Identifying and Addressing Protocol Manipulation Attacks in “Secure” BGP

Yang Song, Arun Venkataramani, Lixin Gao
University of Massachusetts, Amherst
ysong@ecs.umass.edu, arun@cs.umass.edu, lgao@ecs.umass.edu

***Abstract*—Researchers have studied a number of control and data plane attacks on BGP, the Internet’s interdomain routing protocol, in the presence of malicious ASes. These prior efforts have largely focused on attacks that can be addressed using traditional cryptographic mechanisms to ensure authentication or integrity (e.g., S-BGP). Although augmenting BGP with authentication and integrity mechanisms is critical, it is far from sufficient to prevent attacks based on manipulating the complex BGP protocol itself.**

In this paper, we identify two serious protocol manipulation attacks that undermine the two most fundamental goals of the BGP control plane—to ensure reachability and enable ASes to pick routes according to their policies—despite the presence of S-BGP-like mechanisms. Our key contributions are to (1) formalize two critical security properties, (2) experimentally validate using commodity router implementations that BGP fails to achieve them, (3) quantify the extent of the resulting vulnerabilities in the Internet’s AS topology, and (4) design and implement simple modifications to provably ensure that those properties are satisfied. Our experiments show that, a single malicious AS can cause thousands of other ASes to become disconnected from thousands of other ASes for arbitrarily long, while our proposed modifications almost completely eliminates such attacks.

I. Introduction

The Border Gateway Protocol (BGP), the Internet’s defacto interdomain routing protocol, is known to suffer from many security vulnerabilities due to the very nature of its assumptions of trust among independently operated networks or *autonomous systems* (ASes). BGP is a distributed *policy routing* protocol wherein routers exchange reachability information about destination prefixes, each router selects the most preferred route to each destination, and announces the selected route to its neighbors that in turn do the same. As deployed today, BGP utterly lacks the most minimal of security mechanisms, therefore malicious ASes can launch a variety of attacks [5], [18] that include *control plane* attacks such as prefix hijacking, spoofing, altering or faking routes, unauthorized prefix aggregation and de-aggregation, etc. as well as *data plane* attacks such as dropping, rerouting, or delaying packets.

Many comprehensive security measures have been proposed to limit misbehavior in BGP. These include measures to secure the control plane using cryptographic mechanisms to authenticate and verify the integrity of received routes and their implied prefix ownership (e.g., SBGP and a line of related work[10], [26], [11], [17]). Furthermore, measures to ensure data plane security have also been proposed, e.g., verification mechanisms to ensure that the paths selected in the control plane are consistently used in data plane for forwarding traffic [27], [9], [23]. On the surface, it may appear that combining control plane mechanisms for authentication and integrity of route announcements with data plane mechanisms to monitor and verify forwarding behavior is sufficient to eliminate the most potent of attacks in BGP.

Our primary contribution is to show that, despite mechanisms for route authentication and verification of forwarding behavior, BGP continues to be vulnerable to serious attacks based on manipulating the complex BGP protocol itself. We refer to these attacks as *protocol manipulation* attacks, a class of control plane attacks that can not (and were not intended to) be addressed using SBGP-like cryptographic mechanisms alone, yet can cause significant damage. For example, an off-path malicious AS can cause a victim AS to think that it has no usable route to a destination even though a route consisting only of good ASes to the destination is being announced to the victim. More disturbingly, it is possible for the malicious AS to block the destination from the victim *permanently*.

Our position is that manipulation attacks, attacks that can often be counterintuitive and difficult to anticipate, exist because we poorly understand the properties that the complex BGP protocol ought to satisfy in the presence of malicious ASes. To address this state of affairs, we formalize two desirable properties pertaining respectively to the two most fundamental goals of BGP, namely to ensure reachability and enable ASes to pick routes according to their routing policies. These properties can be informally summarized as follows. The first specifies that if a policy-compliant route to a destination consisting of only good ASes is available to an AS, the routing protocol should guarantee that the AS is able to eventually reach the destination. The second specifies that a malicious AS should not be able to force a good AS to pick a less preferred good route if multiple good routes are available.

Although these properties sound rather weak, we identify attack scenarios wherein even a single malicious AS or router suffices to cause BGP to violate both properties. In our attack scenarios, a malicious AS can take advantage of the fact that routers employ Routing Flap Damping (RFD) and/or Minimum Route Advertisement Interval (MRAI) timers. These timers are critical and are used to ensure the stability of the routing protocol and to reduce message overhead, but malicious ASes can abuse them in a manner that effectively makes a good route disappear from the victim’s routing table. Our attacks do not require all routers to deploy these mechanisms or even implement them in the same manner in order to be successful. Furthermore, simple modifications to these timer mechanisms do not suffice to eliminate the attacks identified in this paper. We analytically quantify the extent and experimentally demonstrate the feasibility of these attacks using different commodity router implementations.

A summary of our contributions is as follows.

1) We formalize two basic properties, *eventual reachability* and *policy prevalence*, desired of BGP's control plane (§III) and show that it fails to satisfy them in the presence of even a single malicious AS (§IV).
2) We demonstrate the feasibility of these attacks using commodity routers and derive sufficient conditions for attackers to commit these attacks under realistic Internet-like delays (§V).
3) We experimentally show using the Internet's AS topology that strategically positioned attackers can potentially disconnect thousands of ASes from thousands of other ASes (§VI).
4) We design and implement simple modifications to BGP that can provably ensure eventual reachability and policy prevalence (§VII).

We begin with an overview of BGP and related work.

II. Related Work and Background

A. Related work

As deployed today, BGP implicitly assumes that all ASes make truthful announcements. As a result, it is trivial for a malicious (or even a benignly misconfigured) router or AS to commit crippling attacks on both the control plane and data plane of BGP. For example, it can hijack the prefix, or spoof a route (control plane attacks), or drop, divert or delay forwarding traffic (data plane attacks). For a comprehensive description of BGP attacks, see [5].

A number of schemes have been proposed to secure BGP's control plane as well as data plane. The control plane defenses include origin authentication [17], which uses a trusted database to verify IP prefix ownership, secure Origin BGP (soBGP) [26], which provides both origin authentication and the verification of the physical existence of the announced path, and S-BGP [10], which not only verifies prefix ownership and existence, but also uses digital signatures to authenticate each BGP update message. For data plane security, data-plane verification and detection[5], [27], [9], [23] guarantee that a path that appears in a BGP announcement message is actually being used to forward traffic, and thus protect BGP from dropping and diverting attacks.

Our focus in this paper is on control plane attacks. Specifically, we ask the question: assuming SBGP-like authentication and integrity mechanisms, how vulnerable is BGP to control plane attacks? Somewhat disturbingly, we find that even a single malicious router or AS can launch control plane attacks violate rather basic reachability and policy-related properties expected of BGP. This is because existing proposals for control plane security, although critical for verifying the authenticity of announced routes compared to today's implicit-trust model, are not sufficient to protect against attacks that exploit the dynamics of the complex BGP protocol.

