TOWERDEFENSE: Deployment Strategies for Battling against IP Prefix Hijacking

Tongqing Qiu*, Lusheng Ji†, Dan Pei†, Jia Wang† and Jun (Jim) Xu*
*Georgia Institute of Technology †AT&T Labs – Research

***Abstract*—IP prefix hijacking is known as one of the top security threats targeting today's Internet routing infrastructure. Several schemes have been proposed to either detect or mitigate prefix hijacking events. However, none of these approaches is adopted and deployed in large-scale on the Internet due to reasons such as scalability, economical practicality, or unrealistic assumptions about the collaborations among ISPs. As a result, there is lack of actionable and deployable solutions for dealing with prefix hijacking.**

In this paper, we study key issues related to deploying and operating an IP prefix hijacking detection and mitigation system. Our contributions include (i) deployment strategies for hijacking detection and mitigation system (named as TOWERDEFENSE): a practical service model for prefix hijacking protection and effective algorithms for selecting agent locations for detecting and mitigating prefix hijacking attacks; and (ii) large scale experiments on PlanetLab and extensive analysis on the performance of TOWERDEFENSE . We demonstrate that, by using only a few agents, TOWERDEFENSE can detect and mitigate prefix hijacking with up to 99.8% and 98.2% success ratios respectively.

I. INTRODUCTION

IP Prefix Hijacking attacks threaten the Internet's routing infrastructure. Fundamentally, the inherent assumption of self-policing and trust among participants of BGP [1], the inter-domain routing protocol responsible for exchanging routing information among thousands of Autonomous Systems (ASes) in order to route the traffic globally, opens up the possibility for false route announcements to infiltrate the routing infrastructure. When conducting prefix hijacking, a malicious or misconfigured BGP router (called *hijacker* or *attacker*) either originates an AS path announcement for an IP prefix not owned by the router's AS or announces for an IP prefix (called *target prefix*) an AS path consisting of nonexistent links. Such false announcements render the misbehaving router's AS very attractive for forwarding traffic towards the target IP prefix. Lacking effective means to verify the accuracy and authenticity of such route announcements, ASes that receive such BGP updates may accept and propagate the false route, as well as subsequently forward traffic destined to the target prefix according to the false path. As a result, affected data traffic is diverted, or "hijacked", to ill-intentioned locations, causing performance degradation, service outage, and security breach for the victim prefix.

The importance of defending against IP prefix hijacking is well recognized by both industry and research communities, and many solutions [2]–[23] have been proposed in order to prevent, detect, locate, or mitigate IP prefix hijacking. However, the aforementioned solutions only solve parts of the problem and a critical step is still missing towards an operational deployment on the Internet. They either require changes to current routing infrastructures (e.g., router software, network operations), and/or public key infrastructures, or are compatible with existing routing infrastructures but lack of practical deployment strategies. Since changing the routing infrastructure usually involve more efforts and collaborations among ASes, we aim at making the existing solution deployable by bridging the gap in the need of practical deployment strategies. In particular, we believe the following two key issues need to be addressed in an operational deployment of any existing scheme on the Internet: (i) who should deploy and operate a system that can detect and mitigate hijacking attacks, and (ii) how to deploy the such kind of system (e.g., how to strategically place agents for detection and mitigation of hijacking attacks). Instead of proposing yet another new detection and mitigation scheme, this paper systematically examines these issues and propose two practical deployment strategies that can be used together with existing detection and mitigation schemes for battling against prefix hijacking attacks.

Our first deployment strategy is a new service model in which the service providers in particular the ISPs and CDN providers can deploy and operate a prefix hijacking detection and mitigation system for protecting their customers, due to the following reasons. First of all, the service provides do have strong incentive to operate such a detection and mitigation system. Discussions with the ISP operators indicate that customers often blame their providers if their traffic are hijacked. Hence a direct result of a customer's prefix being hijacked is that its service provider's reputation and even revenue are in jeopardy. Second, a service provider usually has more resources than any of its customers for operating such a system. It is also possible that the service providers

offer hijacking protection as an enhanced service for their customers. On the other hand, customers often trust their providers much more than any other third party for battling against hijacking attacks since they are already buying transit service from their providers.

Our second deployment strategy includes two principles for agent placement based on existing prefix hijacking defense mechanisms. *Detection principle*: to effectively detect a particular prefix hijacking attack, the detection system needs to have agents deployed in the region within which the routers are "polluted" with false route entries injected by the attack. This is because only routing information that these agents gather, from either control or data plane or both, may contain attacker altered routes, the critical information any detection mechanism depends upon. *Mitigation principle*: to effectively mitigate a hijack, traffic to target prefix can be detoured towards pre-deployed relaying agents in order to avoid the polluted region of an prefix hijacking attack. For a given hijacking event, the "polluted" region highly depends on the topological and routing policies used on the Internet as shown in [24] (We will discuss the differences between our study and [24] in Section 4.1.1). While previous detection/mitigation proposals often evaluated their approaches with Internet topology, the agent selection problem has not been systematically studied: given the locations of hijackers are not known prior to the attacks, where to strategically deploy route information gathering agents and relaying agents to effectively detect and mitigate attacks. We show in the paper that the agent location placement problem is NP, and propose effective greedy algorithms for it.

