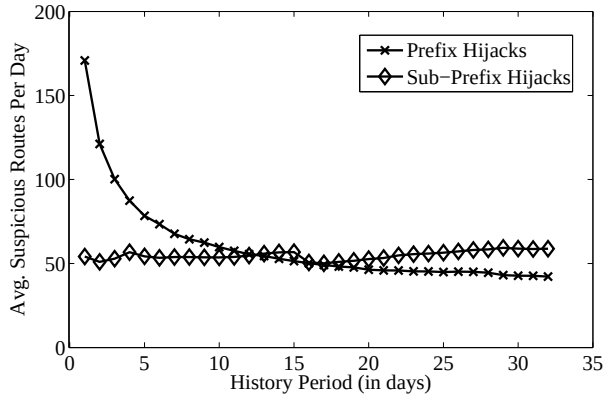

Figure 1: Average number of announcements (per day) classified as suspicious using a suspicious period of 1 day and a variety of history periods (h).

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

The *suspicious period* s and *history period* h are PGBGP's only parameters. They correspond to the time an anomalous route is avoided before being accepted (s) and the time that an origin is viewed as "recently seen" (h). Parameter s should be long enough for network operators to detect and resolve problems before they spread, but no longer than necessary. If s is too long, false positives will be slow to self-correct. A previous study of BGP misconfiguration showed that roughly 45% of new origins and prefixes exist for less than 24 hours [14]. These are temporary routes such as route leaks and hijack attempts. Because 24 hours is also a reasonable length of time for an operator to analyze and fix a routing problem, we use this value for s.

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

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

3.2 Avoiding Suspicious Routes

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

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

An interesting question is how the announcement of a new prefix that is not contained in a larger address block should be handled. In this case, the new announcement provides a

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

We have shown that it is possible to avoid suspicious routes. However, any modification to the decision process needs to consider the possible effects on BGP convergence. Although BGP is not guaranteed to converge for all combinations of routing policies [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 [19]. As long as local preference remains the first step in the decision process, the guidelines in [19] 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 [19] holds when ASs apply more conservative export policies than their business relationships normally suggest.

3.3 Example Routing Scenarios

In order to better explain the PGBGP algorithm and the effect it might have on the Internet, we detail the steps that PGBGP would take in important scenarios. The examples are categorized by the vulnerability that they exploit and in each subsection examples that describe how PGBGP would successfully prevent an attack (true positive) as well as how PGBGP would handle false positives are given.

3.3.1 Prefix Hijack Examples

True Positives: As an example of how PGBGP would respond to a prefix hijack, consider the previously mentioned Con Edison (AS 25706) attack on Panix (AS 2033). At the time of the attack, Panix's prefix (166.84.0.0/16) was correctly originated from AS 2033. On January 22 2006, AS 25706 mistakenly announced the same prefix causing many ASs (including Verio) to route to Con Edison instead of Panix, the rightful owner. When we ran our PGBGP simulator on RouteViews (Equinix Viewpoint) update data from November 2005 to February 2006, the attack along with several other prefixes Con Edison mistakenly announced were caught. In a real deployment PGBGP would then lower the preference of the suspicious routes so that trusted routes (if any) could be used. Short term misconfigurations are often caught within this time period by the operator that made the mistake. For those that are not, a secondary process could be employed to verify the authenticity of suspicious routes such as informing the victim AS that their address space may be under attack. In the Panix example, if the malicious route is not withdrawn before the delay expires Panix may resort to common counter measures, such as sub-prefix hijacking their own prefix (which PGBGP allows) in order to reclaim their space before the hijack spreads. [3]

False Positives: PGBGP does not affect reachability for prefixes that legitimately announce multiple ASs. Some ISPs host their customer's prefixes when the customer is not an AS itself. If such a customer has multiple providers, each must originate the customer's prefix. If a PGBGP AS only saw one such origin then the other might be lowered in preference when it appeared. Reachability would not be affected but the customer's attempts to load-balance its incoming data between the two origins might be ignored (temporarily). Note that this would not affect the use of backup providers as PGBGP only lowers the preference of suspicious routes and does not discard them. Likewise, PGBGP would not interfere with the process of changing providers since once the old provider's announcements were withdrawn the new origin would be selected.