Our control plane attacks are complementary to and exist irrespective of the presence of data plane attacks or verification mechanisms to thwart them. This is because data plane security measures are designed to defend against malicious ASes on the forwarding path, however in our attacks, an *off-path* malicious AS can permanently disable a route consisting of only good ASes. A more detailed comparison of our control plane attacks to known data plane attacks appears at the end of §IV.

Our work is related to earlier studies in [28], [22], where the authors point out an attack example similar to one of our attacks. However, their attacks result only in temporary loss of connectivity. Further, attacks in [28] rely on a specific sequence of updates from other good ASes, and attacks in [22] can only disable the paths that go through the malicious AS. In comparison, the attacks identified in this paper (1) are more serious as they can cause *permanent* loss of connectivity and *persistent* violation of routing policy preferences; (2) can be performed unilaterally by a malicious AS; and (3) do not require the malicious AS to be on the disabled path(s). To our knowledge, this is the first paper to demonstrate attacks with all of the above three properties.

Our work is also related to another study [7], where the authors show that existing control plane security protocols are not enough to prevent inordinate "traffic-attraction", wherein a malicious AS can craft route announcements so as to attract more traffic. As acknowledged by the authors, it is unclear if traffic attraction by itself (in the absence of data plane misbehavior) constitutes an attack as any AS by design has the freedom to export any available routes (including provider-to-provider routes at a higher cost to itself). Furthermore, with S-BGP, traffic attraction alone can not cause unreachability. In contrast, our attacks can cause permanent unreachability and policy inversion without any data plane misbehavior.

B. BGP mechanisms to maintain stability

BGP is a decentralized routing protocol. Whenever a router's best route changes, it announces an update message to its neighbors. If the changes happen too frequently, they can cause unacceptable global message overhead. BGP implements two mechanisms to reduce the frequency of routing changes. The first mechanism is Route Flap Damping (RFD) that aims to suppress a route when it flaps often. The second mechanism is Minimum Route Advertisement Timer (MRAI) that ensures the time interval between consecutive updates to be large enough so as to prevent BGP's otherwise super-exponential message complexity. We explain these in detail next.

1) Routing Flap Damping (RFD): RFD is a mechanism designed to discourage the selection of unstable routes. According to RFC 2439 [24], each router maintains a penalty associated with every route announced by neighbors. The penalty measures the instability of a route. Whenever a route is withdrawn, the route's penalty is increased by a fixed value.

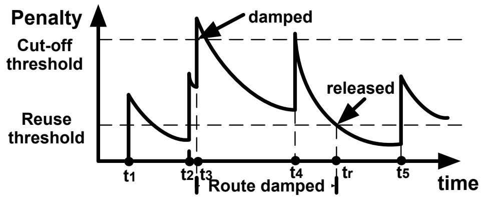


Fig. 1: An example illustrating how RFD works

If the penalty of the route exceeds the *cut-off threshold*, the route cannot be used for selecting the best route, i.e., the route gets *damped*. The penalty value decays exponentially over time according to

$$P_t = P_0 e^{\lambda(t-t_0)} \tag{1}$$

where P_0 and P_t are the penalty at time t_0 and t respectively, where $t_0 < t$, and λ is the decay parameter. Normally the amount of time needed for the penalty to decrease by half is used to indicate the decay speed, and the time is called the *half-life parameter*, represented by $\bar{t}$. Given $\bar{t}$, λ can be calculated using $e^{\lambda \bar{t}} = 0.5$. A route should be unsuppressed only if the penalty falls below the *reuse threshold*. To avoid over punishing a stable route, if a route has been *stable* for longer than the `max-suppress-time`, it will not be damped no matter how unstable it had been earlier.

We illustrate how RFD works in Figure 1. At time t_i, $i = 1, ..., 5$, the route gets flapped. When the penalty exceeds the cut-off threshold at t_3, the route is damped, and when the penalty decays to the reuse threshold at t_r, the route is reused.

The implementation of RFD in commodity routers is slightly different from that suggested in RFC 2439 (see Section V). We show that both the RFC-recommended and commodity router implementation expose routers to the RFD attacks. In addition, partial deployment of RFD [15] and the lack of uniform implementation standards can make the situation even worse.

2) Min Route Advertisement Interval (MRAI): MRAI is the minimum amount of time that must pass between consecutive announcements of a route [20]. It limits the frequency of route announcements sent to neighbors. In a recently published BGP4 protocol specification [21], withdrawals are also suggested to be limited by the MRAI timer. This suggestion departs from the previous BGP protocol specification [20] where withdrawals can be sent without waiting for the expiration of MRAI timer. We will show that, this change that has already been adopted by some commodity routers, exposes routers to MRAI attacks, and can cause permanent loss of connections.

III. Protocol vulnerabilities in BGP

A. Model and Assumptions

We focus on control plane security vulnerabilities in BGP in the presence of one or more compromised or *bad* ASes. Our threat model assumes that bad ASes can behave in a byzantine manner, i.e., they can deviate from the protocol in arbitrary, malicious ways. The rest of the ASes are by definition *good*, i.e., they strictly follow the protocol. Although bad ASes can behave in an arbitrary manner, we assume that they can not subvert standard cryptographic assumptions, i.e., they can not revert one-way hash functions or digitally sign messages on behalf of good ASes.

We further assume that the control plane is secured by authentication and integrity mechanisms such as those in SBGP [10]. As a result, ASes can only initiate route announcements for prefixes they own and prefix ownership is certified by a common certification authority. Furthermore, each route announcement (an AS path vector) carries with its proof that each AS along the route announced the corresponding prefix of the route. Likewise, an AS can verify if a route withdrawal was indeed issued by the immediately downstream AS. Good ASes only process verifiable updates and discard unverifiable updates immediately.

Our focus on control plane attacks means that data plane attacks (such as dropping, delaying, or incorrectly forwarding data packets) are outside the scope of this paper. Although data plane defenses are necessary for end-to-end security, they are in general not sufficient to obviate control plane defenses. Our position is that securing the control plane alone is an important intermediate goal, a position that is consistent with a long line of work in BGP security including SBGP [10][26][18] but is by no means a universally accepted position [25].

The restrictions on the behavior of malicious ASes as described above may naturally lead one to wonder what kind of egregious deviations from the protocol if any are possible. Essentially, a bad AS can announce or withdraw (verifiable) routes at whim, for example, even when no link or node failures or policy changes on part of other ASes occur. We show that even this restricted behavior on part of bad ASes can have serious consequences for good ASes.