We observe the problem of deploying and operating a prefix hijacking protection system is similar to that of a popular strategy computer game genre "Tower Defense" [25], appearing in best-selling game titles such as StarCraft, Age of Empires, and WarCraft. In such a game, a player needs to wisely choose the types, numbers, and locations of its guard towers to deploy based on the cost budget, tower capability, and possible enemy movements in order to win the battle against enemy offenses. Similarly, in the battle against prefix hijacking attacks, one also need to decide how many agents of each type are needed to detect and mitigate hijacking attacks on a set of prefixes that are needed to be protected, and where to deploy these agents to achieve desire protection coverage under certain resource constrains. This is why we name the aforementioned strategies as TOWERDEFENSE and the system built by following TOWERDEFENSE strategies as TOWERDEFENSE system. For the same reason, the deployed agents are sometimes called "towers" in this paper.

To evaluate performance of TOWERDEFENSE , we conducted extensive analysis and large-scale experiments on PlanetLab to show that on a topology like today's Internet by using only a small number (i.e. 6) of vantage points (where agents were deployed), TOWERDEFENSE if deployed by a service provide, can detect and mitigate prefix hijacking targeted at its customers with up to 99.8% and 98.2% success ratios respectively.

To highlight the practicality of TOWERDEFENSE we show through a case study of one Tier-1 ISP that (i) high detection/mitigation ratios can be achieved also through adding an even smaller number of new vantage points (which a service provider can obtain by buying transit from other ISPs) to the service provider's existing vantage point infrastructure, and (ii) even when 800 customers of the ISP sign up for the TOWERDEFENSE service gradually, the number of vantage points remains small ($\sim$20).

The rest of the paper is organized as follows. Section II gives an overview of the TOWERDEFENSE strategies. Section III presents the detailed methodology for vantage point selections for detection and mitigation purpose. Then we analyze the selection results based on extensive simulations in Section IV. Section V evaluates the performance of TOWERDEFENSE on Planetlab. Section VI briefly surveys related works and we conclude in Section VII.

II. TOWERDEFENSE FRAMEWORK

A. Service Model

We believe that *protection against prefix hijacking* is most suitable to be offered by service providers in particular ISPs and CDN providers to their existing customers.

Firstly, since the protection service is provided by an entity that a customer is already buying other services (e.g. communications, content hosting, etc) from, the customer likely has more confidence and convenience to subscribe from them than from any new third parties.

Secondly a major issue in deploying a new service is cost. In this aspect, service providers are positioned far better than other potential parties because of their existing infrastructures. A CDN service provider may have already deployed its servers at a large number of locations ranging from dozens to thousands of ASes. All these locations can potentially be used as vantage points for prefix hijacking protection. For ISPs, firstly, a large ISP (e.g. tier-1 ISPs) may already own a few ASes spanning large geological area; secondly, an ISP is aware of the routes used by its neighboring ASes because its border routers have established BGP sessions with the neighbors; and thirdly, if the identified vantage point location (say AS T) is far away, an ISP can make up the capability simply purchasing a connectivity from AS T as a BGP customer and connect its prefix hijacking

protection equipments with the border router which runs BGP session with AS T.

Moreover, although the service is offered for protecting customers of the service provider, in fact what gets protected are the inbound traffic paths towards the networks of these customers. If a hijacker can only hijack traffic from regions that has very little traffic for the target network, this hijacking is as good as non-effective. Thus knowing who communicate with the protected networks gives tremendous advantage for whoever offers the protection. This is exactly where ISPs and CDN providers have extensive knowledge.

B. Prefix Hijacking Protection

When a hijacker launches its hijacking attack against a target network, using a BGP router in its AS the hijacker spreads out false route announcements for the target prefix. Upon receiving such route announcements, some routers may accept the false routes and subsequently propagate to their neighbor routers while others may ignore such announcements. As a result, a portion of the Internet is *polluted* by the false routes announced by the hijacker. In the polluted region, routers now use the hijacker's false routes for forwarding packets addressed for the target prefix. In other words, any traffic that originates from or passes through the polluted region are now "hijacked".

