Network Protocol Attacks on BGP and Potential Solutions

Abstract

BGP, the Internet's de facto interdomain routing protocol, is well known to have many security vulnerabilities due to the very nature of its underlying assumptions of trust among independently operated networks. As more and more critical services become dependent on the Internet, the risks posed by malicious autonomous systems (ASes) are becoming increasingly worrisome. Over the last decade, researchers have studied a variety of attacks on BGP in the presence of malicious ASes, however most prior efforts have focused on attacks such as prefix hijacking, spoofing, forging routes, etc., that lend themselves well to solutions based on traditional cryptographic techniques to ensure authentication or integrity, e.g., S-BGP and follow-on works. Although augmenting BGP with authentication and integrity mechanisms is critical, they are far from sufficient to prevent attacks based on manipulating the complex BGP protocol itself. In this paper, we identify two serious, previously unknown attacks on two of the most fundamental goals of BGP—to ensure reachability and enable ASes to pick routes according to their routing policies—even in the presence of S-BGP-like mechanisms. Our key contribution is to formalize a series of critical properties, show that (S-)BGP fails to achieve those properties, and propose simple mechanisms to ensure that those properties are satisfied.

1 Introduction

The Internet is composed of thousands of Autonomous Systems (ASes), each of which is managed by a self-administrated domain such as an Internet service provider (ISP) or organization. The interdomain routing protocol, Border Gateway Protocol (BGP), is used to exchange routing information on how to reach a destination through a path vector protocol. Each router announces its best path to a destination to its neighbors and in turn the neighbor determines the best path based on the routes it has received. The design of BGP assumes that independently operated networks are trustworthy. That is, each network announces only prefixes it originates and only the best path to neighbors. In practice, given the utter lack of security mechanisms in BGP as deployed today, malicious ASes can launch a variety of attacks such as prefix hijacking, spoofing [4, 19], altering or faking routes, unauthorized prefix aggregation/de-aggregation, etc.

A number of comprehensive security solutions have been proposed to ensure the integrity of BGP routing information. These solutions use cryptographic techniques to authenticate all routes and/or prefixes received [10, 27, 11, 18]. Furthermore, proposals for ensuring data plane security have been presented. For example, verification mechanisms guaranteeing that the paths selected in the control plane are actually used in data plane for forwarding traffic have been proposed [28].

On the surface, it appears that combining route authentication with data plane security is sufficient to ensure that BGP behaves as desired. Indeed, more recent works have been focusing on other practical challenges related to deployability of BGP security solutions such as incentives for adoption, benefits of incremental deployability, and so on. However, we show, in this paper, that the fundamental security vulnerabilities in BGP are far from solved. Even with authenticated paths and a verified data forwarding plane, BGP is vulnerable to rather serious attacks in the

presence of malicious ASes. For example, a malicious AS or router can cause a well-behaved AS to lose connectivity to a destination even though the malicious AS is not on the path that the well-behaved AS chooses. More disturbingly, it is possible for the malicious AS to block the destination from the victim AS *permanently*.

To better understand and address such vulnerabilities, we first formally identify a series of desirable properties for an interdomain routing protocol. These properties can be informally summarized as follows. The first property specifies that if a router/AS has a policy-compliant route to a destination consisting of only non-malicious ASes, the routing protocol should guarantee the reachability of the destination. The second property 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. The third property generalizes the first and specifies that if a router/AS has a good route to a destination consisting of only non-malicious ASes, the routing protocol should guarantee a nonzero lower bound on the fraction of time when the destination is reachable.

We then show that BGP can violate these properties with a malicious AS or router. In our attack scenarios, a malicious AS can take advantage of the fact that most routers employ Routing Flap Damping (RFD) and/or Minimum Route Advertisement Interval (MRAI) timers. Although these timers are used to ensure the stability of the routing protocol and to reduce message overhead, malicious ASes can abuse them in a manner that effectively makes a good route disappear from the victim's routing table. Finally, we provide simple modifications to address the exposed vulnerabilities in BGP. The proposed solution ensures all of the presented properties can be satisfied. To the best of our knowledge, this is the first work that shows that, even with S-BGP like mechanisms, BGP is vulnerable to serious attacks on two of its most fundamental goals: ensuring reachability and enabling routers to pick routes according to their policy preferences.

The paper is organized as follows. Section 2 describes related work and introduces the background of BGP protocols. In Section 3, we present the desirable properties of BGP. Section 4 describes the attacks that violate the desirable properties. We propose a solution and its effectiveness in Section 5. Finally, we conclude in Section 6.

2 Related Work and Background

2.1 Related work

BGP was designed at the early age of the Internet when the network had not yet been so complex and massive. Consequently, it simply assumes that all routers or ASes are trustworthy. When a neighbor announces a route or prefix, an AS or router assumes that the neighbor indeed has the route to reach the prefix. As a result, it is easy for a malicious router or AS to perform attacks. For example, one can perform a prefix hijacking attack by announcing the prefix it does not own. Further, if it is successful in attracting the traffic to the prefix, it can drop traffic for the prefix once delivered and lead to blackhole the prefix. As another example, a malicious AS can perform path spoofing attack by announcing to its provider a path it does not have in order to attract traffic. Once the traffic is attracted, the malicious AS can degrade or delay the traffic. These scenarios are not fictitious and has contributes to many of Internet connectivity outage incidents [3, 2, 1]. For a comprehensive description of BGP atttacks, see [4].

A number of security solutions have been proposed to secure BGP. Origin authentication [18] uses a trusted database to verify IP prefix ownership. Secure Origin BGP (soBGP) [27] provides both origin authentication and the verification of whether the announced path physically exists. S-BGP [10] is proposed to not only verify prefix ownership and existence, but also to use digital

signatures to authenticate each BGP update message. These methods are proposed for control plane security. For data plane security, a data-plane verification technique [4, 28] is proposed to guarantees that a path that appears in a BGP announcement message is actually being used for forwarding traffic.