B. Desirable properties

1) Definitions: A *route* or *path* is a sequence of distinct[1] ASes. A *good* route is a route consisting of only good ASes. A *bad* route is a route consisting of at least one bad AS.

The network is said to be in *steady-state* when (1) no further link or node failures occur and (2) good ASes do not make any further changes to their routing policies. Note that the latter condition allows a good AS to select a new, more preferred route received in a route announcement or switch to a less preferred route available in its route information base if its current route is withdrawn by the downstream neighbor, however it may not unilaterally decide to switch to a different available route in the absence of an announcement or withdrawal. It should be clear from the definition that in steady-state, only bad ASes can initiate routing events. Unless otherwise stated, all properties discussed in this paper assume that the network is in steady-state.

A *policy-compliant* route to a prefix is recursively defined as follows. A route adopted by a router is policy-compliant either if (1) the route is a single-hop route directly connecting the router to the destination, or (2) the next-hop (downstream)

[1] This definition ignores path-prepending [20], [21] for simplicity. Allowing it does not materially alter our findings.

router along the route has adopted a policy-compliant route to the destination and the next-hop router's policy makes it willing to route traffic destined for that prefix from the (upstream) router via its currently adopted route.

A prefix is said to be *reachable* at a router if the router has currently adopted a policy-compliant route to the prefix. Otherwise, the prefix is *unreachable*. Note that if we assume zero propagation delays and no failures, packets forwarded by a router along an adopted policy-compliant path are guaranteed to immediately arrive at the destination. So, although strictly speaking we focus only on control-plane properties in this paper, the definition of *reachable* does attempt to capture the common notion of forwarding plane reachability.

2) Eventual reachability: A fundamental goal of BGP is to enable reachability in a policy-compliant manner. We informally state a natural property that we expect BGP to satisfy: *If at least one policy-compliant, good route to a destination exists, the destination should be reachable.* In order to assess if or how well BGP satisfies this property, we need to state this property more formally. To this end, we introduce the notion of a *good AS-subgraph* below.

Let $\mathcal{A}$ denote the AS-level multigraph whose nodes are the set of all ASes and edges correspond to interconnections between pairs of adjacent ASes. Let $\mathcal{G}$ denote the subgraph of A obtained by removing all bad nodes as well as all edges adjacent to those bad nodes. We refer to $\mathcal{G}$ as the *good subgraph* of $\mathcal{A}$ or simply as the *good AS-subgraph*. We refer to $\mathcal{A}$ as the *original AS-graph*.

By definition, the routing policies of routers in the good AS-subgraph are identical to their corresponding policies in the original AS-graph except for policies involving bad routes (that are simply unavailable in the good AS-subgraph). Thus, for example, if r_1 and r_2 are two policy-compliant, good routes to a destination from a router X such that X prefers r_1 over r_2 in the original AS-graph, then X prefers r_1 over r_2 in the good AS-subgraph as well. Similarly export policies involving only good ASes are identical in the original AS-graph and the good AS-subgraph. For example, if an AS A chooses to not announce a route via one provider B to another provider C (because of the valley-free routing policy) in the original AS-graph and all three ASes A, B, and C are good, then A will not announce a route via B to C in the good AS-subgraph as well.

Property 1. (Eventual reachability) *If a destination is reachable from a router in steady-state in the good AS-subgraph, then the destination must be eventually reachable from the router in the original AS graph.*

Eventual reachability is a weak property since it only requires the destination to be reachable eventually, or equivalently not find the destination permanently unreachable. Thus, if there exists even an instant of time when the destination is reachable, the property is satisfied. We show that the current BGP can not even satisfy this weak property with malicious ASes.

3) Policy prevalence: BGP is designed to enable ASes to pick their most preferred route to a prefix when multiple choices are available. So, malicious ASes must not be able to force a router to consistently select a less-preferred path from a set of policy-compliant, good paths. Property 2 below captures this requirement.

Property 2. (Policy prevalence)*: If two or more policy-compliant, good routes to a destination always exist at a router in steady state, the destination must be reachable via a path that is at least as preferred as the most preferred of those routes.*

Although the above formal definitions may seem "obviously" true or like an overkill of formalism, we show that they are easily violated by attackers. The point of formalizing these properties is to convince ourselves that these are indeed security breaches, i.e., violations of intrinsically desirable properties, not just eccentric but normal BGP behavior.

IV. Attack Mechanisms

In this section, we show that BGP does not satisfy either of the properties introduced in the previous section. To this end, we present several simple example scenarios to violate the properties. All of the examples involve a single destination prefix to which all ASes attempt to establish a route, as routes to different prefixes are computed independently by BGP.

A. Attacks violating eventual reachability (ER)

We show two different examples of attacks that can violate ER below, one where the malicious AS abuses the RFD and another where it abuses the MRAI timer.

Example 1 Violating ER using RFD:

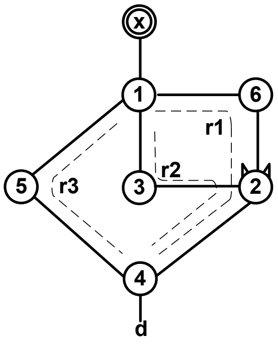


Fig. 2: Example 1: violating eventual reachability using RFD

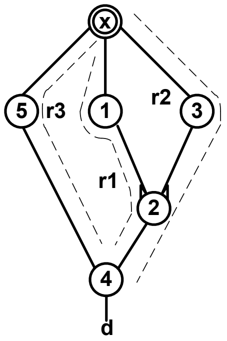


Fig. 3: Example 2: violating eventual reachability using MRAI timer

Example 1 shows how an attacker can abuse RFD to violate ER. Consider the topology shown in Figure 2. Node 1 has three paths to reach d, $r1$: $1-6-2-4-d$, $r2$: $1-3-2-4-d$, and $r3$: $1-5-4-d$. Node 1's preference order is $r1 > r2 > r3$. The malicious node 2 controls the two most preferred paths $r1$ and $r2$. $r3$ is good. We suppose RFD and MRAI use the settings recommended in RFC 2439 and RFC 1771 respectively. That is: (1) an RFD penalty is associated with each route, and (2) only withdrawals count towards the RFD penalty, and (3) MRAI is applied on announcements

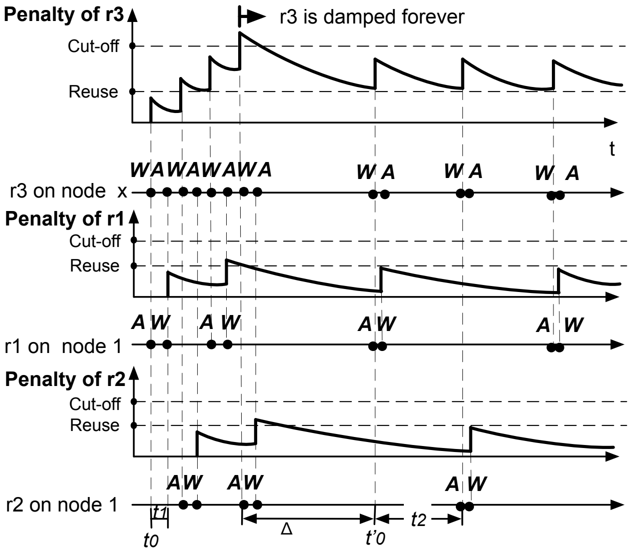


Fig. 4: $r3$'s penalty on node x in response to updates. A: announcement, W: withdrawal.

only, i.e., withdrawals are propagated immediately. We also assume that RFD is enabled on node 1 and node x with the same parameters. These assumptions are made for simplicity of exposition, and it is easy to show that the attack succeeds even without these assumptions.