Because typically only a portion of the Internet is polluted, an attack can only be detected if there are detection agents deployed in or right at the boundaries of the polluted region so that they can gather information regarding the false route for detecting anomalies. Because the location and size of the polluted region of an attack vary depending on the locations of both the hijacker and target network, it is important to study where to place such detection agents to achieve optimum detection ratio for all possible hijacker locations.

Similarly, it is important to study where to deploy agents that may assist in mitigating prefix hijacking attacks. Different from mitigation approaches such as [14] which are aiming at correcting the false routes, we believe that a traffic redirection approach (e.g. IP tunneling and DNS-based redirection [26], [27]) may be more desirable because it can potentially react very rapidly. Also this approach can be applied by a wider range of providers, not only by those who are deeply vested in BGP operations.

For mitigating a hijack, there can actually be two types of redirections, which we refer to as *reflecting* and *mirroring*. When a reflector r is used in mitigation against a hijacking event on the target d, traffic from a source s destined to d will be re-routed to r and then from r to d. [1] On the other hand when a mirror m is used in mitigation, traffic from s to d will be re-directed to m, and m will function as a mirroring site of d and respond to incoming traffic in the same way as d does.

An AS r can be used as a reflector site for s-d during a hijacking event only if both the path from s to r and the path from r to d are not polluted by the hijacking event. In addition, because the hijacker may know who the reflector r is, the path from s to r must not be polluted by hijacking event launched by the same hijacker on r either. On the other hand, the requirement for an AS m being used as a mirror for mitigating hijacking event on target d is that the path from s to m is not polluted by the hijacking event on d and the path from s to m is not polluted by the hijacking event on m. Although the requirement for a mirror site is more relaxed than reflector site, mirrors tend to be more expensive because they need to replicate contents. In addition, mirrors are better for less frequently changed contents.

Here again the key for a successful hijack mitigation service is to place the mitigation agents, reflectors or mirrors, at strategically important locations so that they can mitigate the most attacks for the most sources of the target network. Hence, in this paper we mainly focus on placement strategy for detection and mitigation agents, which we call *towers*.

III. Methodology

In this section, we describe the methodology for a service provider to strategically select locations for its detection towers and mitigation towers to defend its customers against hijacking attacks. Because prefix hijacking is targeting inter-domain routing infrastructure, we consider ASes being the basic element. That is, we refer to hijackers or towers as hijacker ASes or tower ASes while it is understood that the actual hijacker or tower is really one or more machines (e.g. BGP routers, server, etc) within the corresponding AS.

Tower location selection involves evaluating many imaginary hijacking scenarios in the Internet AS topology, and assessing whether ASes may be impacted by the attacks. A service provider can infer Internet AS topology from publicly available BGP tables and updates such as Route Views [28] and RIPE [29]. We leave the discussion on the impact of the well-known topology incompleteness to Section IV-D.

If an AS prefers a fake path to d announced by a hijacker h over the AS' current legitimate path to d, this AS is impacted/polluted by the hijacking. Subsequently not only will this AS propagate the fake path to its neighbors, which in turn determine if they prefer the fake path, any future traffic destined for d passing

[1] We assume that the reverse traffic path from d to s is not subject to the same hijacking.

through the impacted AS is hijacked. In evaluating hijacking scenarios, the selection algorithm determine AS path preference based firstly on inter-domain routing policies, then preferring shorter AS path, and finally using random selection to break any remaining ties. Two widely adopted inter-domain routing policies are "prefer customer routes" and "valley-free routing" [30]. That is, while forwarding traffic an AS always prefers to forward using a link to its customer over a link to its peer over a link to its provider. Moreover, after traversing a provider-to-customer link or a peer link, a path will not traverse another customer-to-provider link or another peer link.

A. Detection Tower Selection

TOWERDEFENSE can employ existing detection mechanism [2], [16], [17], [19], [20], [31] for detecting hijacking events. While the actual detection methods differ by these approaches, they generally require the presence of detection agents in impacted ASes. Thus to keep our evaluation method general, we assume that if the service provider has at least a detection tower deployed in one of the impacted ASes, the hijacking event can be detected.

Therefore the detection tower position selection problem can be formulated as the following. Given a customer prefix d and a set of candidate detection tower locations V_c, we need to find the minimum subset V_d of V_c that the detection towers v in V_d can detect as many as possible hijacking events targeting a customer prefix d. If the candidate set contains all ASes on the Internet, the problem does become a classic *set cover* problem, which is NP hard. But in reality, the set of candidate locations is limited thus the detection tower selection. Therefore, to select the detection tower is at least as hard as to solve the set cover. We adopt a greedy algorithm similar to that for set cover problem to solve this problem.

