How Secure are Secure Interdomain Routing Protocols?

Sharon Goldberg
Microsoft Research

Michael Schapira
Yale & UC Berkeley

Peter Hummon
AT&T Labs

Jennifer Rexford
Princeton

ABSTRACT

In response to high-profile Internet outages, BGP security variants have been proposed to prevent the propagation of bogus routing information. To inform discussions of which variant should be deployed in the Internet, we *quantify* the ability of the main protocols (origin authentication, soBGP, S-BGP, and data-plane verification) to blunt traffic-attraction attacks; *i.e.,* an attacker that deliberately attracts traffic to drop, tamper, or eavesdrop on packets.

Intuition suggests that an attacker can maximize the traffic he attracts by *widely* announcing a *short* path that is not flagged as bogus by the secure protocol. Through simulations on an empirically-determined AS-level topology, we show that this strategy is surprisingly effective, even when the network uses an advanced security solution like S-BGP or data-plane verification. Worse yet, we show that these results *underestimate* the severity of attacks. We prove that finding the most damaging strategy is NP-hard, and show how counterintuitive strategies, like announcing longer paths, announcing to fewer neighbors, or triggering BGP loop-detection, can be used to attract even more traffic than the strategy above. These counterintuitive examples are not merely hypothetical; we searched the empirical AS topology to identify specific ASes that can launch them. Finally, we find that a clever export policy can often attract almost as much traffic as a bogus path announcement. Thus, our work implies that mechanisms that police export policies (*e.g.,* defensive filtering) are crucial, even if S-BGP is fully deployed.

Categories and Subject Descriptors. C.2.2 Computer Communication Networks: Network Protocols.
General Terms. Security.

1. INTRODUCTION

The Internet is notoriously vulnerable to *traffic attraction* attacks, where Autonomous Systems (ASes) manipulate BGP to attract traffic to, or through, their networks. Attracting extra traffic enables the AS to increase revenue from customers, or drop, tamper, or snoop on the packets [2–4]. While the proposed extensions to BGP prevent many attacks (see [5] for a survey), even these secure protocols are susceptible to a *strategic* manipulator who deliberately exploits their weaknesses to attract traffic to its network. Given the difficulty of upgrading the Internet to a new secure routing protocol, it is crucial to understand how well these protocols blunt the impact of traffic attraction attacks.

1.1 Quantifying the impact of attacks

We evaluate the four major extensions to BGP, ordered from weakest to strongest: origin authentication [6,7], soBGP [8], S-BGP [9], and data-plane verification [5,10]. While the stronger protocols prevent a *strictly* larger set of attacks than the weaker ones, these security gains often come with significant implementation and deployment costs. To inform discussions about which of these secure protocols should be deployed, we would like to *quantitatively* compare their ability to limit traffic attraction attacks. Thus, we simulate attacks on each protocol on an empirically-measured AS-level topology [11–13], and determine the percentage of ASes that forward traffic to the manipulator.

Performing a quantitative comparison requires some care. It does *not* suffice to say that one protocol, say S-BGP, is four times as effective as another protocol, say origin authentication, at preventing *a specific type of attack strategy*; there may be *other attack strategies* for which the quantitative gap between the two protocols is significantly smaller. Since these more clever attack strategies can just as easily occur in the wild, our comparison must be in terms of the *worst possible attack* that the manipulator could launch on each protocol. To do this, we put ourselves in the mind of the manipulator, and look for the *optimal* strategy he can use to attract traffic from *as many ASes as possible*.

However, before we can even begin thinking about optimal strategies for traffic attraction, we first need a model for the way traffic flows in the Internet. In practice, this depends on local routing policies used by each AS, which are not publicly known. However, the BGP decision process breaks ties by selecting shorter routes over longer ones, and it is widely believed [14] that policies depend heavily on economic considerations. Thus, conventional wisdom and prior work [14–16] suggests basing routing policies on business relationships and AS-path lengths. While this model (used in many other studies, *e.g.,* [2,17]) does *not* capture all the intricacies of interdomain routing, it is still very useful for *gaining insight* into traffic attraction attacks. All of our results are attained within this model.

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.

1.2 Thinking like a manipulator

If routing policies are based on AS path lengths, then intuition suggests that it is optimal for the manipulator to announce the *shortest path* that the protocol does not reject as bogus, to *as many neighbors* as possible. Depending on the security protocol, this means announcing a direct connection to the victim IP prefix, a fake edge to the legitimate destination AS, a short path that exists but was never advertised, a short path that the manipulator learned but is not using, or even a legitimate path that deviates from normal export policy. Indeed, we use simulations on a measured AS-level topology to show that this "smart" attack strategy is quite effective, even against advanced secure routing protocols like S-BGP and data-plane verification.

Worse yet, we show that our simulations *underestimate* the amount of damage manipulator could cause. Through counterexamples, show that the "smart" attack is surprisingly *not optimal*. In fact, the following bizarre strategies can sometimes attract even more traffic than the "smart" attack: announcing a *longer* path, exporting a route to *fewer* neighbors, or triggering BGP's *loop-detection* mechanism. In fact, we show that prefix hijacking (i.e., originating a prefix you do not own) is *not* always the most effective attack against today's BGP! These counterexamples are not merely hypothetical—we identify specific ASes in the measured AS-level topology that could launch them. Moreover, we prove that it is NP-hard to find the manipulator's optimal attack, suggesting that a comprehensive comparison across protocols must remain elusive.