While these proposed security extensions protect BGP from many well-known routing protocol attacks such as prefix hijacking and path spoofing [4], it is not clear that they are sufficient to achieve BGP security. In this paper, we ask the question that if S-BGP like authentication and integrity mechanism and data plan verification technique are in place, is BGP vulnerable to attacks. Our answer is positive. That is, BGP is still vulnerable to attacks. Our work is related to an earlier study [6], where the authors also point out that the existing security protocols are not enough to prevent attacks. However, they focus on attacks that attract traffic, while our focus is on attacks that cause undesirable behavior of BGP such as loss of connectivity and choosing less preferred paths.

2.2 BGP mechanisms to maintain routing stability

BGP is an incremental protocol. Whenever a router's best route changes, it would announce an update message to its neighbors. Thus, if the changes happen too frequently, they may cause global update flooding. 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 Min Route Advertisement Timer (MRAI) that ensures the time interval between consecutive updates to be large enough. Even though these two mechanisms can improve stability, they are known to delay routing convergence as shown in [5, 23]. In this paper, we show that, they can also be used to commit routing protocol attacks.

2.2.1 Routing Flap Damping (RFD)

RFD [24] is a mechanism designed to discourage the selection of unstable routes. Each router maintains a route penalty associated with every route announced by neighbors. The penalty measures the instability of a route. Whenever the route changes, the route penalty is increased by a fixed value. If the penalty of the route exceeds the *cut-off threshold*, the route cannot be used for selecting the best route. That is, the route gets damped. The penalty value decays exponentially over time. The decay rate is determined by a parameter called the *half-life parameter*. It represents the amount of time it takes for the penalty to decrease to half of its value. When the penalty decays below the *reuse threshold*, the route can be used for selecting the best route. That is, the route will not be damped. To avoid over punishing a stable route, if a route has been *stable* for more than the *max suppress time*, it will not be damped no matter how unstable it had been before the stable period.

We illustrate how RFD works in Figure 1. Suppose the penalty of a route is zero initially. At time t_1, the route gets flapped, and its penalty increases. Because the penalty value does not reach the cut-off threshold, the route is not damped. The penalty decays exponentially between t_1 and t_2. At t_2, the route gets flapped again which increase the penalty of the route. Similarly, between t_2 and t_3, the penalty decays exponentially. At time t_3, the route flaps again and this pushes the penalty above the cut-off threshold. As a result, the route gets damped. The route cannot be used until the penalty decays below the reuse threshold at t_5. During the damping period between t_3 and t_5, the penalty increases once again because of a flap at t_4. At time t_5, penalty falls at the reuse threshold, and the route is no longer damped.

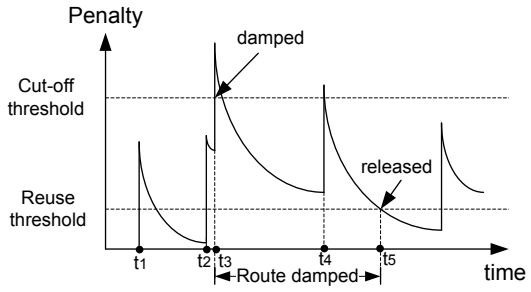

Figure 1: An example illustrating how RFD works

Table 1: RFD default parameters in Cisco and Juniper routers

RFD parameter	Cisco	Juniper
Withdrawal penalty	1000	1000
Re-advertisement penalty	**0**	**1000**
Attributes change penalty	500	500
Cutoff threshold	**2000**	**3000**
Half-life (min)	15	15
Reuse threshold	750	750
Max suppress time (min)	60	60

RFD is not a compulsive option. ASes can enable or disable it. As shown in [15], RFD has not been fully deployed even within the Internet core. RFD also lacks uniform implementation standards [17]. Different router vendors have different default cut-off thresholds. They also count penalty differently. For example, Cisco routers ignore announcements in penalty, while Juniper routers treat them the same as withdrawals. The default parameter settings for Juniper and Cisco routers are shown in Table 1.

2.2.2 Min Route Advertisement Interval (MRAI)

MRAI is the minimum amount of time that must pass between consecutive announcements of a route [21]. It limits frequency of route announcements that each node sends to neighbors. In a recently published BGP4 protocol specification [22], in addition to announcements, withdrawals are suggested to be limited by the MRAI timer. This departs from the previous BGP protocol specification [21] where route withdrawals can be sent without waiting for the expiration of MRAI timer.

The default value of MRAI timer recommended by RFC is 30 seconds. However, the setting is disputable. Some router vendors and network operators lower or remove the timer entirely [8, 9], because it impacts the performance of time-sensitive applications [25, 12]. But another study [5] shows that setting different values for MRAI timers in different routers may prolong BGP convergence.

3 Protocol vulnerabilities in BGP

3.1 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 and in collusion with other bad ASes. 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 it proof that each AS along the route announced the corresponding prefix of the route. For example, if an AS receives a route $[AS_k, \cdots, AS_2, AS_1]$ for some prefix P, then it can verify that AS_1 owns P, and for each $1 < i < k$, AS_i announced the route $[AS_i, \cdots, AS_1]$ to AS_{i+1}. 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, spoofing, or incorrectly forwarding data packets) are outside the scope of this paper. Although data plane attacks necessarily need to be addressed for end-to-end security, 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 [10][27][11][19] but is by no means a universally accepted position [26].

The stringent 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.

3.2 Desirable properties

3.2.1 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.

[1]This definition ignores path-prepending [21, 22] for the sake of simplicity. Allowing path-prepending does not materially alter our results.

A route to a prefix is said to be *policy-compliant* at a router if every intermediate AS (the first AS) along the route is currently willing to route traffic destined for that prefix from the immediately upstream AS (the router's AS) via the immediately downstream AS, and the last AS along the route owns the prefix. A route to a prefix is said to have been *adopted* by a router if it has been selected for forwarding packets to the prefix. At any point in time, multiple routes to a destination may exist in a router's routing table, but at most one of those is adopted by the router.

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. We also note that *unreachable* does not necessarily imply that forwarded packets will not arrive at the destination. A router may forward packets along a recently adopted route that is no longer policy-compliant (as a downstream router may have changed its route), but packets may still arrive at the destination (provided the downstream router's newly adopted route does not result in a loop or a blackhole).