More specifically, the undetected hijacker AS set H_u was first initialized to all possible hijacker ASes set H for hijacking d and the selected detection tower set V_d is empty. In each iteration, we select a detection tower v from candidate set V_c that can detect the most hijackers H_v from the undetected set H_u and move it out of the candidate set V_c into the selected detection tower set V_d. At the same time, we update the set of undetected hijacker AS set H_u by taking out the hijacker ASes that v can detect. The selection process can be terminated either after a fixed number of detection towers are selected (up to all candidate ASes) or after the gain in the detection coverage by adding a new detection tower becomes marginal (e.g., below a given threshold). More formally, we define *detection effectiveness* $DE(v,d)$ of a detection tower v against hijackers attacking d as $DE(v,d) = |H_v|/|H|$. Then the *detection effectiveness of a subset of detectors* V_d is:

$$\mathcal{DE} = |\bigcup_{v \in V_d} H_v|/|H|$$

The above greedy algorithm is to maximize the detection effectiveness.

B. Mitigation Tower Selection

Similar to detection tower selection, mitigation tower selection is a variant of set cover and can be done by a very similar greedy algorithm with one difference, the criteria for picking one candidate mitigation tower location over the others during each iteration. We first define *mitigation effectiveness* of a mitigation tower m against an *individual* hijacker h attacking d as the following:

$$\mathcal{ME}_I(h,m,d) = \frac{|MS(h,m,d)|}{|S(h,d)|},$$

where $S(h,d)$ is the set of d's sources whose traffic will be hijacked by h and $MS(h,m,d)$[2] is the subset of sources of $S(h,d)$ that m can mitigate. Then we define the *mitigation effectiveness against a set of hijackers* of a mitigation tower as:

$$\mathcal{ME}_S(m,d) = \sum_H \mathcal{ME}_I(h,m,d)/|H|,$$

where H is the set of hijackers in question. Until now, we defined the mitigation effectiveness of one vantage point.

The mitigation tower selection algorithm, which tries to maximize the mitigation effectiveness, is now described as follows. Initially, the unmitigated hijacker AS set H equals to all possible hijacker ASes for hijacking d and the selected mitigation tower set M_d is empty. In each iteration, we select a mitigation tower m from candidate set M_c that has the highest mitigation effectiveness against hijackers in H and move it out of the candidate set M_c into the selected mitigation tower set M_d. At the same time, we update the mitigation effectiveness for each mitigation tower in remaining candidate set M_c. The selection process can be terminated either after a fixed number of mitigation towers selected or after the gain in the mitigation coverage by adding a new mitigation towers becomes marginal.

C. Remarks

The detection tower and mitigation tower selection algorithms described above have implicitly assumed that all ASes have equal probability hosting hijacker and all sources are equally important. In reality, better knowledge regarding both aspects may be available and the

[2]Although we do not explicit distinguish reflectors from mirrors here, obviously in actual computation the $MS(h,m,d)$ of an mitigation AS used as a reflector will be different from that of as a mirror.

algorithms can be enhanced accordingly by applying different weighting factors, which are based on hijacker hosting probability and source importance, for different hijackers and sources when selecting the "best" candidate during each iteration.

IV. Analysis Results

In this section we evaluate the effectiveness of the detection and mitigation selection methods proposed in Section III, by exhaustively simulating hijacking events on the AS level topology of the Internet with all possible locations of hijacker ASes and target ASes.

A. AS Resilience

In our experiments, we construct the AS level topology graph using BGP tables and routing updates obtained from RouteViews and RIPE in 2008. The resulting AS topology has over 28K ASes. [3] Using the method proposed in [32], we classify them into five tiers. Stubs are the lowest tier ASes with only customer-to-provider links. There are 22856 stub ASes in total. On the other hand, 9 well-known ISPs [4] with no providers are classified as Tier-1 ASes. The rest ASes are classified into Tier-2, Tier-3, and Tier-4 based on their relationships (e.g., provider, customer, or peer) to other ASes. The number of Tier-2, Tier-3, Tier-4 ASes are 221, 2638, 3156 respectively.

Because AS relationship and AS path length are two of the key factors in BGP best path selection process, the impact of a hijacking event not only depends on locations of the target AS and the hijacker AS in the Internet topological hierarchy but also on the providers to which the target AS connects. We use the metrics "resilience of a target AS d" for quantitatively measuring the impact of hijacking events targeted at d. Assuming that there is an equal probability for where the hijacker may be on the Internet, we hence define the *resilience* of a target AS d as the *average* of the portions of unaffected-source-ASes for all possible hijacker locations. [5]

What is interesting about the resilience of a target AS is that it has opposite significance for detection service and mitigation service. For detection service, since only detection towers which are polluted by a hijacking attack will detect the hijacking event, the more resilient an AS is, the more difficult it is to find effective detection tower locations. On the other hand, the more resilient an AS is, the easier it is to find locations for mitigation towers because an important qualification for being a reflector/mirror location is that it is not hijackable by the same hijacker attacking the target.