The attack works by forcing node x to keep the good path damped forever. The attack is mounted in two stages. In the first stage, node 2 forces $r3$ to be flapped until it is damped by node x. In the second stage, node 2 forces $r3$'s penalty value to be consistently above the reuse threshold so that it remains damped forever. Stage 1 involves node 2 making the following sequence of announcements and withdrawals.

Stage 1:

- (Step 1) t_0: Announce $r1$ to node 6.
- (Step 2) $t_0 + t_1$: Withdraw $r1$ to node 6.
- (Step 3) $t_0 + 2t_1$: Announce $r2$ to node 3.
- (Step 4) $t_0 + 3t_1$: Withdraw $r2$ to node 3.

. . .

Repeat steps 1–4 until Step n when $r3$ is damped.

When node 2 announces r_1, node 1 withdraws r_3 and announces r_1 to node x. From x's perspective, r_1 and r_3 get one announcement and one withdrawal respectively. Similarly, when node 2 withdraws r_1, node 1 withdraws r_1 and announces r_3 instead. From node x's perspective, r_1 and r_3 get one withdrawal and one announcement respectively. Thus, by this point x gets one withdrawal for both r_1 and r_3. A similar sequence of flaps happens when node 2 announces and withdraws $r2$. Thus, after r_2 is withdrawn, x's withdrawal counter for r_3 has reached 2 while its counter for both r_1 and r_2 is 1. By construction, it should be clear that r_3's penalty grows at twice the rate of r_1 or r_2, so x will eventually damp r_3 but r_1 and r_2 will remain undamped. Figure 4 illustrates why the sequence of steps result in node x damping r_3.

Let Δ denote the length of time since x's penalty for r_3 exceeds the cut-off threshold until it decays back to the reuse threshold. During this period, node 2 does not announce either r_1 or r_2. Thus, x will find d unreachable for the duration of length Δ when r_3 remains damped.

Stage 1 above shows that an attacker can force a destination to become unreachable for time Δ. Next, we extend the attack to make the destination unreachable forever. This stage, referred to as stage 2, is shown below. Stage 2 begins at time t'_0 that denotes the first time when x's penalty decays to the reuse threshold. The parameter t_2 below refers to the time it takes for the penalty to decay back to the reuse threshold after one withdrawal has pushed it above the reuse threshold. The step index i is a nonnegative integer.

Stage 2:

- (Step 1) t'_0: Announce r_1 to node 6 and withdraw it immediately after.
- (Step 2) $t'_0 + t_2$: Announce r_2 to node 3 and withdraw it immediately after.

. . .

- $t'_0 + 2it_2$: Repeat step 1.
- $t'_0 + (2i + 1)t_2$: Repeat step 2.

. . .

The highest penalty $r3$ can reach during the stage 2 is one withdrawal penalty above the reuse threshold. Since normally at least two back to back withdrawals are needed for penalty to increase from the reuse threshold to the cut-off threshold, $r3$'s penalty is always below the cut-off threshold. Because the penalty of $r1$ and $r2$ are always smaller than the penalty on $r3$, $r1$ and $r2$ can never reach the cut-off threshold. Throughout the stage 2, $r3$ remains damped and neither $r1$ nor $r2$ is available to x, so the destination d remains unreachable to x forever, thereby violating ER. Figure 4 demonstrates the steps in Stage 2.

We note that the attacks described above are serious and can not be prevented with simple modifications to RFD or MRAI or changes to their parameters. A naive solution may be to allow an AS to reuse an unstable path if no other path exists. However, this solution allows the attacker to arbitrarily flap routes if it controls the only path, and can cause unacceptable global update flooding, contradicting the purpose of RFD.

Example 2 Violating eventual reachability using MRAI:

Consider the topology in Figure 3. Node x has three routes to reach d, $r1$: $1 - 2 - 4 - d$, $r2$: $3 - 2 - 4 - d$ and $r3$: $5 - 4 - d$. Node x prefers $r1$ and $r2$ to $r3$. By virtue of its position, the attacker node 2 controls $r1$ and $r2$. $r3$ is a good path. We assume that the MRAI timer is in use and is applied to both announcements and withdrawals as recommended in RFC 4271[21]. We also assume that the RFD timer is disabled on Node 1, 3 and x.

The attack involves the malicious node 2 making a sequence of announcements and withdrawals that result in d becoming permanently unavailable to x. The sequence of steps and the timing are listed below, M refers to one MRAI interval (typically 30 seconds [20], [21]).

- (Step 1) t_0: Announce r_1 to node 1 and withdraw it immediately after.
- (Step 2) $t_0 + M$: Announce r_2 to node 3 and withdraw

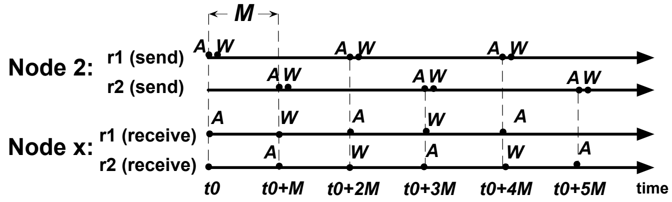


Fig. 5: Example 1: updates on node 2 and node x. A: announcement, W: withdrawal

it immediately after.

. . .

- $t_0 + 2iM$: Repeat step 1.
- $t_0 + (2i + 1)M$: Repeat step 2.

. . .

Each time node 2 announces a path, say r_1, and withdraws it immediately, only the announcement is propagated right away, and the withdrawal is delayed for one MRAI. During this time, node x continues to adopt the path r_1 even though it is not policy-compliant (and any packets forwarded by x along r_1 will get dropped at node 1). When the withdrawal for r_1 arrives at node x, it is followed immediately by an announcement for r_2 but the corresponding withdrawal is delayed by an MRAI. As before, node x will switch to r_2, which is not policy-compliant rendering d unreachable. Note that because RFD is assumed disabled, $r1$ and $r2$ never get damped. As a result, x finds d unreachable forever even though the path r_3 exists throughout. Figure 5 illustrates the resulting events at nodes x and 2.