3.2.2 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. Thus, if there exists a period of time when the destination is reachable, then the property is satisfied. We show that the current BGP can not even satisfy this weak property.

3.2.3 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)*: In steady-state, when two or more policy-compliant, good routes to a destination exist at a router, the destination must be reachable via a path that is at least as preferred as the most preferred of those routes.*

3.2.4 η-reachability

The eventual reachability property above is rather weak as it only requires reachability to become *eventually* true. So, a protocol that simply satisfies said condition for an instant will technically satisfy the property. Furthermore, if the reachability condition becomes true for a tiny period followed by a long period when it is not true, and this alternating pattern repeats forever, then Property 1 is satisfied as at any point in time the reachability condition is guaranteed to eventually become true. Clearly, Property 1 even if satisfied is too weak to be satisfying. Thus, we introduce Property 3 that is a stronger generalization of Property 1 as follows.

Property 3 (η-reachable): *If a destination is reachable from a router in steady-state in the good AS-subgraph, then the destination must be reachable for at least a fraction $\eta < 1$ of time in steady-state in the original AS-graph.*

The parameter η is a constant and the higher its value is, the better. η is defined to be strictly less than 1 as it is not possible to *always* satisfy the condition in Property 3 (even with just benign failures), and our focus is on properties that are desirable as well as achievable with simple modifications to BGP.

4 Attack Mechanisms

In this section, we show that BGP does not satisfy any of the properties introduced in the previous section. To this end, we present several simple example scenarios in which bad nodes execute a sequence of actions so as 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. For simplicity, the examples in this section assume that each AS consists of a single router; they can be easily extended to multiple-router ASes.

4.1 Attacks violating eventual reachability (ER)

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

4.1.1 Example 1: Violating ER using MRAI

Consider the topology in Figure 2. 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 $r2$ over $r3$. 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 [22]. For simplicity, we assume in this example that the RFD timer is disabled on all nodes (an assumption that is removed in the next example).

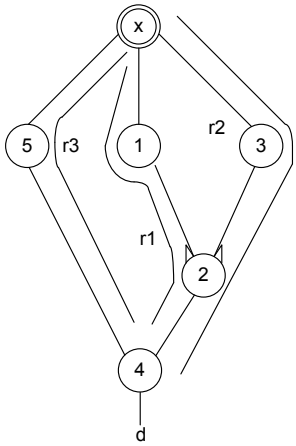

Figure 2: Example 1: violating eventual reachability using MRAI

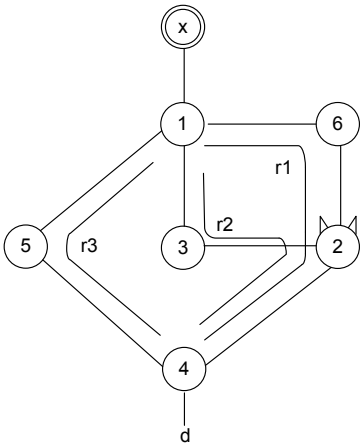

Figure 3: Example 2: violating eventual reachability using RFD

The attack involves the malicious node 2 making a sequence of announcements and withdrawals that result in d becoming permanently unavailable to x. This sequence of steps and the times when node 2 executes them is as listed below, where t_0 is an arbitrary starting time, M refers to the length of an MRAI interval (typically about 30 seconds [21, 22]), and the step index i is a nonnegative integer.

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

To appreciate why the above sequence of steps causes d to become unreachable to x, consider Figure 4 that illustrates the resulting events at nodes x and 2 respectively. 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 interval. 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 interval. 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.

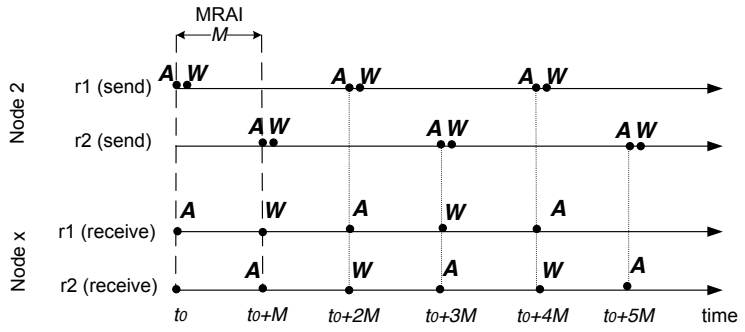

Figure 4: Example 1: updates on node 2 and node x. A: announcement, W: withdrawal

4.1.2 Example 2: Violating ER using RFD

Next, we show how an attacker can abuse the RFD timer to violate ER. Consider the topology shown in Figure 3. 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 $r2$. Path $r3$ is good. We make the following assumptions: (1) RFD is enabled on node 1 and node x with the same parameters, (2) both announcements and withdrawals count towards the RFD penalty, and (3) MRAI is applied on announcements only, i.e., withdrawals are propagated immediately. These assumptions are made for simplicity of exposition. Subsequently, we show that the attack can be mounted 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. The first stage involves node 2 making the following sequence of announcements and withdrawals.

Stage 1:

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

  ...