1.3 Our findings and recommendations

While we necessarily *underestimate* the amount of damage a manipulator could cause, we can make a number of concrete statements. Our main finding is that secure routing protocols only deal with one half of the problem: while they do restrict the *paths* the manipulator can announce, they fail to restrict his *export policies*. Thus, our simulations show that, when compared to BGP and origin authentication, soBGP and S-BGP significantly limit the manipulator's ability to attract traffic by announcing bogus short paths to all its neighbors. However, even in a network with S-BGP or data-plane verification, we found that a manipulator can still attract traffic by cleverly *manipulating his export policies*. Indeed, we found that announcing a *short* path is often less important than exporting that path to the *right set of neighbors*. Thus:

- Advanced security protocols like S-BGP and data-plane verification do *not* significantly outperform soBGP for the "smart" attacks we evaluated.
- Defensive filtering of paths exported by stub ASes (*i.e.,* ASes without customers) provides a level of protection that is *at least* comparable to that provided by soBGP, S-BGP and even data-plane verification.
- Tier 2 ASes are in the position to attract the largest volumes of traffic, even in the presence of data-plane verification and defensive filtering (of stubs).
- *Interception attacks* [2,3]—where the manipulator both attracts traffic and delivers it to the destination—are easy for many ASes, especially large ones.

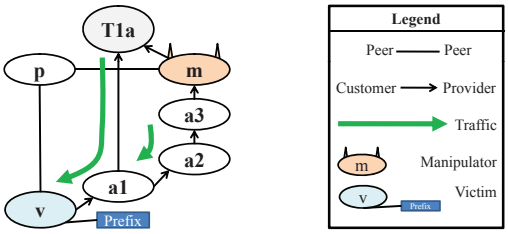

Figure 1: Anonymized subgraph of CAIDA's AS graph.

We could quibble about whether or not manipulating export policies even constitutes an *attack*; after all, each AS has the right to decide where it announces paths. However, our results indicate that a clever export policy can attract almost as much traffic as a bogus path announcement. Indeed, Section 6.1 presents an example where an AS in the measured topology gains almost as much exporting a provider-learned path to another provider, as he would by a prefix hijack (announcing that he owns the IP prefix). Thus, our results suggest that addressing traffic attraction attacks requires both mechanisms that prevent bogus path announcements (*e.g.,* soBGP or S-BGP) as well as mechanisms that police export policies (*e.g.,* defensive filtering).

Full version. This paper is a compressed summary of our results; the full version [1] presents additional information, graphs, related work, and proofs of our theorems.

2. MODEL AND METHODOLOGY

We first present a model of interdomain routing and routing policies, based on the standard models in [18] and the Gao-Rexford conditions [15], followed by our threat model for traffic attraction, and finally our experimental setup.

2.1 Modeling interdomain routin.

The AS graph. The interdomain-routing system is modeled with a labeled graph called an *AS graph*, as in Figure 1. Each AS is modeled as a single node and denoted by its AS number. Edges represent direct physical communication links between ASes. Adjacent ASes are called *neighbors*. Since changes in topology typically occur on a much longer timescale than the execution of the protocol, we follow [18] and assume the AS-graph topology is static. BGP computes paths to each destination IP prefix separately, so we assume that there is a unique *destination IP prefix* to which all other nodes attempt to establish a path. As shown in Figure 1, there is a single AS v that rightfully 'owns' the destination IP prefix under consideration.

Establishing paths. In BGP, an AS first chooses an outgoing edge on which it forwards traffic based on a local ranking on outgoing paths, and then announces this path to some subset of its neighbors. To model this, we assume that each node n has a set of *routing policies*, consisting of (a) a *ranking* on outgoing paths from n to the destination d, and (b) a set of *export policies*, a mapping of each path P to the set of neighbors to which n is willing to announce the path P. We say that node n has an *available path* aPd if n's neighbor a announced the path "aPd" to n. If an available path aPd is ranked higher than the outgoing path that node n is currently using, then an *normal* node n will (a) forward

traffic to node a, and (b) announce the path $naPd$ to all his neighbors as specified by his export policies.

Business relationships. We annotate the AS graph with the standard model for business relationships in the Internet [15]; while more complicated business relationships exist in practice, the following is widely believed to capture the majority of the economic relationships in the Internet. As shown in Figure 1, there are two kinds of edges: *customer-provider* (where the customer pays the provider for connectivity, represented with an arrow from customer to provider), and *peer-to-peer* (where two ASes owned by different organizations agree to transit each other's traffic at no cost, represented with an undirected edge). Because some of our results are based on CAIDA's AS graph [11], we also consider *sibling-to-sibling* edges. Details about our treatment of siblings is in the full version [1]. Finally, our theoretical results sometimes use [15]'s assumption that an AS cannot be its own indirect customer:

GR1 The AS graph contains no customer-provider cycles.

2.2 Modeling routing policies

In practice, the local routing policies used by each AS in the Internet are arbitrary and not publicly known. However, because we want to understand how false routing information propagates through the Internet, we need to concretely model routing policies. Since it is widely believed that business relationships play a large role in determining the routing policies of a given AS [14, 15], and we have reasonably accurate empirical maps of the business relationships between ASes [11–13], we base our model on these relationships.

Rankings. BGP is first and foremost designed to prevent loops. Thus, we assume that node a rejects an announcement from its neighbor b if it contains a *loop*, *i.e.,* if node a appears on the path that node b announces. Beyond that, we can think of the process ASes use to select routes as follows; first applying local preferences, then choosing shortest AS paths, and finally applying a tie break. Since the local preferences of each AS are unknown, and are widely believed to be based (mostly) on business relationships, we model the three step process as follows:

LP **Local Preference.** Prefer outgoing paths where the next hop is a customer over outgoing paths where the next hop is a peer over paths where the next hop is a provider.

SP **Shortest Paths.** Among the paths with the highest local preference, chose the shortest ones.

TB **Tie Break.** If there are multiple such paths, choose the one whose next hop has the lowest AS number.[1]

Our model of local preferences is based on on Gao-Rexford condition **GR3**, and captures the idea that an AS has an economic incentive to prefer forwarding traffic via customer (that pays him) over a peer (where no money is exchanged) over a provider (that he must pay). Notice that this implies that an AS can sometimes prefer a *longer* path! (*e.g.,* in Figure 1, AS m prefers the five-hop customer path through $a3$ over the four-hop provider path through Tier 1 $T1$.)

[1]We need a consistent way to break ties. In practice, this is done using the intradomain distance between routers and router IDs. Since our model does not incorporate geographic distance or individual routers, we use AS number instead.