Man-in-the-middle: The current design of PGBGP is not perfect, and an adversary could find ways to defeat it. For example, a man-in-the-middle attack could be accomplished by announcing very short routes to the legitimate origin that pass through her AS. PGBGP would not detect this event because it does not monitor suspicious edges along AS paths. As mentioned above, we decided to focus first on the most important and common cases of misconfiguration.

3.3.2 Sub-Prefix Hijack Examples

True Positives: In a sub-prefix hijack attempt, an AS would originate a prefix that is contained within another existing prefix in the routing table. If the route contained one of the larger prefix's origins, it would be considered safe because the traffic would be in the legitimate origin's control. For this example, the route is disruptive and therefore does not traverse one of the known origins for the super-prefix. PGBGP would consider such a route suspicious. It would refrain from entering the route into its routing table, though it would remember it for later use. Traffic would continue to flow based on the larger address block. If the route was found to be malicious the delay would have prevented an attack. Otherwise, after the delay period had passed, the saved routes would be entered into the routing table and used normally.

[3] By announcing sub-prefixes of their own address space, routers will select the more specific prefixes at packet forwarding time as opposed to the hijacker's route to the super-prefix.

False Positives: There are a few scenarios in which a sub-prefix might legitimately appear. Occasionally, an AS will announce sub-prefixes of its own blocks for fine-grained control. This does not result in a hijack and PGBGP would not interfere as the super-prefix originator is the same as the origin of the sub-prefixes. Similarly, PGBGP would not interfere if an AS announces sub-prefixes of its own prefixes in order to gain traffic back during a prefix hijack.

Different blocks with the same origin: 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 connected to multiple providers and announced the sub-prefix to control the flow of inbound traffic, the situation would be 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.

Changing providers and keeping the old provider's IP space: On rare occasions an origin AS will switch 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 if the old provider does not forward the traffic to the new. 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.

3.4 Decision Process

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 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.

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 license (citation removed for anonymity), 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 In-

ternet is a difficult problem. Much of the topology can be inferred from the BGP routing announcements themselves. For example, suppose that an AS A announces the paths (A,C,D,E) and (A,C,D,T,Y) for two different prefixes. These paths imply the existence of several edges in the AS-level topology, namely (A,C), (C,D), (D,E), (D,T), and (T,Y). The AS paths also provide a glimpse into the business relationships between ASs. For example, the path (A,C,D,E) implies that AS A is permitted to transit traffic through AS C to AS D. As such, we can infer that AS A and AS D cannot both be providers or peers of AS C. Each path implies a set of constraints on the relationships between ASs. By combining these constraints across a large number of paths, inference algorithms can classify the relationship between each pair of adjacent ASs as customer-provider or peer-peer [23].

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

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, suspicious routes are ranked lower than trusted routes 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.

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)

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 PGBGP on the 62 core nodes (i.e., the ASs with fifteen or more peers) and a random chosen subset of the remaining nodes, modeling a likely scenario in which a small number of large service providers deploy the enhanced protocol, along with a random set of other ASs. The simulator also has the option of ranking suspicious routes lower than trusted routes 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