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

Figure 5 illustrates why the above sequence of steps results in node x damping r_3. 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's flap counter for both r_1 and r_3 is 2. A similar sequence of flaps happens when node 2 announces and withdraws $r2$. Thus, after r_2 is withdrawn, x's flap counter for r_3 has reached 4 while its counter for both r_1 and r_2 is 2. As a result, r_3's penalty exceeds the cut-off threshold and x damps r_3. More generally, suppose the cut-off threshold is T_c, and the penalty value increases by p whenever the route gets flapped, and

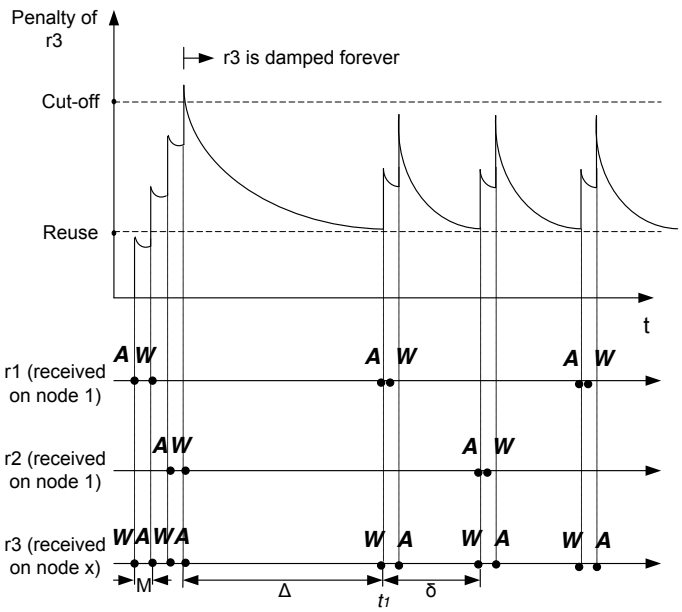

Figure 5: $r3$'s penalty on node x in response to updates. A: announcement, W: withdrawal.

during one MRAI timer, the penalty value decays by d. Then, to make node x damp $r3$, node 2 has to commit at least n steps, where $n = \lceil \frac{T_c - p}{p-d} + 1 \rceil$. The time when node x damps $r3$ is $t_0 + \lceil \frac{T_c - p}{p-d} \rceil M$.

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 at that point.

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 a victim AS for a finite length of time Δ. Next, we extend the attack to make the destination unreachable forever, thereby violating ER. This stage, referred to as Stage 2, is as shown below. Stage 2 begins at time t_1 that denotes the first time when x's penalty decays to the reuse threshold. The parameter δ below refers to the time it takes for the penalty to decay back to the reuse threshold after two flaps have pushed it above the reuse threshold. The step index i is a nonnegative integer.

Stage 2:

- (Step 1) t_1: Announce r_1 to node 6 and withdraw it immediately after.
- (Step 2) $t_1 + \delta$: Announce r_2 to node 3 and withdraw it immediately after.
  ...
- $t_1 + 2i\delta$: Repeat step 1.
- $t_1 + (2i+1)\delta$: Repeat step 2.
  ...

As shown in Figure 5, Stage 2 causes x's penalty to remain above the reuse threshold forever. Although as in Stage 1, the flaps also cause x to increase the penalty for r_1 and r_2, the increase is not sufficient for their penalties to exceed the cut-off threshold. Throughout Stage 2, r_3 remains

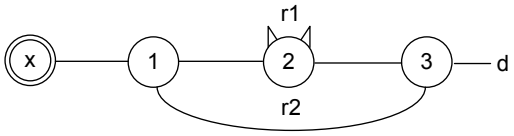

Figure 6: Example 4: violating η-reachability with RFD

damped and neither r_1 nor r_2 is available to x, so the destination d remains unreachable to x forever, thereby violating ER.

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. We discuss several such changes. First, if RFD is applied to only withdrawals and not announcements, the attack in Example 2 still violates ER; the only consequence is that Stage 1 takes longer to execute. Second, if MRAI is applied to both announcements and withdrawals at each node, the attack in Example 2 still violates ER. In this case, node 6 (node 3) will not immediately propagate the withdrawal for r_1 (r_2), so x may continue to adopt the route r_1 (r_2) for an MRAI interval, however the adopted route is not policy-compliant as node 6 (node 3) has locally withdrawn the route, so the destination is *unreachable*. Third, we have verified that if RFD is implemented on a per-neighbor basis as opposed to a per-route basis, i.e., the flap counter is maintained for each neighbor and incremented when any route to the destination through that neighbor is announced or withdrawn, we can construct an example (omitted for brevity) even simpler than Example 2 that violates ER.

4.2 Attacks violating eventual policy prevalence

Example 3: Example 2 can be easily extended so as to violate the policy prevalence property. Consider the same topology as in Figure 3, but with one additional good path r_4 from node x to destination d. The new figure is shown in Appendix A, Figure 8. Suppose x prefers r_3 over r_4. With the strategy outlined in Example 2, 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 eventual policy prevalence property.

4.3 Attacks violating η-reachability

The property of η-reachability is a stronger requirement compared to eventual reachability. Thus, all the examples that violate eventual reachability also violate η-reachability. Indeed, for many values of η, an attacker can violate η-reachability with even simpler examples than the ones described above. In this section, we analyze one example in detail. More examples are presented in Appendix C.

Example 4: Consider the situation in Figure 6, where RFD is enabled on node x and node 1 with the same parameters. Node 1 has two paths to reach d, $r1$: $1-2-3-d$, and $r2$: $1-3-d$. Node 1 prefers $r1$ over $r2$. Node 2 is the attacker.

We make the following assumptions: (1) both announcements and withdrawals count towards RFD penalty, and (2) MRAI is applied on announcements only, and (3) node x and node 1 need k updates to damp a route, where k is an even number. In Appendix C, we show that the attack can be mounted even without these assumptions.

Suppose the initial state is that node 2 does not announce $r1$ to node 1, and thus node 1 announces $r2$ to node x. Then, node 2 can follow the sequence of announcements and withdrawals

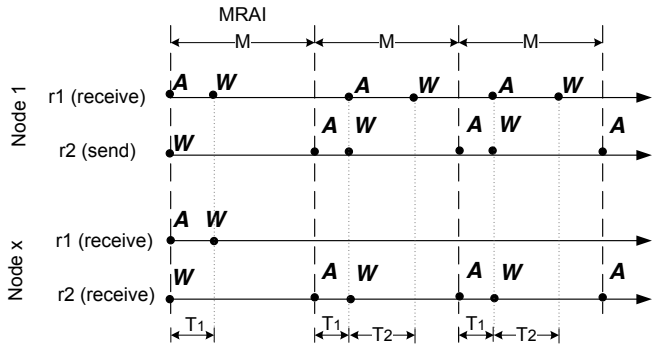

Figure 7: Example 4: updates on node 1 and node x (k is an even number)

below to force x to damp $r2$. T_1 and T_2 are time intervals where $T_1 + T_2 < M$, and the step index i is a nonnegative integer.