Export Policies. Our model of export policies is based on the Gao-Rexford condition **GR2**:

GR2 AS b will only announce a path via AS c to AS a if at least one of a and c are customers of b.

GR2 captures the idea that an AS should only be willing to load his own network with transit traffic if he gets paid to do so. However, because **GR2** does *not* fully specify the export policies of every AS (for instance, an AS could decide to export paths to only a *subset* of his customers), it does not suffice for our purposes. Thus, we model normal export policies as follows:

NE An AS will announce *all* paths to *all* neighbors *except* when **GR2** forbids him to do so.

2.3 Threat model.

One strategic manipulator. We assume that all ASes in the AS graph behave *normally*, *i.e.,* according to the policies in Section 2.1 - 2.2, except for a *single manipulator* (*e.g.,* AS m in Figure 1). We leave models dealing with colluding ASes for future work.

Normal ASes and normal paths. We assume that every *normal* AS uses the routing policies in Section 2.2; thus, the *normal path* is the path an AS (even the manipulator) would choose if he used the normal rankings of Section 2.2, and *normal export* is defined analogously. (*e.g.,* In Figure 1, the manipulator m's *normal path* is through his customer AS $a3$.) We shall assume that every normal AS knows its business relationship with his neighbors, and also knows the next hop it chooses for forwarding traffic to a given destination. In order to evaluate the effectiveness of each secure routing protocol, we assume that ASes believe everything they hear, *except* when the secure routing protocol tells them otherwise. As such, we do not assume that ASes use auxiliary information to detect attacks, including knowledge of the network topology or business relationships between distant ASes, *etc.*, unless the secure routing protocol *specifically* provides this information.

Attraction *v.s.* Interception attacks. In an *attraction attack*, the manipulator's goal is to attract traffic, *i.e.,* to convince the maximum number of ASes in the graph to forward traffic that is destined to the *victim IP prefix* via the manipulator's own network. To model the idea that a manipulator may want to eavesdrop or tamper with traffic before forwarding it on to the legitimate destination, we also consider *interception attacks*. In an interception attack, the manipulator has the additional goal of ensuring that he has an *available path to the victim*. This is in contrast to an attraction attack, where the manipulator is allowed, but not required, to *create a blackhole* where he has no working path to the victim IP prefix (*e.g.,* Figure 6).

The fraction of ASes attracted. In this paper, we measure the success of an attack strategy by counting *the fraction of ASes in the internetwork* from which that manipulator attracts traffic; this amounts to assuming that every AS in the internetwork is of equal importance to the manipulator.[2] However, it is well known that the distribution

[2]We acknowledge that a manipulator may want to attract traffic from a *specific subset* of ASes. We avoid analyzing this, because we lack empirical data to quantify that subset of ASes that a given manipulator may want to attract.

of traffic in the Internet is *not* uniform across the ASes; to address this, we also report the fraction of ASes of *various sizes* from which the manipulator attracts traffic, where we measure size by the number of direct customers the AS has.

Attack strategies. To capture the idea that the manipulator is strategic, we allow him to be more clever than the normal ASes; specifically, we allow him to use knowledge of the global AS graph and its business relationships in order to launch his attacks. (However, most of the strategies we considered require only knowledge that is locally available at each AS.) An attack strategy is a set of routing announcements and forwarding choices that deviates from the normal routing policies specified in Section 2.2. An attack strategy may include, but is not limited to:

- Announcing an unavailable or non-existent path.
- Announcing a legitimate available path that is *different* from the *normal path*.
- Exporting a path (even the legitimate normal path) to a neighbor to which *no path* should be announced to according to the normal export policies.

Indeed, one might argue that some of these strategies do not constitute 'dishonest behavior'. However, it is important to consider these strategies in our study, since we shall find that they can sometimes be used to attract as much traffic as the traditional 'dishonest' strategies (*e.g.,* announcing non-existent paths).

Scope of this paper. This paper focuses on traffic attraction attacks; we do not consider other routing security issues, for instance, mismatches between the control- and data-plane [4, 10], or traffic deflection attacks, where a manipulator wants to divert traffic from himself or some distant, innocent AS [5]. Moreover, we do not cover issues related to the adoption of secure routing protocols, nor their effectiveness under partial deployment [19]. See the full version [1] for more discussion of related work.

2.4 Experiments on empirical AS graphs

All the results and examples we present are based on empirically-obtained snapshots of the Internet's AS graph annotated with business relationships between ASes. Our experimental results were obtained via algorithmic simulations; details are in the full version [1].

Average case analysis. Since the influence of an attack strategy depends heavily on the locations of the manipulator and the victim in the AS graph, we run simulations across many (manipulator, victim) pairs. Rather than reporting average results, we plot the *distribution* of the fraction of ASes that direct traffic to the manipulator. We by no means believe that a manipulator would select its victim at random; however, reporting distributions allows us to measure the extent to which a secure protocol can blunt the power of the manipulator, determine the fraction of victims that a manipulator could effectively target, and identify positions in the network that are effective launching points for attacks. Ideally, to determine how damaging a given attack strategy can be, we would have liked to run simulations over *every* (manipulator,victim) pair in the AS graph. However, this would require $(30\text{K})^2$ simulations per dataset, which would be prohibitive. Instead, we run experiments on randomly-chosen (manipulator, victim) pairs. We found that 60K experiments of each type were sufficient for our results to stabilize.

Multiple datasets. Because the actual AS-level topology of the Internet remains unknown, and inferring AS relationships is still an active area of research, we run simulations on a number of different datasets: multiple years of CAIDA data [11], and Cyclops data [12] augmented with 21,000 peer-to-peer edges from [13]'s IXP dataset. Even though these datasets use different relationship-inference algorithms, the trends we observed across datasets were remarkably consistent. Thus, all the results we present are from CAIDA's November 20, 2009 dataset (with slight modifications to the sibling relationships, see the full version); counterparts of these graphs, computed from Cyclops and IXP data [12, 13] are in the full version [1].