We now look at the resilience of the stub networks which TowerDefense is aimed at protecting. We classify the stub networks based on the number of providers a stub network has. In our Internet AS topology, there are 22856 stubs in total, among which 12941 stubs have a single provider, 3897 have two providers, 1387 have three providers, and 4631 have more than three providers, respectively. Figure 1 shows the distribution of the resilience of the stub ASes of each class. We can see that the more providers a stub has, the more resilient it likely is. This observation is consistent with that in [24].

Different from [24], we further classify stub networks based on which tier their providers are in the Internet AS topology and study the impact of the providers' locations on the resilience of the stub networks. We use the single-provider stub networks to illustrate our findings because they are vast majority of the stub networks and most vulnerable to hijacking events. Multi-provider stub networks are more complicated to characterize because the providers often belong to different AS tiers (Due to the space limit, we do not report the results on them here). Among the 12941 single-provider stubs, 1812 are customers of Tier-1 ASes, 3876 are customers of Tier-2 ASes, 4313 are customers of Tier-3 ASes, and the rest are customers of Tier-4 ASes. Figure 2 shows the distribution of resilience values for each single-provider stub group.

We find that the resilience of the single-provider stub ASes highly depends on where their providers are in the Internet's AS hierarchy. The stubs connected to Tier-2 ASes are more resilient than those connected to ASes of other tiers. Tier-1 ASes are more likely being affected by a hijacking attack than Tier-2 ASes because the advertisements for a false route produced by a hijacking attack will appear as a customer route or a peer route to the Tier-1 ASes while for Tier-2 ASes such route often appear to be a provider route. Thus Tier-1 ASes are more likely to accept such false route advertisements than Tier-2 ASes. On the other hand, lower-tier ASes (i.e., Tier-3 and Tier-4 ASes) tend to have low resilience for two reasons: (i) they have fewer connections compared with Tier-2 ASes; (ii) paths reaching them tend to be longer because they often go through higher-tier ASes. Since Tier-3 and Tier-4 ASes are similar in resilience distribution, we group them together as *others* in the AS hierarchy from now on.

In summary, our resilience analysis offers two in-

[3] Note that inferred AS topology can be incomplete due to limited vantage points used in the data collection. We will show that missing links in the AS topology has minor impact on the performance of our detection and mitigation methods.

[4] Tier-1 ASes are AS1668 (AOL), AS7018 (AT&T), AS3549 (Global Crossing), AS3356 (Level 3), AS2914 (NTT), AS209 (Qwest), AS1239 (Sprint), AS701(Verizon), and AS3561 (SAVVIS).

[5] In this paper we assume equal probability distribution of both source and hijacker. The definition of resilience can easily be extended to more complex model if more data regarding for a particular target where its sources are distributed and the likelihood of having the hijacker at different location.

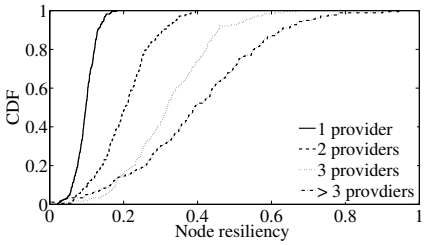


Fig. 1. The resilience of stub networks

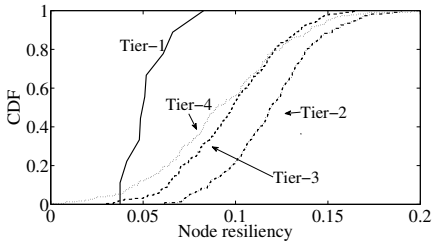


Fig. 2. The resilience of one-provider stub networks.

sights. (i) The more providers a stub network has, the more resilient it is. For such a stub network, it is relatively difficult to find effective detection tower locations but easy to find effective mitigation tower locations. (ii) Tier-2 ASes are more resilient than other tiers and so are stub customers of Tier-2 ASes. This makes Tier-2 ASes be good candidates for providing mitigation services, but not detection services.

1) Difference from Previous Resilience Study: Although our resilience study shares similar simulation approaches with [24], there are three major differences. First, the fundamental goals of the papers are different. [24] studies via simulation that, for individual AS, what kind of ASes are more resilient given the unpredictable hijacking events. Our goal is to select *multiple* vantage point ASes to detect and mitigate the hijacking events, requiring more algorithmic effort. Second, even for the resilience result, we provide more detailed results, e.g. we further classify the stub networks based on which tier their providers are in the Internet AS topology and study the impact of the providers' locations on the resilience of the stub networks. Finally, as shown in Section III-B, when studying mitigation towers we have to consider the impact of different sources (polluted ASes), ensuring that the traffic from multiple sources to the mitigation towers cannot be hijacked, which is not considered in [24].