- (Step 1) t_0: Announce $r1$.
- (Step 2) $t_0 + T_1$: Withdraw $r1$.
  ...
- $t_0 + iM + T_1$: Repeat Step 1.
- $t_0 + iM + T_1 + T_2$: Repeat Step 2.
  ...

The process is illustrated in Figure 7. Because k is an even number, the k-th update of $r1$ on node 1 is a withdrawal. After the withdrawal, node 1 would damp $r1$ and announce $r2$ to node x. The announcement would be the k-th updates of $r2$ on node x, and node x damps $r2$ as well. As a result, d is not reachable for node x. Suppose node x damps $r2$ for time T_d, and then the fraction of reachability under this attack is $\frac{\frac{k}{2}T_1}{\frac{k}{2}M+T_d}$. Note that, since T_d is much longer than T_1 and M, the fraction of reachability should be a very small number. The attacks under different conditions can be found in Appendix C.

5 Addressing BGP's vulnerabilities

In this section, we describe a simple modification to BGP in order to achieve the three properties—eventual reachability, policy prevalence, and η-reachability—despite the presence of malicious nodes. The key mechanism we introduce to accomplish this goal is *verifiable root cause information*, as described next.

5.1 Verifiable root cause information

A naive approach to prevent the attacks presented in the previous section would be to simply disable RFD and MRAI timers in BGP. Indeed, disabling them suffices to achieve eventual reachability and policy prevalence (that as defined are rather weak properties), but introduces more serious problems.

Without these timers, attackers can arbitrarily increase route fluctuations and messaging overhead in BGP. Even under benign conditions, it is well known that disabling MRAI timers can result in a super-exponential message complexity in BGP [14]. Thus, a practical solution must preserve the benefits of these timers while limiting the impact of the vulnerabilities they introduce.

Our key insight based on the attack scenarios presented earlier is that the vulnerabilities exist because good ASes do not have enough visibility into the root cause of an update. This allows attackers to implicate good, upstream ASes or good routes passing through them in the eyes of further upstream ASes. Even under benign conditions, the lack of visibility is known to cause reachability problems in BGP, e.g., the scenario in Example 1 could occur because of an unstable link and result in unavailability for one MRAI interval despite the availability of an alternate route. However, malicious nodes can exploit this lack of visibility and dramatically exacerbate the consequences.

To address this problem, we propose to include *root cause information* (RCI) in each route update message. The idea of RCI itself is not new, and has been explored in [16] [20] to improve BGP convergence time by reducing number of update messages and intermediate route changes before convergence. In [13], RCI has been used to prevent routing loops during convergence. In contrast to these works, our focus is on using an RCI-based scheme to protect against security vulnerabilities in BGP.

Our proposed *verifiable RCI* mechanism works as follows. In each route update issued by a router, it includes a 16-bit RCI field that contains the AS number of the "root cause" AS corresponding to the update. This root cause AS is determined as follows. If the update is the result of a router or intra-AS link failure, the root cause AS is the AS that owns that router or link. If the update is the result of an inter-AS link failure, the ASes at both ends of the link are considered as root cause ASes; the resulting update issued by each 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. Finally, if a router issues an update in response to an update received from a neighboring router, it retains the same RCI as in the received causal update.

The RCI is verifiable, i.e., any AS receiving an update can verify that the root cause AS is indeed the one contained in the received update. To this end, any router that sets the RCI field also includes a digital signature computed over the RCI and a current timestamp using a private key that is common to all routers in the AS. Upon receiving an update, a router checks that (1) the update contains a timestamp greater than all previously initiated updates from that root cause AS, (2) the RCI and timestamp tuple has been signed by the AS included in the RCI field. Otherwise, the router discards the update[2].

The RFD procedure is modified using verifiable RCI as follows. For each route in the route information base, a router maintains the RFD penalty value. The router increases the penalty value for a route iff it receives an update (an announcement or a withdrawal) for the route such that the verifiable RCI in the update contains an AS that is along that route. All good routers are required to implement RFD with the same parameters.

Finally, we require MRAI timers to be enforced at both the sender and receiver as follows. After an AS issues an update announcing a route to a neighbor, it can not withdraw that route or announce a different route for the same prefix to that neighbor for an MRAI interval. All good

[2]Note that this simple design allows a bad upstream AS A to supplant a downstream root cause AS B and insert itself as the root cause AS in an update, but in this case we consider A to be the root cause AS thereafter; as we shall see, the distinction between whether or not an AS claiming to be the root cause is actually the root cause is immaterial from a security standpoint.

routers are required to implement the same value of the MRAI interval. If the AS sending the updates violates this requirement, the receiving AS will set the penalty of all routes to the prefix announced by the sending AS to the RFD cut-off threshold.

In the rest of this section, we refer to the combination of the modified RFD and MRAI mechanisms as described above as *update verification*.

5.2 Ensuring eventual reachability

Theorem 1 *BGP with update verification satisfies the eventual reachability property.*

Proof: Suppose eventual reachability (ER) is violated at a node x with respect to a destination d. Then one of the following is always true at x: (1) x has no route to d in its routing table (as none are being announced by any of its neighbors); (2) x has damped all routes to d in its routing table; (3) x has adopted a route that is not policy-compliant.

We first show that cases 1 and 2 above can not occur by contradiction. ER violation at x with respect to d implies that in the good AS-subgraph in steady-state, x finds d reachable, i.e., x adopts a policy-compliant path to d. Let $x, y_1, y_2, \cdots, y_k, d$ denote this path. By definition of the good AS-subgraph, all ASes in this path are good. For convenience of notation below, we use y_0 to refer to x.

Now consider the original AS-graph. If x has no route to d being announced by any of its neighbors, then it must be the case that y_1 is not announcing any route for d to x. So either y_1 has no route to d or it has damped all the routes to d in its routing table. Let $i < k$ be the smallest index such that y_i has at least one route to d in its routing table. As y_i is not announcing any route for d to y_{i-1}, it must be the case that y_i has damped all routes to d in its routing table.