Realistic examples. Rather than providing contrived counterexamples, we give evidence that the attack strategies we discuss could succeed in wild by ensuring that *every example we present comes from real data*. All the examples we present here were found in CAIDA's November 20, 2009 dataset [11], and then "anonymized" by replacing AS numbers with symbols (*e.g.,* in Figure 1, m for manipulator, v for victim, $T1$ for a Tier 1 AS, *etc.*). We do this in order to avoid 'implicating' innocent ASes with our example attacks, as well as to avoid reporting potentially erroneous AS-relationship inferences made in the CAIDA dataset (see Section 6.4 for further discussion).

3. FOOLING BGP SECURITY PROTOCOLS

This section overviews the security protocols we consider, and presents *the set of (possibly) bogus paths that a manipulator can announce to each neighbor without getting caught*. We use the anonymized subgraph of CADIA's AS graph in Figure 1 to demonstrate the fraction of traffic a manipulator m could attract by announcing one of these (possibly) bogus paths to all its neighbors.

Our focus is on protocols with well-defined security guarantees. Thus, we consider the five major BGP security variants, ordered from weakest to strongest security, as follows: *(unmodified) BGP, Origin Authentication, soBGP*, *S-BGP*, and *data-plane verification*. Because we focus on security guarantees and not protocol implementation, we use these as an umbrella for many other proposals (see [5] for a survey) that provide similar guarantees using alternate, often lower-cost, implementations. Furthermore, our ordering of protocols is strict: an attack that succeeds against a strong security protocol, will also succeed against the weaker security protocol. We also consider *defensive filtering* as an orthogonal security mechanism.

BGP. BGP does not include mechanisms for validating information in routing announcements. Thus, the manipulator can get away with announcing any path he wants, including (falsely) claiming that he is the owner of the victim's IP prefix. Indeed, when the manipulator m in Figure 1 (an anonymized Canadian Tier 2 ISP) launches this attack on the v's IP prefix (an anonymized Austrian AS), our simulations show that he attracts traffic from 75% of the ASes in the internetwork.[3]

[3]In fact, another strategy, called a *subprefix hijack*, is available to manipulator; by announcing a longer, more specific subprefix of the victim's IP prefix, he can attract traffic from 100% of the ASes in the internetwork. This work does not

Origin Authentication. Origin authentication [6] uses a trusted database to guarantee that an AS cannot falsely claim to be the rightful owner for an IP prefix. However, the manipulator can still get away with announcing any path that *ends* at the AS that rightfully owns the victim IP prefix. For instance, in Figure 1, the manipulator m can attract traffic from 25% of the ASes in the internetwork by announcing the path (m, v, Prefix), even though no such path physically exists.

soBGP. Secure Origin BGP (soBGP) [8] provides origin authentication as well as a trusted database that guarantees that any announced path *physically exists* in the AS-level topology of the internetwork. However, a manipulator can still get away with announcing a path that *exists* but is not actually *available*. In Figure 1, the manipulator m can attract traffic from 10% of the ASes in the internetwork by announcing the path (m, p, v, Prefix). Notice that this path is unavailable; **GR2** forbids the Swiss Tier 2 ISP p to announce a peer path to another peer.

S-BGP. In addition to origin authentication, Secure BGP [9] also uses cryptographically-signed routing announcements to provides a property called *path verification*. Path verification guarantees that every AS a can only announce a path abP to its neighbors if it has a neighbor b that *announced* the path bP to a. Thus, it effectively limits a single manipulator to announcing available paths. For instance, in Figure 1, the manipulator's *normal path* (see Section 2.3) is the five-hop customer path (m, $a3$, $a2$, $a1$, v, Prefix); announcing that path allows him to attract traffic from 0.9% of the ASes in the internetwork. However, with S-BGP the manipulator could instead announce the *shorter* four-hop provider path (m, $T1$, $a1$, v, Prefix), thus doubling attracted traffic to 1.7%. Indeed, S-BGP does *not* prevent the manipulator from *announcing* the shorter, more expensive, provider path, while actually *forwarding traffic* on the cheaper, longer customer path.

Data-plane verification. Data-plane verification [5, 10] prevents an AS from announcing one path, while forwarding on another. Thus, if the manipulator in Figure 1 wants to maximize his attracted traffic, he must also forward traffic on the provider path.

Defensive Filtering. Defensive filtering polices the BGP announcements made by stubs. A *stub* is an AS with no customers, and in our model, **GR2** implies that a stub should *never* announce a path to a prefix it does not own. Thus, our model of defensive filtering has each provider keep a "prefix list" of the IP prefixes owned by its direct customers that are stubs. If a stub announces a path to *any* IP prefix that it does not own, the provider drops/ignores the announcement, thus enforcing **GR2**. In most of our analysis, we assume that *every provider* in the internetwork correctly implements defensive filtering (see also the discussion in Section 8). As such, we assume that *defensive filtering completely eliminates all attacks by stubs.*

4. SMART ATTRACTION ATTACKS

We simulate attraction attacks on measured graphs of the Internet's AS-level topology [11–13] to determine how much traffic a manipulator can attract in the *average case*. This section first presents the attack strategies we simulated, and then reports our results.

consider subprefix hijacks, mostly because these attacks are well understood, but also because they can be prevented by the filtering practices discussed in [5].

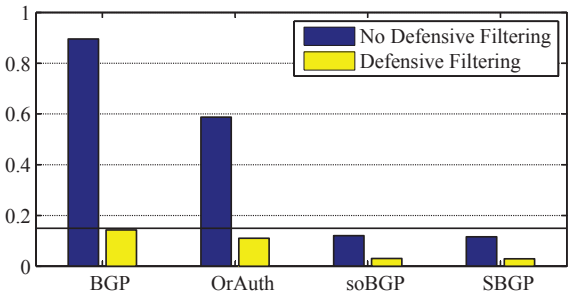

Figure 2: Lower bounds on the probability of attracting at least 10% of ASes in the internetwork.