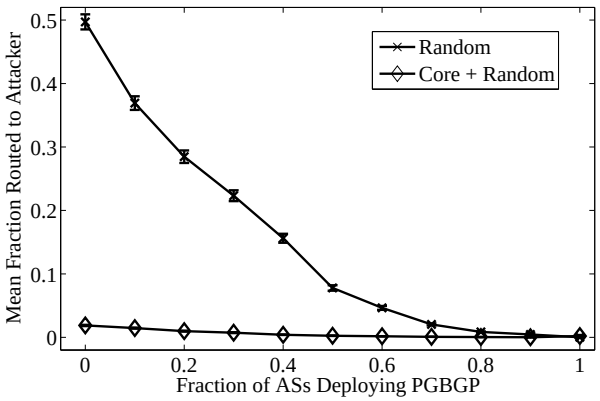

Figure 2: Both Deployments, Prefix Hijack, Day One

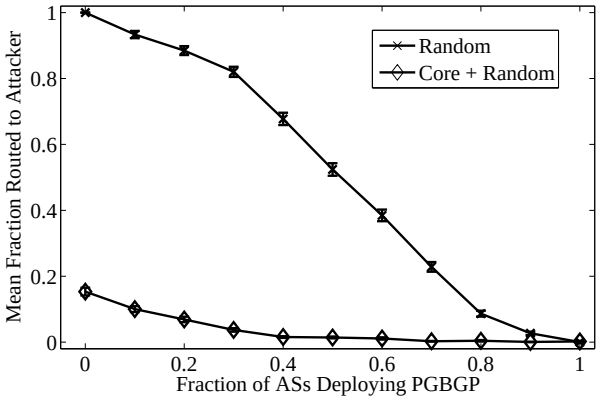

Figure 3: Both Deployments, Sub-Prefix Hijack, Day One

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.

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 larger-scale deployment. 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 selects trusted routes over suspicious 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 offers substantial benefits, as shown in the bottom curve in Figure 2. Running PGBGP just on these 62 ASs (and 0% of the remaining ASs) ensures that, on average, less than 2.5% of the ASs in the Internet select a route to the 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 wider PGBGP deployment is required to achieve the same gains, as shown in Figure 3. With zero deployment of PGBGP, which represents BGP today, every AS directs traffic to the 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

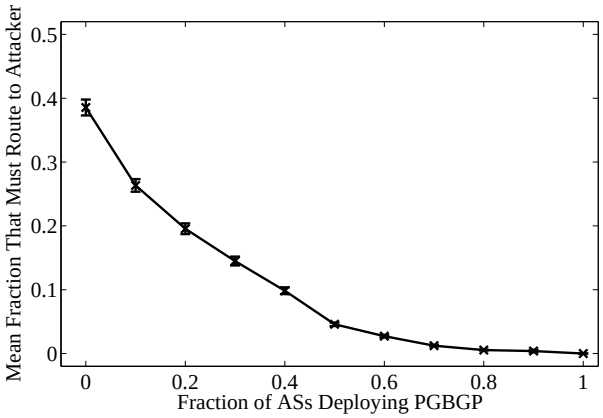

Figure 4: Random Deployment, Prefix Hijack, Cannot Avoid

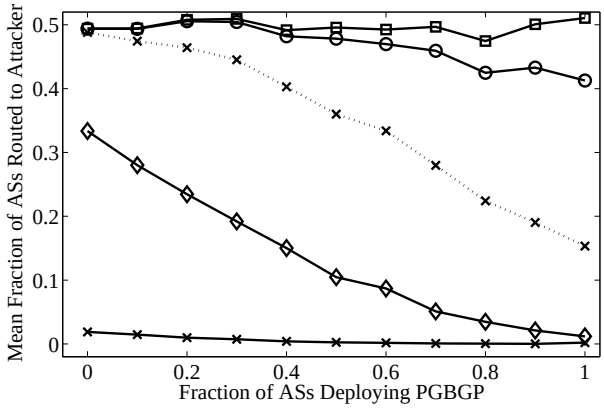

Figure 5: Core + Random Deployment, Prefix Hijack, 5 Days

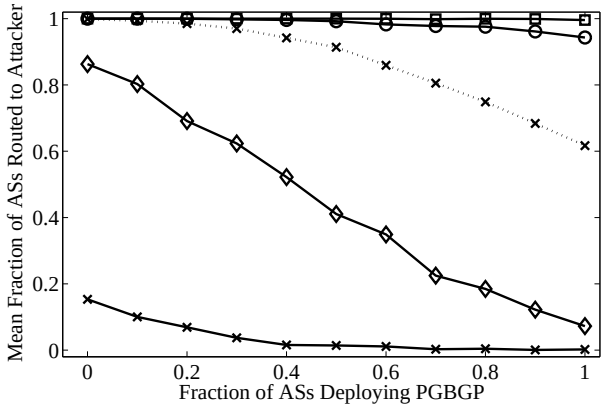

Figure 6: Core + Random Deployment, Sub-Prefix Hijack, 5 Days

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?