However, y_i could not have damped the route, if any, announced by y_{i+1}. The reason is that, as y_i and y_{i+1} are both good and therefore employ the same RFD parameters, y_{i+1} would also have damped that route and not announced it to y_i in the first place. Thus, it must be the case that y_{i+1} is not announcing any route to y_i, which can happen only if y_{i+1} has damped all routes to d in its routing table. We can continue with this reasoning and arrive at the conclusion that y_1 has damped all routes to d, which is clearly a contradiction.

Thus, x must always have in its routing table at least one undamped route for d announced by y_1 in the original AS-graph. So ER can be violated only if case 3 holds, i.e., x always adopts a route that is not policy-compliant. We argue that this can not happen if network propagation delays are much smaller than the MRAI interval (as is true in practice) and the number of attackers is small (specifically, smaller than the ratio of the time-to-decay from the RFD cut-off to reuse threshold and the MRAI interval)[3]. If x always adopts a route to d that is not policy-compliant, it means that at least one AS further downstream along the adopted route always changes its route after announcing it and before x receives it. However, the modified MRAI scheme above prevents an AS from changing its route for at least an MRAI interval, and damps the route if the AS attempts otherwise. Thus, eventually x either adopts a good route to d (that is guaranteed to be policy-compliant for at least an MRAI interval) or adopts a policy-compliant route through one or more bad ASes. In either case, ER is satisfied. Hence, the theorem is true.□

η-reachability Update verification ensures more than just eventual reachability. We can show that in the absence of pathological policy configurations resulting in dispute wheels [7], update verification ensures η-reachability for $\eta > 0$. Let M denote the MRAI interval, and T the maximum

[3]We note that typical values of the former are on the order of tens of minutes and the latter tens of seconds

propagation delay along any (acyclic) path in the AS graph, D the time-to-decay from the RFD cut-off threshold to the reuse threshold, and K the number of bad ASes. Then, BGP with update verification ensures η-reachability for $\eta > \min(1 - \frac{T}{M}, 1 - \frac{KM}{D})$.

We note that the assumptions underlying the above result are rather restrictive. Most importantly, bad nodes have no reason to restrict their policies in any particular way, and may indeed attempt to introduce dispute wheels. The presence of dispute wheels means that routing may never converge and ASes may go through several configurations where they select inconsistent (non-policy-compliant) routes resulting in forwarding loops. However, ASes have an economic disincentive to create dispute wheels, which is why they are known to be rare in practice. Furthermore, the absence of dispute wheels is a sufficient but not a necessary condition to achieve the η above. We also note that although the achievable η in the result above is rather modest (compared to expectations of "multiple nines" people have for Internet availability), it is vastly better than what can be said for BGP in the presence of malicious ASes today as well as with SBGP.

5.3 Ensuring policy prevalence

Theorem 2 *BGP with update verification satisfies the policy prevalence property.*

Proof: Violating policy prevalence requires that a router x has at least two policy-compliant, good routes r_1, r_2 to a destination in its routing table in steady-state such that r_1 is more preferred that r_2, but it adopts a route that is less preferred than r_1. The only reason this can happen is if x has damped r_1. However, update verification by design never damps a good route in steady-state, as good ASes are never the root cause of updates (as there are no failures or policy changes in steady-state). Thus, r_1 must be undamped. □

6 Conclusion

Although BGP has served us well as the Internet's interdomain de facto routing protocol, it has long been recognized as having serious security vulnerabilities. BGP security has seen over a decade of work, however most prior work has focused on augmenting BGP with properties such as authentication and integrity and, more generally, on security issues that can be addressed in significant part using traditional cryptographic techniques. However, BGP being a complex protocol, is still vulnerable to manipulation by malicious ASes even when augmented with cryptographic security mechanisms such as those in SBGP. In this paper, we identify two serious and previously unknown attacks based on manipulating timers that show that, in the presence of even a single malicious AS, (S)BGP poorly satisfies its two most fundamental objectives, namely, to ensure reachability and enable ASes to choose routes according to their policy preferences. Our key contribution is to formalize these essential and desirable properties, show that BGP does not satisfy them, and propose mechanisms so as to achieve them in the presence of malicious ASes.

References

[1] R. Barrett, S. Haar, and R. Whitestone. *Routing snafu causes Internet outage.* Interactive Week, 1997-04-25.

[2] Rensys Blog. *Con-Ed Steals the Net.* `http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml`, 2006.

[3] Rensys Blog. *Pakistan Hijacks YouTube*. `http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml%`, 2008.

[4] Kevin Butler, Toni R. Farley, Patrick McDaniel, and Jennifer Rexford. A Survey of BGP Security Issues and Solutions. In *Proceedings of the IEEE*, Jan 2010.

[5] Alex Fabrikant, Umar Syed, and Jennifer Rexford. There is something about MRAI: Timing diversity may exponentially worsen BGP convergence. In *Proceedings of Infocom*, 2011.

[6] Sharon Goldberg, Michael Schapira, Peter Hummon, and Jennifer Rexford. How Secure are Secure Interdomain Routing Protocols? In *Proceedings of ACM SIGCOMM*, New Delhi, India, Jun 2010.

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

[8] Paul Jakma. *Revisions to the BGP: Minimum Route Advertisement Interval*. Internet Draft draft-ietf-idr-mrai-dep-02, 2010.

[9] Juniper. *Out-delay*. https://www.juniper.net/techpubs/software/junos/ junos57/swconfig57-routing/html/bgp-summary32.html, 2010-07-30.

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

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

[12] Nate Kushman, Srikanth Kandula, and Dina Katabi. Can You Hear Me Now?! It Must be BGP. In *Computer Communication Review*, March 2007.

[13] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce Maggs. R-BGP: Staying Connected in a Connected World. In *4th USENIX Symposium on Networked Systems Design and Implementation*, Cambridge, MA, April 2007.

[14] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet routing convergence. In *Proceedings of ACM SIGCOMM*, pages 175–187, 2000.