B. Detection Results and Analysis

In this section we evaluate the detection effectiveness of a service provider who would like to offer TOWERDEFENSE service to its stub customers. Single-provider stubs and multi-provider stubs are analyzed separately because the former's results are easier to analyze. We run the detection tower selection algorithm presented in Section III for each TOWERDEFENSE service provider (X) and each of its stub customers as the target d. We compute the average detection effectiveness over d for each X, which is then averaged over service providers' locations in the AS hierarchy (Tier-1, Tier-2, and Others). In order to trade between number of detection towers selected and the detection effectiveness, the selection process is terminated after the gain in the detection effectiveness by adding a new detection tower becomes marginal (below 0.5%). Based on our results,

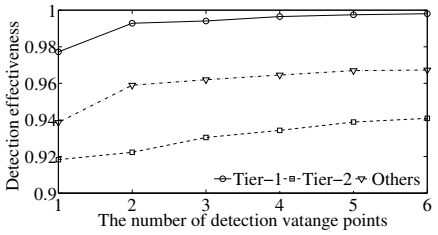


Fig. 3. Detection effectiveness for single-provider stubs as the number of detection towers increases.

we further summarize the guidelines about the detection towers selected by our greedy algorithm.

1) Single-Provider Stubs: Figure 3 shows the average detection effectiveness in *Tier-1*, *Tier-2*, and *Others* when increasing the number of detection towers. We make the following observations. (i) The first selected detection tower can cover a very high percentage (e.g. more than 93% in Tier-1) of hijackers. (ii) The gain on the effectiveness by adding additional detection towers becomes marginal after 4 detection towers are selected. (iii) None of the Tier-1 ASes is selected as the detection tower by any of the service providers. (iv) Tier-1 service providers achieve highest detection effectiveness (e.g. up to 99.8%) while Tier-2 service providers achieve the lowest. The third observation is consistent with our expectation based on the resilience-based analysis at the end of Section IV-A. Detailed analysis of the first two observations are provided below.

Which AS is selected as the first detection tower? Our greedy algorithm chooses the AS with the best detection effectiveness as the first detection tower. We use real examples from our simulation traces to illustrate the insights behind such selections in Figure 4. In Figure 4, there are three examples, one for a TOWERDEFENSE service provider at each tier: AS7018 for Tier-1, AS13249 for Tier-2, and AS2854 for Tier-3. The shaded node is the first detection tower selected by the greedy algorithm. d is one representative single-provider stub customer AS of the service provider X (the detection effectiveness of any other single-provider stub customer ASes of the same provider X is the same as that of d).

In Figure 4(a), AS3261 (a small ISP with some

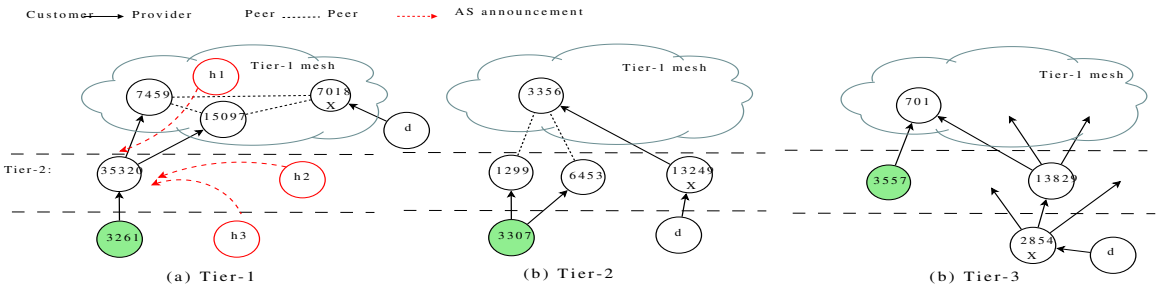


Fig. 4. Examples from simulation traces to explain tower selection for single-provider stub target d with provider at different locations (shown in the subfigure captions).

customers but only one provider AS35320) is chosen as the first detection tower for TOWERDEFENSE service provider Tier-1 AS7018. AS3261 can observe more than 96.1% of hijacking events targeted at d. This is mainly because its sole provider AS35320 (a Tier-2 AS) can be easily impacted by the hijacking event of target d, and then propagates the polluted path to AS3261. In addition, AS3261 can observe some other hijacking events if the attacker is a customer of AS3261, which AS35320 cannot observe.