4.1 A smart-but-suboptimal attack strategy

We assumed that ASes make routing decisions based on business relationships and path length, and that a manipulator m cannot lie to his neighbor a about their business relationship (*i.e.,* between m and a). Thus, intuition suggests that the manipulator's best strategy is to widely announce the shortest possible path:

"Shortest-Path Export-All" attack strategy. Announce to *every* neighbor, the *shortest* possible path that is *not flagged as bogus by the secure routing protocol.*

Every "Shortest-Path Export-All" attack strategy on S-BGP is also an attack on data-plane verification. The "Shortest-Path Export-All" attack strategy on S-BGP has the manipulator announce his shortest *legitimate available path* to the victim, instead of his *normal path* (see Sections 2.3 and 3). Notice that if the manipulator actually decides to *forward* his traffic over the announced path, he has a successful attack on data-plane verification as well! Thus, the "Shortest-Path Export-All" attack strategy on data-plane verification is *identical* to the attack on S-BGP. (To reduce clutter, the following mostly refers to the attack on S-BGP.)

We underestimate damage. Section 6 shows that the "Shortest-Path Export-All" attack strategy is *not* actually optimal for the manipulator, and Section 7 shows that finding the optimal attack strategy is NP-hard. Thus, we give up on finding the *optimal* attack strategy, and run simulations assuming that the manipulator uses this smart-but-suboptimal attack. This means that the results reported in this section *underestimate* the amount of damage a manipulator could cause, and we usually *cannot* use these results to directly compare different secure routing protocols. In spite of this, our simulations do provide both (a) useful lower bounds on the amount of damage a manipulator could cause, and (b) a number of surprising insights on the strategies a manipulator can use to attract traffic to his network.

4.2 Defensive filtering is crucial

Our first observation is that defensive filtering is a crucial part of any Internet security solution:

Figure 2: We show the probability that, for a randomly chosen (manipulator,victim) pair, the manipulator can attract traffic destined for the victim from at least 10% of the ASes in the internetwork. The manipulator uses the

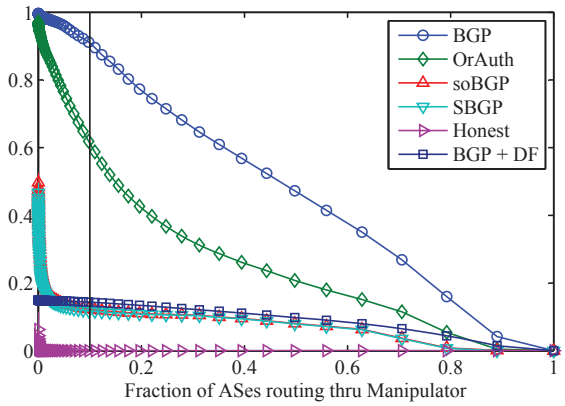

Figure 3: CCDF for the "Shortest-Path Export-All" attack strategy.

"Shortest-Path Export-All" attack strategy. The first four bars on the left assume that network does *not* use defensive filtering. We show the success of the manipulator's strategy on each of the four BGP security variants, in a network with and without defensive filtering of stubs. The horizontal line in Figure 2 shows the fraction of attacks that are completely eliminated by defensive filtering; since 85% of ASes in the CAIDA graph are stubs, properly-implemented defensive filtering guarantees that only 15% of manipulators can successfully attack any given victim.

Despite the fact that we used *sub-optimal* strategies for the manipulator, we have two concrete observations:

1. Even if we assume the manipulator runs the sub-optimal "Shortest-Path Export-All" attack strategy on a network that has S-BGP but not defensive filtering, he can still attract 10% of the ASes in the internetwork with probability $> 10\%$. Furthermore, more clever strategies for S-BGP (*e.g.,* Figure 9 and 10) might increase the manipulator's probability of success to the point where defensive filtering *alone* performs even better than S-BGP alone.

2. Even if both S-BGP *and* defensive filtering are used, there is still a non-trivial 2% probability that the manipulator can attract 10% of the ASes in the internetwork. Better attack strategies could increase this probability even further. This is particularly striking when we compare with the normal case, where the manipulator manages to attract 10% of the ASes in the internetwork with about 10^{-4} probability (not shown).

4.3 Attack strategy on different protocols

The reader may wonder why we chose to focus specifically on the probability of attracting 10% of the ASes in the internetwork in Figure 2. In the interest of full disclosure, we now present the full picture:

Figure 3: We show the complimentary cumulative distribution function (CCDF) of the probability that at least a x-fraction of the ASes in the internetwork forward traffic to the manipulator when he uses the "Shortest-Path Export-All" attack strategy. Probability is taken over the uniform random choice of a victim and manipulator, and observe that Figure 2 simply presents a crosssection of these results at the x-axis value of $x = 10\%$. We briefly highlight a few details about this figure:

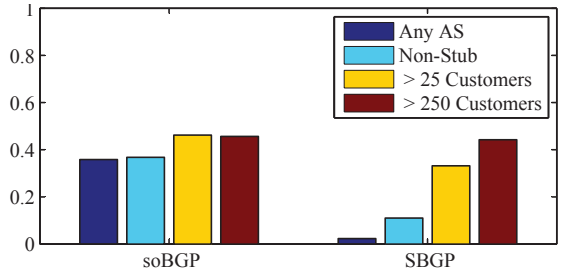

Figure 4: Probability of finding a shorter path.

BGP curve. Here, the manipulator *originates*, *i.e.,* announces that he is directly connected to, the victim prefix. This curve looks almost like the CCDF of a uniform distribution, since the manipulator and the victim both announce one-hop paths to the prefix, and are thus are about equally likely to attract traffic.

Origin Authentication curve. This time the manipulator announces that he has a direct link to the AS that legitimately owns the victim prefix. Because the manipulator's path is now two hops long, the amount of traffic he can attract on average is reduced.

soBGP and S-BGP curves. For the attack on soBGP, the manipulator announces the shortest path that *exists* in the AS graph. For the attack on S-BGP (and data-plane verification), the manipulator announces the shortest *available* path that he learned from his neighbors. Oddly, the soBGP and S-BGP curves are almost identical, despite the fact that S-BGP provides stronger security guarantees than soBGP (see also Section 4.4).