[15] Mohit Lad, Xiaoliang Zhao, Beichuan Zhang, Dan Massey, and Lixia Zhang. Analysis of BGP update surge during slammer worm attack. In *Proceedings of 6th International Workshop on Distributed Computing (IWDC)*, 2003.

[16] Jiazeng Luo, Junqing Xie, Ruibing Hao, and Xing Li. An approach to accelerate convergence for path vector protocol. In *Proceedings of Globecom*, November 2002.

[17] Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. BGP beacons. In *Proceedings of Internet Measurement Conference*, Miami, Florida, USA, 2003.

[18] Patrick McDaniel, William Aiello, Kevin Butler, and John Ioannidis. Origin authentication in interdomain routing. *Computer Network*, volume 50, issue 16, Nov 2006.

[19] Ola Nordstrom and Constantinos Dovrolis. Beware of BGP attacks. *SIGCOMM Computer Communication Review*, voluem 34, 2004.

[20] Dan Pei, Matt Azuma, Dan Massey, and Lixia Zhang. BGP-RCN: Improving BGP convergence through root cause notification. *Computer Networks ISDN System*, volume 38, June 2005.

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

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

[23] Martin Suchara, Alex Fabrikant, and Jennifer Rexford. BGP safety with spurious updates. In *Proceedings of Infocom*, 2011.

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

[25] Feng Wang, Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A measurement study on the impact of routing events on end-to-end Internet path performance. In *Proceeding of SIGCOMM*, 2006.

[26] Dan Wendlandt, Ioannis Avramopoulos, David G. Andersen, and Jennifer Rexford. Don't secure routing protocols, secure data delivery. In *Proceedings of 5th ACM Workshop on Hot Topics in Networks*, 2006.

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

[28] Edmund L. Wong, Praveen Balasubramanian, Lorenzo Alvisi, Mohamed G. Gouda, and Vitaly Shmatikov. Truth in advertising: Lightweight verification of route integrity. In *Proceedings of PODC*, 2007.

A The figure of example 3 violating eventual policy prevalence

B A general example that violates eventual reachability with RFD

Let us consider a more general situation shown in Figure 9. Suppose node x must go through node 1 to reach destination d. Node 1 has several available paths, but the n ($n \geq 1$) most preferred paths are controlled by the attacker node 2, and the $(n+1)$-th preferred path is a good path.

We first consider the case that node 1 has a higher cut-off threshold than node x. In that case, the attacker can flap any of the bad paths until the bad path is damped by x. If node 2 keeps announcing the same bad path to node 1, node x only has the damped bad path rendering d unreachable. Whenever the RFD penalty drops to the reuse threshold, node 2 should flap the bad path again so that it is never released by x. As a result, d is unreachable forever.

Next, we suppose node 1 has an equal or lower cut-off threshold than node x. Let node 2 announce and withdraw the n paths one by one. Eventually node x would damp the good path. Whenever the penalty of the good path decays to the reuse threshold, let node 2 re-announce and immediately withdraw one of the n paths. As a result, d is unreachable forever.

Suppose every malicious path is flapped k times before the good path gets damped. As a result, the good path would have nk flaps. Assume that node 1 and node x need T_1 and T_x flaps to damp a route respectively, where $T_1 \leq T_x$. To force node x to damp the good path, we require: $T_x < nk$. At the same time, the penalty of the malicious paths should be under node 1's cut-off threshold so

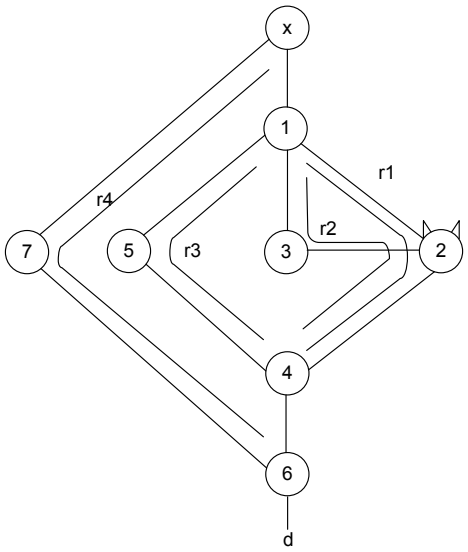

Figure 8: An example violating eventual policy prevalence

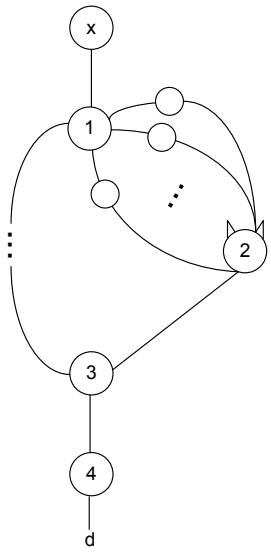

Figure 9: A general topology that violates eventual reachability with RFD

that they can flap the route to keep the good path beyond node x's reuse threshold forever. That is, $T_1 > k$. Therefore, given T_1 and T_x, node 2 should control at least n most preferred paths of node 1 to violate eventual reachability, where $n \geq \lceil \frac{T_x+1}{T_1-1} \rceil$.

C More examples of violating η-reachability

C.1 Example 4 under different conditions

We first analyze the case where (1) both announcements and withdrawals are counted in RFD penalty, and (2) MRAI is only applied on announcements, and (3) k is an odd number.

Suppose the initial state is that node 2 announces $r1$ to node 1. Then, node 2 can follow the steps below to force x to damp $r2$.

- **(Step 1)** t_0: Withdraw $r1$.
- **(Step 2)** $t_0 + T_1$: Announce $r1$.
- **(Step 3)** $t_0 + T_1 + T_2$: Withdraw $r1$.
  ...

- $t_0 + iM + T_1$: Announce $r1$.
- $t_0 + iM + T_1 + T_2$: Withdraw $r1$.
  ...

Because $r1$ and $r2$ are damped at the same time, the malicious node cannot commit further attacks until both paths are released. Thus, unlike example 2, the attacker cannot always keep $r2$'s penalty beyond the reuse threshold. This results in intermittent unreachability.