B. Attacks violating eventual policy prevalence

Example 3 Violating policy prevalence:

Example 1 can be easily extended so as to violate the policy prevalence property. Consider the same topology as in Figure 2, but with one additional good path r_4 from node x to destination d. Suppose x prefers r_3 over r_4. With the strategy outlined in Example 1, node x perpetually adopts r_4 to destination d instead of the more preferred good path $r3$ (as r_3 remains damped forever), thereby violating the policy prevalence property.

a) Discussion: All of the attack examples presented above require the attacker to be on the most preferred path. So it is natural to wonder why these attacks are worrisome given that the attacker can simply drop or reroute the victim's packets in the data plane? There are several reasons for this. First, ASes can detect data plane misbehavior using existing techniques to verify consistency between control plane and data plane actions and react in the event of inconsistencies. However, in the control plane manipulation attacks presented above, the attacker does not violate any specification of "correct" protocol behavior (such as saying one thing and doing another), so it is more difficult to thwart these attacks. Second, protocol manipulation attacks exist irrespective of whether or not data plane verification mechanisms are deployed, so BGP remains insecure unless these attacks are systematically addressed. Third, and most importantly, an attacker can use these attacks to impact routes that it does not directly control, e.g., the attack violating policy prevalence can force an AS to adopt a provider path over a peer path (in violation of typical AS preferences) even though the attacker is on neither of those two paths. The ability to launch such "indirect" attacks makes them significantly more problematic than data plane attacks.

b) Generalizing attack topologies: It is straightforward to observe that if an arbitrary number of ASes are added between any connected ASes in all three examples above, the attacks are still valid. Depending on which BGP mechanism is used to commit the attacks, we name the topologies that are vulnerable to the corresponding attacks the *RFD-vulnerable topologies* and *MRAI-vulnerable topologies* respectively. In a RFD-vulnerable topology, the node that is at the same position as node 1 in Figure 2 is named the *pre-attacker AS*. More formally, the pre-attacker ASes are those ASes that are upstream of the attacker along the (two) most preferred bad route(s) up until and including the first AS that belongs to both of the two most preferred routes. (See [3] for a formal, generalized definition of *vulnerable topology*).

V. Experimental validation

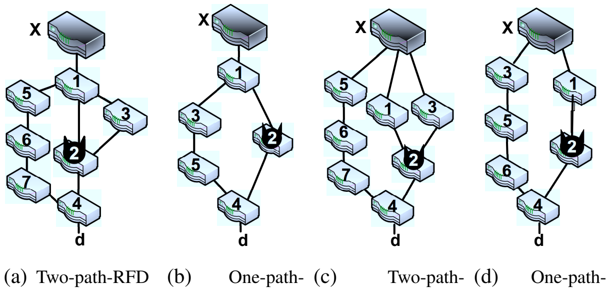
The attacks described "on paper" in the previous section naturally raise the question: Are realistic router implementations vulnerable to these attacks and if so, under what conditions? To answer these questions, we experimentally show that the strategies described above are successful in commodity routers. We also derive sufficient conditions to successfully commit these attacks under realistic Internet-like environments that involves unpredictable delays and MRAI timers.

A. Implementation in commodity routers

Commodity routers typically maintain a penalty value for each peer-destination pair. This implementation reduces CPU and memory usage comparing to the recommendation in RFC 2439 that suggests counting penalty on a per-route base. With per-peer penalties, when a router receives an attribute change or a withdrawal from a peer, the peer's penalty for the destination is increased. After the accumulated penalty reaches the cut-off threshold, any route to the destination announced by the peer will be damped. Unlike the mechanism in RFC 2439, where all the routes (newly announced or old) from the peer can be reused only after the penalty decays to the *reuse threshold*, some commodity implementations adopt a *newly* announced route if the current penalty is below the *cut-off threshold* [28].

B. Experiments with commodity routers

In this section, we present experiments using a commodity Cisco router and Quagga software routers (modified to be consistent with the Cisco implementation) to show that malicious ASes can successfully commit our attacks. The experimental topologies are shown in Figure 6. Each router emulates a single-router AS, so we use *router* and *AS* interchangeably in this section. Routers labeled with numbers are Quagga routers, and router x is a Cisco router. In all the experiments, AS x is the victim, and AS 2 is the attacker, and the destination prefix is announced by AS 4. AS 1 is set to prefer any route announced by the attacker over other routes.



(a) Two-path-RFD (b) One-path-RFD (c) Two-path-MRAI (d) One-path-MRAI

Fig. 6: Experimental topologies showing victim AS X under attack when the attacker AS 2 is an (a) RFD attacker controlling two paths; (b) RFD attacker controlling one path; (c) MRAI attacker controlling two paths; (d) MRAI attacker controlling one path.

1) RFD attacks: We tailor the attack strategy shown in Example 1 in order to account for the specific settings in Cisco routers. First, the attacker must ensure every update in the second stage brings the penalty above the cut-off threshold instead of just the reuse threshold as Cisco settings in some cases allow a route to be used even if the penalty is above the reuse threshold (as explained in Section V-A above). Second, it is unnecessary to send withdrawals immediately after the announcements in the second stage as in Example 1 (though the original strategy also works). This is because Cisco routers maintain per-peer penalties, which makes the attacker's job even easier.

An attacker controlling two paths can disrupt eventual reachability with Cisco settings as follows. Our experimental topology is shown in Figure 6(a), and is similar to Example 1. The attack as before proceeds in two stages. In the first stage, the attacker (AS 2) flaps its route with time interval t_1. After x crosses the cut-off threshold, the attack enters the second stage with a flap time interval of t_2. The actions of AS 2 are shown in Figure 7. The penalties maintained by ASes are shown in Figure 8. AS 1's penalties for AS 2 and AS 3 are roughly the same, so we only show one of them. The figure shows that in the second stage, every flap initiated by the attacker pushes AS x's penalty for AS 1 above the cut-off threshold, so AS 1 remains damped forever. Eventual reachability is violated even though the good path 1-5-6-7-4-d is available throughout.

If AS 1 disables RFD, an attacker flapping one path can disrupt eventual reachability with an even simpler vulnerable topology as shown in Figure 6(b). In this case, attacker AS 2 can flap its route every 1.5 min. The penalty at AS x is shown in Figure 9. Since RFD is disabled at AS 1, AS 2 can continue to flap the route without getting damped, and AS 1 is damped forever by AS x, thereby violating eventual reachability.