Honest curve. Here the manipulator behaves 'normally', *i.e.,* using the ranking and export policies of Section 2.2.

BGP+Defensive Filtering curve. Defensive filtering eliminates all "Shortest-Path Export-All" attack strategies on BGP by stubs, *i.e.,* by 85% of ASes. Thus, this is approximately 'BGP' curve scaled down to 15%.

Different-sized ASes are equally affected. This paper consistently measures the manipulator's success by *counting the number of ASes* that route through him as a result of his attack strategy. We also produced versions of Figure 3 that count the fraction of ASes *of a given size* that route through the manipulator: (a) All ASes, (b) ASes with at least 25 customers, and (c) ASes with 250 customers. We omit these graph as they were almost identical.

4.4 S-BGP forces long path announcements

Figures 2 and 3 show that S-BGP is *not* much more effective in preventing "Shortest-Path Export-All" attack strategies than the less-secure soBGP. To understand why, let's compare the lengths of the path that the manipulator can announce with soBGP and S-BGP:

Figure 4: We show the probability that the manipulator can announce a path that is shorter than the *normal path*, *i.e.,* the path he would have chosen if had used the rankings in Section 2.2. Probability is taken over a randomly-chosen victim, and a manipulator that is randomly chosen from one of the following four *classes*: (a) Any AS in the graph, (b) *Non-stubs*, or ASes with at least one customer (c) Medium-sized ASes with at least 25 customers, and (d) Large ASes with at least 250 customers. If we focus on the results for

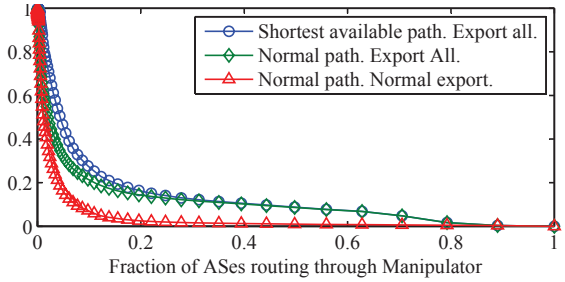

Figure 5: Aggressive export policies.

S-BGP, it is clear that *larger* ASes are more likely to find *shorter* paths through the network; this follows from the fact that these ASes are both more richly connected (*i.e.,* they have large degree), as well more central (*i.e.,* they are closer to most destinations in the internetwork). Furthermore, we can also see that ASes (especially small ASes) are more likely to find short paths with soBGP than they are with S-BGP.

From Figure 4, we can conclude that S-BGP is doing exactly what it is designed to do: it is limiting the set of paths the attacker can announce, thus forcing him to announce longer paths. However, in light of the results in Figures 2-3, we must ask ourselves why forcing the manipulator to announce longer paths does not seem to significantly limit the amount of traffic he attracts. We could explain by arguing that path lengths in the Internet are fairly short, (averaging about 5 hops in our simulations); so the paths that the manipulator can get away with announcing in soBGP are only a few hops shorter than the paths he can announce with S-BGP. Indeed, as we show in the next section, the fact that AS paths are normally so short means that the *length* of the manipulator's path often plays less of a role than the *set of neighbors that he exports to.*

4.5 Export policy matters as much as length

We now show that the attacker's export policy is as important as the length of the path he announces:

Figure 5: We show another CCDF of the probability that at least a x-fraction of the ASes in the internetwork forward traffic to the manipulator; probability is taken over a randomly-chosen victim, and a manipulator chosen randomly from the class of ASes that have at least 25 customers. We consider three different strategies: (a) Announce the shortest available path to all neighbors (equivalent to the "Shortest-Path Export-All" attack strategy on S-BGP), (b) Announce the normal path to all neighbors, and (c) Announce the normal path using the normal (**GR2** and **NE**) export policy.

This figure shows that, on average, announcing a *shorter* path is much less important than announcing a path to more neighbors (*i.e.,* the curves for (a) and (b) are very close, while the curves for (b) and (c) are quite far apart). Indeed, when we considered smaller manipulators (not shown), the curves for (a) and (b) are even closer together. One way to explain the small gap between (a) and (b) is to note that the manipulator's normal path is very often also his shortest path (this holds for 64% of (manipulator, victim) pairs from this class); and even when it is not, his normal path tend to be quite short.

To understand the large gap between (b) and (c), we note that by violating the normal export policy, the manipulator can announce paths to his providers, even when his normal path is not through a customer. His providers are more likely to choose the customer path through the manipulator, over some possibly shorter, non-customer path.

4.6 Different sized manipulators and victims

Next, we would like to determine which ASes in the Internet are likely to be the most successful manipulators, or the most vulnerable victims. We consider ASes from four different classes: (a) All ASes (b) Non-stubs (ASes with at least 1 customer), (c) ASes with at least 25 customers, (roughly modeling "Tier 2 ASes"), and (d) Large ASes with at least 250 customers ("Tier 1 ASes"). In the interest of space, we only summarize our findings here. Graphs and detailed results are in the full version [1].

Manipulators. We make the surprising observation that (c) "Tier 2s" tend to be the most effective manipulators, attracting more traffic than even the (d) "Tier 1s". In fact, we found that in many cases, even smaller (b) non-stubs tend to attract more traffic than the "Tier 1s". Here we assume that the victim is chosen from the set of all ASes.

Victims. We found that (c) "Tier 2" ASes tend to be the least vulnerable to attacks. Furthermore, when we considered attacks on soBGP or S-BGP, we make the surprising observation that the (d) "Tier 1" ASes are even more vulnerable than (a) smaller ASes at the edge of the internetwork. Here the manipulator is chosen from the set of all ASes.

One might expect Tier 1 ASes to attract more traffic than other classes of ASes, but these results indicate that this is not the case; instead, Tier 2s tend to attract the most traffic. To see why, notice that while Tier 1s are more central (and thus have short paths to most ASes in the internetwork), they are also more *expensive.* That is, a Tier 1 is always a provider/peer of its neighbors, so even if those neighbors learn a short path through the Tier 1, they will prefer to route over a (potentially longer) path through one of their own customers. On the other hand, Tier 2s tend to be both central as well as the customer of large Tier 1 ASes, and therefore in the position to attract the maximum amount of traffic. Thus, these results again follow from the fact that creating *customer* paths is often more important than creating *short* paths.