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.

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, corre-

sponding 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.

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, before considering whether a route is suspicious or not. 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%

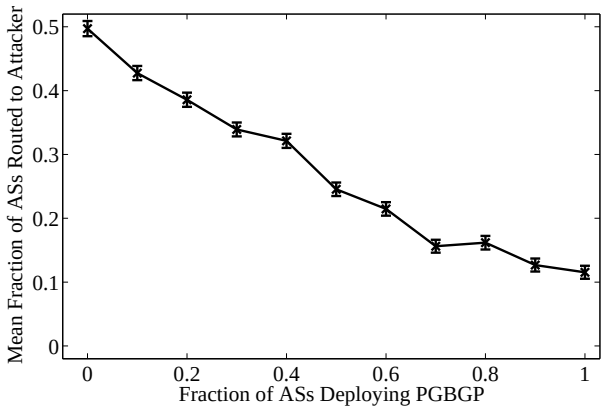

Figure 7: Random Deployment, Prefix Hijack, Operator Preference First

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. However, some intentional, malicious attacks would be difficult for PGBGP, or any other solution, to stop. For example, suppose the adversary controls an AS that lies on all paths to the legitimate origin AS—i.e., if the adversary is the provider for the legitimate origin AS. (Admittedly, such an attack seems unlikely because a provider would not have an incentive to disrupt reachability to its own customers, but this situation might happen due to an insider attack.) In future work, we plan to evaluate the effects of targeted attacks such as these, in which the adversary chooses the most damaging possible attack location. 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 perform a "man in the middle" attack by adding or removing 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 an-

nounced (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. Our 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 construct the history information necessary to determine the legitimacy of a route.

6 Implementation and Deployment

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

Implementing both functions on the routers: Implementing PGBGP on the routers requires extending how the routing software processes incoming BGP update messages. Upon receiving a BGP announcement, the router would need to compare the origin AS with the recently-seen ASs for this destination prefix to determine if the route is suspicious. Suspicious prefixes would be assigned a lower local-preference value, and suspicious sub-prefixes would be suppressed, to ensure that the router uses trusted routes where possible. In addition, the router would need to update the set of trusted (prefix, origin AS) pairs as new update messages arrive. This approach requires modifying the routing software but does not introduce any additional components into the network.

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

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

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

7 Related Work

Many proposed BGP security solutions, such as sBGP [6] and soBGP [7], 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 [16] and MOAS lists [15] (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.* [17] 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 [28] and age-based tie-breaking [1]. First, route-flap damping temporarily excludes unstable routes from the BGP decision process, whereas PGBGP simply lowers the ranking of suspicious routes. Second, route-flap damping operates at the level of (prefix, neighbor) pairs, rather than considering the attributes of the route (such as the origin AS). 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 [29], for example, throttles back abnormally high rates of outgoing connection attempts to ensure that Internet viruses propagate 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 in that it does not actually delay packet delivery. PGBGP could also be viewed as a form of temporary quarantine [30], in which suspicious routes are temporarily assigned a lower preference, to allow the router to select trusted routes when possible.

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 PGBGP, 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.

References

[1] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol (BGP-4)," *Internet Draft draft-ietf-idr-bgp4-26.txt*, October 2004.

[2] S. Murphy, "BGP security vulnerabilities analysis." RFC, January 2006. http://rfc4272.x42.com/.

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

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

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

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

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

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

[9] 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.

[10] T. Wan, E. Kranakis, and P. van Oorschot, "Pretty secure BGP, psBGP," *NDSS*, 2005.

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

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

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

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

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

[16] 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.

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

[18] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, "Working around BGP: An incremental approach to improving securityandaccuracyof interdomain routing," in *Proc. Network and Distributed Systems Security*, February 2003.

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

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

[21] 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.

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

[24] X. Dimitropoulos, D. Krioukov, M. Fomenkova, B. Huffaker, kc Claffy, and G. Riley, "AS relationships: Inference and validation." In submission.

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

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

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

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

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

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