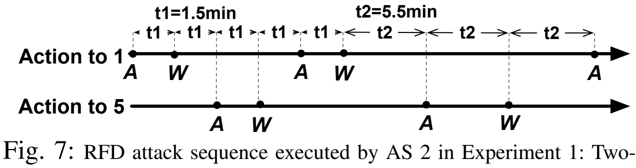
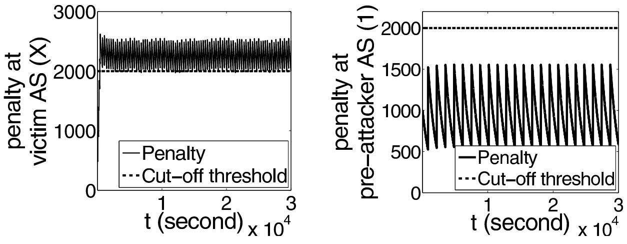


Fig. 7: RFD attack sequence executed by AS 2 in Experiment 1: Two-path RFD attack sequence.



(a) Penalty of AS 1 at AS x (b) Penalty of AS 3 at AS 1

Fig. 8: Experiment 1: Penalty value

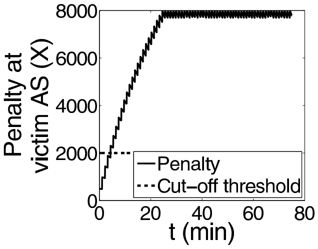


Fig. 9: Experiment 2: Penalty of AS 1 at AS x

2) MRAI attacks: We demonstrate two experiments for MRAI attacks with an attacker controlling two and one most preferred path respectively. We modify Quagga software routers so that MRAI is applied to withdrawals as per the Cisco implementation. To commit a successful attack, the withdrawal message must be sent after the announcement is *adopted* by a victim AS. Thus, in both experiments, we attach a user level machine to the Cisco router, and use `traceroute` to figure out when the announcement is adopted.

The first experiment is conducted with the topology in Figure 6(c), where AS 2 controls two paths, and AS x, 1 and 3 disable RFD. Figure 10 shows the actions of AS 2. t_1 (t_2) represents the time from when AS 2 sends an announcement (withdrawal) to when the announcement (withdrawal) is adopted by AS x. By issuing consecutive ping messages, we record when the path between x and 4 is down. Figure 12 shows the result. Even though the attack was designed to theoretically violate eventual reachability, it only makes the destination unreachable for 88% of time in this experiment. That is because there exists a period of time between when an announcement is sent by the attacker and when the announcement is adopted by x, and during this time interval, the good path $x-5-6-7-4-d$ is used to reach the destination. Nevertheless, the small fraction of reachable time is sufficiently worrisome in practice.

It is possible to conduct an MRAI attack even if the attacker controls just one path. The vulnerable topology is shown in Figure 6(d) where RFD is disabled on AS 1 and AS x. The actions of AS 2 are shown in Figure 11. t_1 and t_2 represent the time from when AS 2 sends an update to the time when the update is adopted by AS x. Figure 13 shows the connectivity between AS x and AS 4 under one path attacks. The destination is unreachable for 45% of the time.

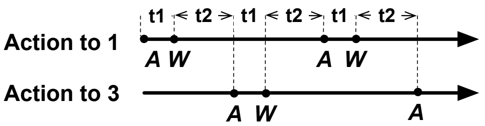


Fig. 10: MRAI attack sequences executed by AS 2 in Experiment 3: Two-path MRAI attack sequence.

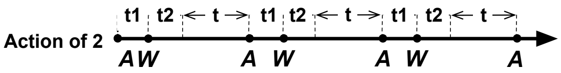


Fig. 11: MRAI attack sequences executed by AS 2 in Experiment 4: Single-path MRAI attack sequence..

C. Sufficient conditions under realistic network delays

In realistic environments, propagation delays—defined here as the time since an update is announced by an attacker until it is received by the victim—may be difficult to predict because of several reasons including route changes, queuing delays, and protocol timers. For example, a given update may face any residual fraction of the MRAI timer at each hop. We have derived sufficient conditions to commit attacks under realistic and unpredictable Internet-like delays. Due to space constraints, we formally describe only the conditions in this paper with detailed proofs are deferred to [3].

1) Sufficient conditions for RFD attacks: Let t_1 and t_2 denote the inter-flap intervals during the first and second stage of RFD attacks. Let C represent the cut-off threshold on the victim AS, and p represent one flap penalty. Then the condition for the penalty at the victim AS to go over the cut-off threshold in the first stage can be written as,

Condition 1: $d < t_1 < \frac{\ln(1-p/C)}{\lambda} - d$

,where d is the longest delay *possible* from the attacker to the victim AS. We can estimate d as the product of the number of AS hops and one MRAI timer. To make the penalty on pre-attacker ASes below the cut-off threshold, the resulting condition that t_1 must satisfy is as follows.

Condition 2: $F(n) < C$

,where $F(n) = 2p\frac{1-e^{\lceil \frac{n-2}{4} \rceil \lambda 4t_1}}{1-e^{\lambda 4t_1}} + 2p$, for all n that $1 \leq n \leq \lceil \frac{\ln(1-\frac{(C-p)1-e^{\lambda t_1}}{pe^{\lambda(t_1+d)}})}{\lambda t_1} \rceil$. Similarly, to maintain the damping forever at the victim AS in the second stage, t_2 should satisfy the following condition.

Condition 3: $Ce^{4\lambda(t_2-d)} + 2p < C$.

We have the following theorem.

Theorem 1. In an RFD-vulnerable topology, if at least one pre-attacker AS enables RFD, then an attacker can violate eventual reachability or policy prevalence using the two-path RFD attack if t_1 satisfies Condition 1 and 2, and t_2 satisfies Condition 1 and 3.

2) Committing MRAI attacks: MRAI attacks are more sensitive to timing than RFD attacks, and we are unable to state credible (i.e., practically achievable) sufficient conditions for an attacker to guarantee that MRAI attacks violate eventual reachability. The key to make the MRAI attack effective in practice is to ensure MRAI victim AS receives and adopts the announcement just before the withdrawal is sent by the attacker. To this end, the attacker can deploy a user-level machine in the MRAI victim AS to monitor when the announcement is adopted as shown in MRAI attack experiments in Section V-B, and a destination can be unreachable for a significant fraction of the time.

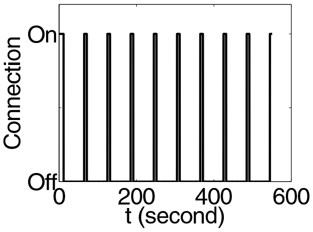


Fig. 12: Attacker controls two best paths

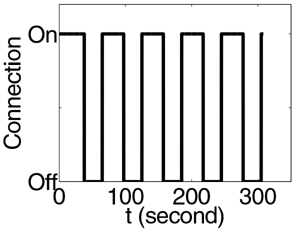


Fig. 13: Attacker controls the best path

VI. Vulnerability in the Internet

In this section, we study the extent to which the Internet is vulnerable to RFD and MRAI attacks, and the topological characteristics of ASes that are particularly vulnerable or those that make for powerful attackers.