4.7 Summary

In some sense, the results of this section suggest that secure routing protocols like S-BGP and soBGP are only dealing with one half of the problem: while they do restrict *the path* the manipulator can choose to announce, they fail to restrict his *export policies.* Indeed, because defensive filtering restricts both the export policies and the paths announced by stubs, we find that it provides a level of protection that is at least comparable to that provided by S-BGP, and even data-plane verification, alone.

Even if we eliminate attacks by stubs via defensive filtering, we found that the internetwork is still vulnerable to non-stub ASes that both (a) deviate from normal routing policies by announcing shorter paths, and (b) deviate from normal export policies by announcing non-customer paths to *all* their neighbors. Furthermore, we have seen that it is exactly these non-stub ASes (and in particular, the Tier 2s) that are in the position to launch the most devastating attacks. The success of these attack strategies can be limited with soBGP, S-BGP, or data-plane verification.

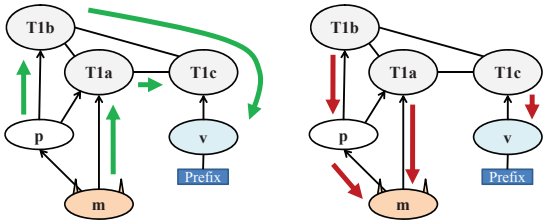

Figure 6: (a) Normal outcome. (b) Blackhole.

5. SMART INTERCEPTION ATTACKS

We now turn our attention to traffic interception attacks [2, 3, 5]. In an interception attack, the manipulator would like to attract as much traffic as possible to his network (in order to eavesdrop or tamper with traffic) before forwarding it on to the victim IP prefix. Thus, we require that an interception attack *preserves an available path from the manipulator to victim.*

5.1 A stub that creates a blackhole

To provide some intuition, we first show how a manipulator could lose a working path to a victim:

Figure 6: For simplicity, let's consider an attack on BGP where the manipulator falsely originates the victim's prefix. The manipulator m is a web-hosting company in Illinois, and wants to attract traffic destined for the victim v a web-hosting company in France. The manipulator is a multi-homed stub with two providers, a Tier 1 AS $T1a$, and a Chicago-area telecom provider p. The left figure shows the normal outcome, where the manipulator has a path to victim available through each of his providers. The right figure shows what happens when the manipulator announces the victim's prefix to each of his providers; since each of them prefer short customer paths, they will forward their traffic through the manipulator. The manipulator has now created a *blackhole*; he has no available path to the victim v through either of his providers.

5.2 When do interception attacks succeed?

The reader may be surprised to learn that there are many situations in which blackholes are *guaranteed not to occur.* We can prove that, within our model of routing policies, the manipulator can aggressively announce paths to certain neighbors while still preserving a path to the victim:

THEOREM 5.1. *Assume that* ***GR1*** *holds, and that all ASes use the routing policies in Section 2.2. Suppose the manipulator has an available path through a neighbor of a type x in the normal outcome. If there is ✓ in entry (x, y) of Table 1, then a path through that neighbor will still be available, even if the manipulator announces* any *path to* any *neighbor of type y.*

The full version [1] presents the proofs. We also not that the results marked with ✓* hold even if the internetwork does not obey **GR1**. We also observe that this theorem is 'sharp'; if there is an X in entry (x, y) of Table 1, we show by counterexample that the manipulator *can sometimes* lose an available path of type x if he announces certain paths to a neighbor of type y. Indeed, Figure 6 is a counterexample that proves the X in the lower-right entry of Table 1.

Results of this form were presented in an earlier work [2]. However, [2] claims that a peer-path *cannot* be lost by announcing to a provider (and vice versa). In the full version [1] we present an example contradicting this, that proves the remaining X entries in Table 1.

To preserve a path of type...	May announce to neighboring... Customers	Peers	Providers
Customer	✓*	✓*	✓
Peer	✓*	✓*	X
Provider	✓	X	X

Table 1: Guidelines for interception.

Tier 1s and Stubs. Theorem 5.1 leads to a number of observations, also noted by [2]. First, interception is easy for Tier 1s. Since Tier 1s have no providers, they need only concern themselves with the four upper-left entries in Table 1, which indicate that they can announce paths to all their neighbors. Secondly, interception is hard for stubs. A stub's neighbor is always a provider, putting it in the bottom-right entry of Table 1, indicating that aggressive announcements could cause a blackhole (*e.g.,* Figure 6).

5.3 When do "Shortest-Path Export-All" attack strategies cause a blackhole?

The observations of Section 5.2 are borne out by our experiments. Recall that in the "Shortest-Path Export-All" attack strategy, the manipulator announces his shortest (non-rejected) to *all* of his neighbors. We now show that this simple attack strategy often allows the manipulator to intercept traffic without creating a blackhole:

Figure 7: We show the probability that the manipulator has some available path to the victim if he uses the "Shortest-Path Export-All" attack strategy for each of the four BGP security variants. We present results for a randomly-chosen victim, and a manipulator chosen from the usual four classes (see Figure 4). We assume that manipulator runs the "Shortest-Path Export-All" attack strategy on each BGP security variant. We can make a number of observations:

1. Manipulators with the *most* customers are *least* likely to create a blackhole. As discussed in Section 5.2, these manipulators are most likely to have an available customer path to the victim, and as shown in the first row of Table 1, can get away with announcing to *all* their neighbors without creating a blackhole.

2. The attack on BGP is most likely to cause a blackhole (*cf.,* the attack on origin authentication, or soBGP). Because the manipulator announces a more attractive (*i.e.,* short) path, he is more likely to convince *all* of his neighbors to forward traffic to him, and thus create a blackhole.

3. The "Shortest-Path Export-All" attack strategy on S-BGP, *never* creates a blackhole (as long as the manipulator had a path to the victim in the normal outcome). This observation matches intuition; since S-BGP forces the manipulator to announce an available path, the manipulator must of course have an available path to the victim.