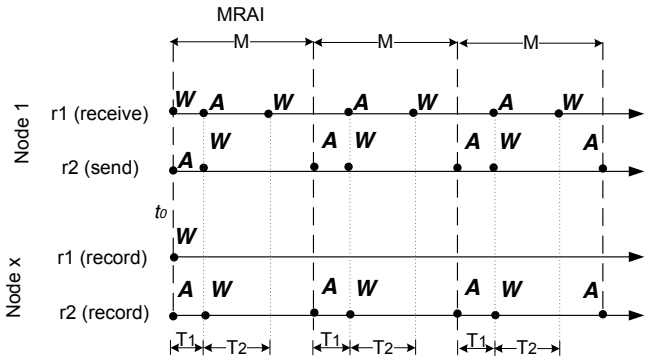

Figure 10: Example 4: violate η fraction reachability (k is an odd number)

Next, we analyze how to repeat the attacks to form intermittent reachability. After the first attack, node 2 does not announce $r1$ to node 1. Thus, to commit another attack, node 2 needs to re-announce $r1$, and waits until $r1$ and $r2$'s penalty to decay to a value so that both of them need k more flaps to be damped. Suppose the waiting time is δ. Then, every time node 2 repeats the attack, the fraction of reachability is $\frac{\delta+\frac{k-1}{2}T_1}{\delta+\frac{k-1}{2}M+T_d}$, where T_d is the duration of damping.

When only withdrawals count towards RFD penalty, the strategy with k as an even number can be used to commit the attack. Under this strategy, $r2$ has one more withdrawal on x before $r1$ has on node 1. Thus, the method can force x to damp $r2$. After that, node 2 needs to withdraw $r1$ once again to make sure node 1 announces $r2$ to x.[4] As a result, node 1 damps $r1$ as well. When MRAI is applied to both announcements and withdrawals, node 2 can send announcements and withdrawals of $r1$ iteratively whenever the MRAI timer on node 1 expires. As a result, node 1 would announce $r1$ and $r2$ iteratively whenever its MRAI timer expires, and the previous path is automatically withdrawn. In that case, $r2$ still gets the same number of update records on node x as $r1$ does on node 1. Eventually, $r2$ would be damped by x. How to repeat the attack is the same as mentioned above.

C.2 Violating η-reachability with MRAI

Next, we give two examples showing that the property of η-reachability can be violated as well.

Example 5: Consider the topology in Figure 11. The victim node x has two paths $r1$ and $r2$. $r1$ is the more preferred path which is controlled by the attacker node 2, and $r2$ is a good path. Suppose MRAI is applied on both announcements and withdrawals.

Let node 2 announce $r1$ and withdraw it immediately. Node 1 can send the announcement without delay, but it cannot withdraw the path until MRAI timer expires. During this time, node x keeps using $r1$ rather than the good path $r2$, and d is unreachable. $r1$'s updates on node x and node 1 are shown in Figure 12. Because RFD is disabled, the attacker can repeat the attack without being damped. As a result, for $\frac{1}{2}$ fraction of the time, d is unreachable.

Example 6: In this example, the attacker uses MRAI timer to form a routing loop. Consider the topology in Figure 13. Node $x1$ and $x2$ have three paths to reach the destination d. They both prefer the path directly going through node 2. If that path is not available, they would use the path going through each other. Their least preferred path goes through node 3. We show node x1's three paths in Figure 13. Its preference order is $r1 > r2 > r3$.

[4]Note that, $r2$ is never damped by node 1.

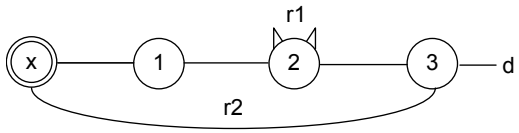

Figure 11: Example 5: violating η-reachability with MRAI

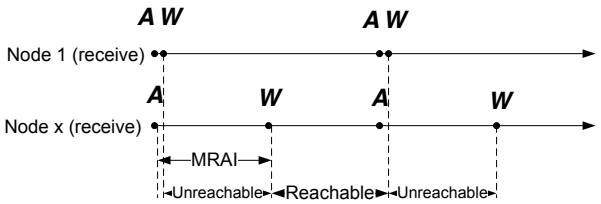

Figure 12: $r1$'s record on node 1 and x in example 5. W: withdrawal. A: announcement.

Suppose RFD is disabled, and MRAI is applied on both announcements and withdrawals. We construct an attack which delays the withdrawals of the paths directly going though node 2, and thus $x1$ and $x2$ would form a loop by choosing each other as the next hop.

Node 2 can commit the following actions.

- (Step 1) t_0: Withdraw its path to $x1$ and $x2$.
- (Step 2) $t_0 + M$: Announce its path to $x1$ and $x2$, and withdraw immediately.
  ...
- $t_0 + 2iM$: Repeat step 2.
  ...

According to the steps, whenever node 2 announces its path, the announcement can be propagated without delay. However, the following withdrawal would be delayed for one MRAI. During this time, $x1$ and $x2$ would prefer each other as the next hop, and a loop is formed. Figure 14 shows the updates on node x1 and node x2 with time. As can be seen in the figure, d is unreachable for $\frac{1}{2}$ fraction of the time.

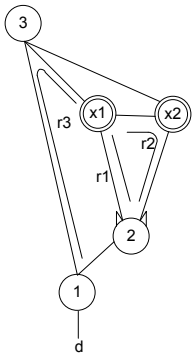

Figure 13: Example 6: violating η-reachability with MRAI

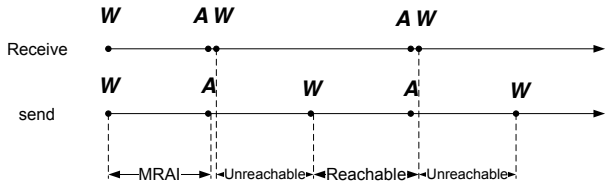

Figure 14: Events on node x1 and x2 in example 6. W: withdrawal. A: announcement.