Let us explain why AS35320 can be easily polluted now. AS35320 has two Tier-1 providers AS15097 and AS7459. It also peers with many (45) large Tier-1/Tier-2 ASes. Originally, AS35320 will choose the route AS35320 - AS7459 (or AS15097) - AS7018 to destination d. This original route is a provider route, which is less preferred than a peer route or a customer route, according to the BGP best path selection process. Therefore, AS35320 will be polluted if (i) the hijacker is Tier-1 provider of AS35320 (e.g. $h1$ in Figure 4) because the route AS35320-h1 is shorter than the original route to d; (ii) the hijacker is in its Tier-2 peers (e.g. $h2$ in Figure 4) because it prefers a peer route than a provider route; or (iii) the hijacker is in a lower-tier ASes (e.g. $h3$ in Figure 4) and the fake announcement reaches any of AS35320's peer/customer ASes.

In Tier-2 and Tier-3 cases shown in Figures 4 (b) and (c), AS3307 and AS3557 were first selected as the detection tower, respectively. They share two commonalities. First, the selected ASes will receive the provider route from the destination AS. Second, the selected ASes are either the Tier-2 ASes, or poorly connected to (with one or two connections) Tier-2 ASes.These commonalities are also observed on other detection towers selected by our algorithm.

Which ASes are selected after the first detection tower? Figure 3 shows that the second tower selected offers good improvement of detection effectiveness than towers selected later, especially for Tier-1 and Tier-2 cases. We now investigate the similarity between the towers selected first and second by our algorithm. We

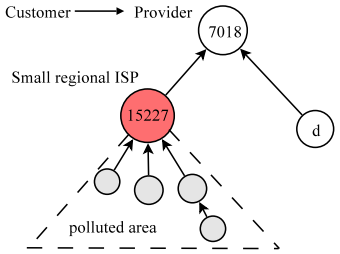


Fig. 5. Locally polluted example

define the term *Tier-2 peering set* of the tower, given the key role of Tier-2 ASes in detection effectiveness. If a Tier-2 AS is selected, then the Tier-2 peering set is the set of the ASes peering with this Tier-2 AS. Otherwise, the Tier-2 peering set is the set of ASes peering with the AS' Tier-2 provider(s)[6]. We compute the *Jaccard coefficient*[7] of Tier-2 peering sets of the first two selected towers to compute their similarity. The Jaccard coefficient for the first two selected towers on average is 0.18, with maximum 0.27; while the overall Jaccard coefficient for any two Tier-2 ASes on average is 0.46. This result indicates that the first two selected towers have significant different peering sets. In other words, they are diverse from each other.

Why does the coverage gain become marginal after a few detection towers? We noted that the coverage become stable after selecting first few detection towers. The reason is that some hijacking cases are difficult to detect, making it difficult to achieve 100% overall detection effectiveness, thus there is not much room for coverage increase from the already-high coverage provided by the first few selected towers. We investigated those hard-to-cover hijacking cases, and found that, generally speaking, these are *locally polluted cases*, where only several stub nodes are polluted by the hijackings. Figure 5 shows a real example of locally polluted

[6]This definition does not apply on Tier-1 AS since no Tier-1 AS was selected by our algorithm.

[7]The Jaccard coefficient measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

case. Hijacker AS15227, which has only one provider AS7018, advertises the prefix p belonging to the target stub AS d. AS7018 then has two equally good routes, both from customers and with the same path length of 1. Therefore, AS7018 has a 50% chance to select either path. In case it sticks to the original path learned from d, it will not propagate the fake announcement to other ASes. Therefore, only AS15227's direct or indirect customer ASes (the four gray nodes in the figure) are impacted in this case, and unless we have detection tower in these ASes, this hijacking will not be not detected.

2) Multi-Provider Stubs: Figures 6 (a), (b), and (c) show the detection effectivenesss for multi-provider stub customers of *Tier-1*, *Tier-2*, and *Others*, respectively. [8] We group multi-provider stub customers into three different groups based on the number of their providers. Then we compute the average effectiveness in each group varying the number of detection towers. We have the following observations. (i) The first few selected detection towers can cover a very high percentage of hijacking cases, and the effectiveness gain becomes marginal after more than 8 detection towers are selected. (ii) The more providers a stub customer has, the more detection towers are needed to achieve good detection effectiveness. (iii) The first detection tower is almost always a Tier-2 AS or its immediate customers which has only one or two providers. The second tower is also very diverse from the first tower. These observations are similar to those on the single-provider stubs.