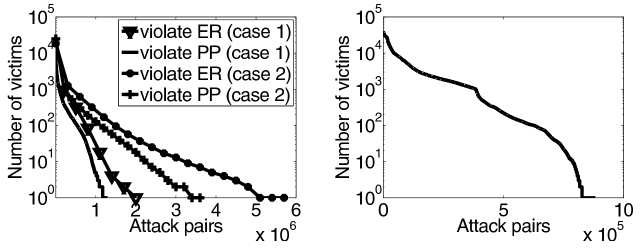
Our high-level methodology is to search for RFD- or MRAI-vulnerable structures in the Internet's AS-level topology that we construct using BGP tables collected between Sep 1-30, 2011 from the Routeviews[2] and RIPE RIS projects [1]. In order to identify vulnerable topologies, we need to know ASes' routing policies that are usually private. However, it is commonly believed that routing decisions largely depend on commercial peering relationships with ties being broken by selecting shorter routes over the longer ones. Our experiments incorporate these assumptions, and the relationships between ASes are inferred using the algorithm proposed in [6].

A. Number of victim ASes

We first analyze the number of potential victim ASes in the Internet. For each destination-attacker pair (referred to as an *attack pair*), we compute the number of vulnerable structures to which they belong and identify the corresponding victims. Because of the compute-intensive nature of this search, we only consider attackers that are on the best path of more than 20 ASes in RFD attacks and more than 1000 in MRAI attacks.

In RFD attack experiment, we assume that all the ASes enable RFD (Figure 14(a)). There are a total of 1.9×10^7 attack pairs, and more than 2.0×10^6 and 1.3×10^6 destination-attacker pairs can make at least one victim AS violate eventual reachability and policy prevalence respectively. Some attack pairs can attack more than 90% of the ASes in the Internet by virtue of the attacker's position. If we assume the pre-attacker ASes in each vulnerable topology disable RFD, the corresponding number is 5.8×10^6 and 3.9×10^6 pairs. This result shows that *with some of the ASes disabling RFD, more ASes can end up being victims.*

In the MRAI attack experiment, we assume that ASes apply the MRAI timer to both announcements and withdrawals. There are a total of 9.4×10^5 attack pairs, and in 8.8×10^5 of those pairs, the attacker can make the destination unreachable to at least one victim AS for a fraction of the time. Note that this fraction can be significant, especially for small vulnerable structures as shown experimentally in Section V-B2.



(a) RFD attacks (ER: eventual reachability, PP: policy prevalence)

(b) MRAI attacks

Fig. 14: Number of victim ASes under attack

B. *Vulnerability of popular destinations*

We evaluate how vulnerable popular destinations are to reachability attacks. As a dominant fraction of Internet traffic is generated by a small number of ASes that are usually large content producers or CDNs[14], analyzing their reachability is of particular interest. We choose the ASes that belong to the top 14 websites ranked by [4] and two popular CDNs[2] as the destination ASes , and compute how many of these destination ASes can become unreachable if any AS can launch RFD or MRAI attacks. We compute the number of victim ASes assuming that (1) pre-attacker ASes disable RFD in the RFD-vulnerable topologies discovered; and (2) MRAI timers are applied to both announcements and withdrawals. Our experiments show that the RFD attack can make *all* the destination ASes persistently unreachable for 3337 (or 8.3%) ASes, and 24032 ASes (or 60.3%) can find at least one of the destination ASes persistently unreachable, while MRAI attacks can cause 18576 ASes (or 46.6%) to find at least one destination AS intermittently unreachable.

We also conducted experiments to assess the characteristics of ASes that make them particularly vulnerable or particularly powerful attackers (deferred to [3]). Our main findings are that under RFD attacks, ASes with a small degree are more vulnerable and ASes with a high degree make for powerful attackers. However, MRAI attacks do not exhibit such a correlation.

VII. Addressing BGP's vulnerabilities

In this section, we describe simple modifications to BGP in order to achieve the two properties—eventual reachability and policy prevalence—despite the presence of malicious nodes. The two key mechanisms we introduce to accomplish this goal are *secure root cause information* and *route stabilization*.

A. *Why root cause information?*

A naive approach to prevent the proposed attacks would be to simply disable RFD and MRAI timers in BGP. However, without these timers, attackers can arbitrarily increase route fluctuations and messaging overhead. Even under benign conditions, it is well known that disabling MRAI timers can result in a super-exponential message complexity in BGP [13]. A practical solution must preserve the benefits of these mechanisms while limiting the impact of the vulnerabilities they introduce.

Our key insight based on the attack scenarios is that the vulnerabilities exist because good ASes do not have enough visibility into an update's root cause. To address this problem, we propose to include *root cause information* in each route update. Root cause has been used to shorten BGP convergence time[16], [19], prevent routing loops [12], etc, and we use it to prevent protocol manipulation attacks.

B. *Secure root cause information*

We determine the root cause of an update as follows. If the update is the result of a router or link failure, the root cause AS is the AS that owns (in case of intra-AS failures) or directly connects with (in case of inter-AS failures) the router or link; the resulting update issued by that AS includes itself as the root cause AS. If the update is a result of a local policy change at a router (i.e., one that is not caused by the receipt of an update from a neighboring router), the root cause AS is the AS to which the router belongs.

Our *secure root cause information* (SRCI) mechanism works as follows. When an update is *initialized* by a router in response to a local failure or policy change, the router includes an SRCI message. The SRCI message contains three fields: the AS number of the "root cause" AS, a timestamp indicating the update initialization time and the destination prefix.

The SRCI message is verifiable using public-key cryptography. Any router receiving an update is required to verify that the root cause AS is indeed the one contained in the received update. Then, the router verifies that the timestamp is greater than or equal to all previously initiated updates from the same root cause AS. Finally, the router verifies that the update it receives has the same destination prefix as the one in the SRCI. If any of these checks fail, the router discards the update.

The RFD procedure is modified using SRCI as follows. A router increases the penalty value for a route iff it receives an update (an announcement or a withdrawal) for the route with a current timestamp such that the root cause AS is along that route. All good routers need to implement RFD with the same parameters. With SRCI, we have the following lemma (with a formal proof in [3]):

Lemma 1 In steady state, with RFD with SRCI, a good route can never get damped.

Lemma 1 ensures policy prevalence. This is simply because if a good route can never get damped, and two or more policy-compliant, good routes to a destination always exist at a router, the router will necessarily either choose the most preferred of those good routes or an even more preferred route if available.