5.4 Two interception strategies

Figure 7 immediately suggests a simple interception strategy that seems to work every time:

"Shortest-Available-Path Export-All" attack strategy: The manipulator should announces his shortest *available* path from the normal outcome to all his neighbors.

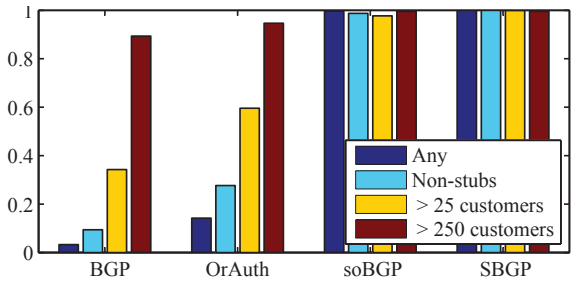

Figure 7: Probability that the "Shortest-Path Export-All" attack strategy does *not* create a blackhole.

Recall that this is exactly the "Shortest-Path Export-All" attack strategy on S-BGP.

Figure 3, shown that this strategy attracts more traffic than the normal strategy, but also suggests that when the network does *not* use S-BGP, there may be better interception attack strategies. Indeed, Figure 7 shows that there is a non-trivial probability that the manipulator has an available path to the victim, even if he launches the "Shortest-Path Export-All" attack strategy on the BGP. This suggests the following two-phase strategy:

"Hybrid Interception" attack strategy: First, run the "Shortest-Path Export-All" attack strategy on the secure routing protocol, and check if there is an available path to the victim. If no such path is available, announce the shortest path that was available in the normal outcome to all neighbors.[4]

By no means do we believe that these two strategies are optimal; indeed, while we evaluated more clever attack strategies, we omitted them here in the interest of brevity and simplicity. What is surprising is that even these trivial strategies can be quite effective for certain manipulators.

5.5 Evaluating interception strategies

From the discussion above (Figures 6 and 7, Section 5.2), it is clear that ASes with very few customers are unlikely to attract large volumes of traffic without blackholing themselves. For this reason, we focus our evaluation on manipulators with at least 25 customers, and for brevity only present attacks on BGP:

Figure 8: This is a CCDF of the probability that at least a x-fraction of the ASes in the internetwork forward traffic to the manipulator, under the assumption that the network uses BGP. We compare the (a) "Shortest-Path Export-All" attack strategy where the manipulator *is allowed to create a blackhole* (and thus tends to attract more traffic than the interception strategies above), with (b) the two interception strategies above, as well as (c) the normal strategy. Our key observation is that the "Hybrid Interception" attack strategy intercepts a large fraction of traffic; *e.g.,* at least 10% of the ASes in the internetwork with probability over 50%!

[4]We note that while this strategy will attract at least as much traffic as the "Shortest-Available-Path Export-All" attack strategy, the manipulator stands a higher chance of getting caught if he creates a blackhole in the first phase of the strategy.

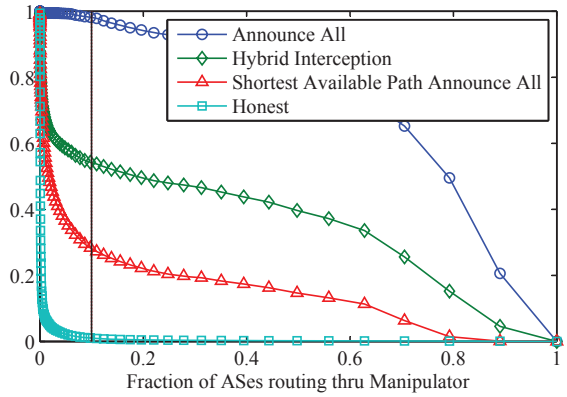

Figure 8: Interception attacks on BGP.

6. SMART ATTACKS ARE NOT OPTIMAL

We now prove that the "Shortest-Path Export-All" attack strategy is *not* optimal for the manipulator. We present three surprising counterexamples[5], found in CAIDA's AS graph and then anonymized, that show that (a) announcing *longer* paths can be better than announcing shorter ones, (b) announcing to *fewer* neighbors can be better than to announcing to more, and (c) the *identity* of the ASes on the announced path matters, since it can be used to *strategically trigger BGP loop detection*. In fact, (c) also proves that announcing a longer path can be better than a prefix hijack (where the manipulator originates a prefix he does not own)!

6.1 Attract more by announcing longer paths!

Our first example is for a network with soBGP, S-BGP or data-plane verification. We show a manipulator that *triples* his attracted traffic by announcing a *legitimate path to the victim, that is* not *his shortest path.* (This contradicts the optimality of the "Shortest-Path Export-All" attack strategy, which requires announcing shortest paths.) In fact, this strategy is so effective, that it attracts almost as much traffic as an aggressive prefix hijack on unmodified BGP!

Figure 9: The manipulator m is a small stub AS in Basel, Switzerland, that has one large provider $a1$ that has almost 500 customers and 50 peers, and one small provider AS $a2$ in Basel that has degree only four. The victim is European broadband provider v with over 100 customers and 26 peers.

Prefix hijack. In a network with (unmodified) BGP, the manipulator could run a simple prefix hijack, announcing "m, Prefix" to both his providers, and attract traffic from 62% of the ASes in the internetwork (20550 ASes), including 73% of ASes with at least 25 customers, and 88% of ASes with at least 250 customers. However, this strategy both creates a blackhole at the manipulator, and fails against soBGP or S-BGP.

Naive strategy. The upper (green) figure shows the "Shortest-Path Export-All" attack strategy, where the manipulator naively announces a *three-hop* available path, (m, $a1$, v, Prefix) to his provider $a2$. Since ASes $a2$ and $a3$ prefer the customer path that leads to the manipulator, over their existing peer paths, both will forward traffic to the manipulator. He intercepts traffic from 16% of the ASes in

[5]Each example was chosen to contradict the optimality of one aspect of the "Shortest-Path Export-All" attack strategy.