Note that SRCI as defined above does *not* ensure that a route containing a misbehaving bad AS will get damped. This is because bad ASes can potentially reuse timestamps from the most recent routing event that they or other downstream ASes may have initiated. Thus, with SRCI alone, it is possible for

[2] Google, Facebook, Microsoft, Yahoo, Wikipedia, Twitter, LinkedIn, Apple, Ebay, Craigslist, Livejournal, Ask, CNN, About, Akamai, Limelight.

bad ASes to cause nontrivial route flux and message overhead. Furthermore, damping fluctuating routes or ASes alone is insufficient to ensure eventual reachability. The reason is that, although SRCI prevents good routes from being damped, it is not sufficient to ensure that a good router eventually adopts a policy-compliant route (as defined in §III-B1) and finds the destination reachable. For example, it is *possible* that a good router forever keeps adopting routes that result in forwarding loops or blackholes in a network that contains *dispute wheels* [8]. To address these problems, we introduce an additional mechanism called *route stabilization* as described next.

C. Route stabilization

Route stabilization does not require any coordination between routers and works as follows. Each router demarcates its actions corresponding to each destination into epochs. If more than a threshold time T_1 has elapsed since the beginning of the current epoch and the router has not adopted any route that has remained stable for at least a threshold time T_2 in the current epoch, the router switches to a different route if one is available in round-robin order. If the currently adopted route has remained stable for time T_2, that marks the end of the current epoch and the beginning of the next. The time T_1 can be any value (setting it to infinity is equivalent to disabling route stabilization, while setting it to a smaller values intermittently prioritizes stable routes over the router's default policy preferences), but it is critical that $T_2 > Nd$ where N is the length of the longest possible AS path and d is the longest time that each AS can hold an update. Essentially, route stabilization prioritizes stable routes over flappy routes, but unlike RFD that only damps flappy routes, route stabilization additionally selects stable routes proactively.

Putting all of the above together, we have the following theorem (with a formal proof in [3]):

Theorem 2. *If RFD is implemented with SRCI and route stabilization and MRAI is applied only to announcements, BGP achieves eventual reachability and policy prevalence.*

Finally, we note that RFD with SRCI and route stabilization exactly preserve the behavior of BGP when all ASes are good.

VIII. Conclusion

Although BGP has served us well as the Internet's de facto inter domain routing protocol, it is a complex protocol that is still vulnerable to manipulation by malicious ASes despite being augmented with cryptographic security mechanisms such as those in S-BGP. In this paper, we have identified two serious control plane attacks based on manipulating timers. Our key contribution is to formalize these essential and desirable properties; show that BGP does not satisfy them; experimentally validate the feasibility of these attacks on commodity router implementations; and design and implement mechanisms so as to achieve these properties in the presence of malicious ASes.

Acknowledgements

This work is supported by U.S. NSF grants CNS-1040781 and CNS-0917078.

References

[1] *RIPE RIS project.* www.ripe.net/ris.

[2] *Routeviews.* www.routeviews.org.

[3] *Technical report.* `http://rio.ecs.umass.edu/~ysong/protocol_attack_TR.pdf`.

[4] *Top 500 websites.* `www.alexa.com/topsites`.

[5] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford. A Survey of BGP Security Issues and Solutions. In *Proceedings of the IEEE*, Jan 2010.

[6] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9:733–745, 2000.

[7] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford. How Secure are Secure Interdomain Routing Protocols? In *Proceedings of ACM SIGCOMM*, New Delhi, India, Jun 2010.

[8] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10:232–243, 2002.

[9] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying black holes in the internet with hubble. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 247–262, Berkeley, CA, USA, 2008. USENIX Association.

[10] S. Kent, C. Lynn, J. Mikkelson, and K. Seo. Secure border gateway protocol (S-BGP). *IEEE Journal on Selected areas in Communications*, 18, 2000.

[11] E. Kranakis, P. C. V. Oorschot, and T. Wan. On inter-domain routing security and pretty secure BGP (psBGP). *ACM Transactions on Information and System Security (TISSEC)*, 2005.

[12] N. Kushman, S. Kandula, D. Katabi, and B. Maggs. R-BGP: Staying Connected in a Connected World. In *4th USENIX Symposium on Networked Systems Design and Implementation*, Cambridge, MA, 2007.

[13] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proceedings of ACM SIGCOMM*, pages 175–187, 2000.

[14] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-domain traffic. In *proceedings of SIGCOMM*, pages 75–86. ACM, 2010.

[15] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang. Analysis of BGP update surge during slammer worm attack. In *Proceedings of 6th International Workshop on Distributed Computing (IWDC)*, 2003.

[16] J. Luo, J. Xie, R. Hao, and X. Li. An approach to accelerate convergence for path vector protocol. In *Proceedings of Globecom*, November 2002.

[17] P. McDaniel, W. Aiello, K. Butler, and J. Ioannidis. Origin authentication in interdomain routing. *Computer Network*, volume 50, issue 16, Nov 2006.

[18] O. Nordstrom and C. Dovrolis. Beware of BGP attacks. *SIGCOMM Computer Communication Review*, voluem 34, 2004.

[19] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP-RCN: Improving BGP convergence through root cause notification. *Computer Networks ISDN System*, volume 38, June 2005.

[20] Y. Rekhter and T. Li. *A Border Gateway Protocol 4*. RFC 1771, 1998.

[21] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4*. RFC 4271, 2006.

[22] K. Sriram, D. Montgomery, O. Borchert, O. Kim, and D. R. Kuhn. Study of bgp peering session attacks and their impacts on routing performance. *IEEE J.Sel. A. Commun.*, 24(10):1901–1915, Oct. 2006.

[23] L. Subramanian, V. Roth, I. Stoica, S. Shenker, R. H. Katz, and Y. H. Katz. Listen and whisper: Security mechanisms for bgp. In *In Proceedings of First Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

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

[25] D. Wendlandt, I. Avramopoulos, D. G. Andersen, and J. Rexford. Don't secure routing protocols, secure data delivery. In *Proceedings of 5th ACM Workshop on Hot Topics in Networks*, 2006.

[26] R. White. Securing BGP through secure origin BGP (soBGP). *The Internet Protocol Journal*, 6, September 2003.

[27] E. L. Wong, P. Balasubramanian, L. Alvisi, M. G. Gouda, and V. Shmatikov. Truth in advertising: Lightweight verification of route integrity. In *Proceedings of PODC*, 2007.

[28] K. Zhang, S.-T. Teoh, S.-M. Tseng, R. Limprasittipom, K.-L. Ma, and S. Wu. Performing BGP experiments on a semi-relistic Internet testbed environment. In *The 2nd International Workshop on Security in Distributed Computing Systems (SDCS-2005)*, 2005.