How does the number of providers' impact the detection effectiveness? We can clearly see that the more providers a stub customer has, the more detection towers are needed. For example, as shown in Figure 6(b) the detection effectiveness for stub ASes with more than 3 providers appears to be only slightly over 80% though those for majority of stub customers range from 90% to 99%. To explain this, we define that for a given target stub customer d and a hijacker AS h, the *impact* of the hijacking event as the number of polluted ASes divided by the total number of ASes. We use the stub customers of AS7018 (a Tier-1 AS) as a case study to illustrate our findings. Figure 7 shows the average impact of a hijacker on the stub customers of a given number of providers. The hijackers on x axis are ranked in decreasing order of their average impact. We observe that (i) different hijacker ASes have different impacts on the same set of stub customers, and (ii) the more providers a stub customer has, the smaller impact a hijacker has, and hence the harder to detect with a small number of detection towers. These observation also suggest that it is more important to detect the hijacking events of bigger

[8]Note that a customer can be multi-homed to a Tier-1 AS and a Tier-2 AS, thus it will be considered in both Figure 6 (a) and (b).

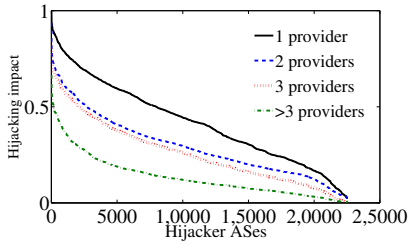


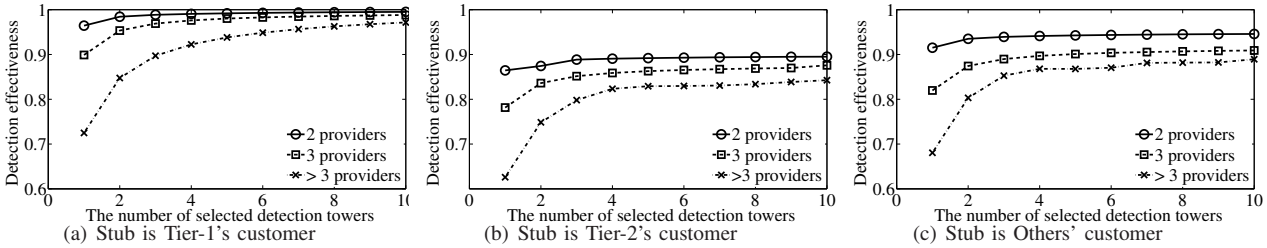
Fig. 7. The average impact of different hijacker ASes on stub customers of AS7018

impact than those with smaller impact. We normalize the detection effectiveness by the impact of hijacking events and find that the normalized detection effectiveness is always higher than 94% if using 8 detection towers. These numbers are much higher than those shown in Figure 6 specially for non-tier-1 multi-provider stub customers (please refer to [33] for details).

How does a multi-provider stub select the TOWERDEFENSE provider? For a multi-provider stub customer d, if two or more of its providers provide TOWERDEFENSE service, which provider should d choose? The answer is that d can choose any of its providers. The detection effectivenesss of using different providers for d are very similar because the detection towers are selected based on the same set of information (e.g., AS topology). Please refer to [33] for detailed discussion.

3) Detection Tower Selection Strategies: Based on our analysis results, we summarize the strategies on selecting detection towers for a given service provider X and a given target d. These guidelines help service providers not only understand the usefulness of existing vantage points, but also determine adding new vantage points. When the service provider has no complete AS topology or simply do not want to run our selection aglorithm, it can still choose the vantage points based on local topology information of the candidate vantage points according to the following strategeis.

1) Select v that has multiple providers and is connected to many peers such that v uses a provider or a peer route to reach as many targets as possible, making it easier to be polluted by the fake routes from peers or customers, respectively. Some (not all) well-connected tier-2 nodes satisfy this requirement.
2) Select v which is relatively far away from d so that AS path to d is more likely to be polluted by a shorter fake route.
3) Select the immediate poorly connected (e.g. single-provider) customer of v as the alternative.
4) Select v that is diverse from existing detection towers. For example, one should avoid selecting v in an AS which is directly connected to an already selected detection towers.



(a) Stub is Tier-1's customer (b) Stub is Tier-2's customer (c) Stub is Others' customer

Fig. 6. The detection effectiveness of multi-provider stub customers of Tier-1, Tier-2, and Others ASes

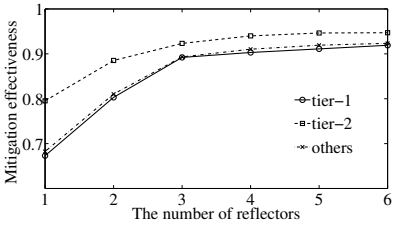


Fig. 8. single-provider, reflector case.

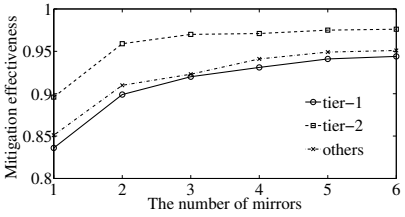


Fig. 9. single-provider, mirror case.

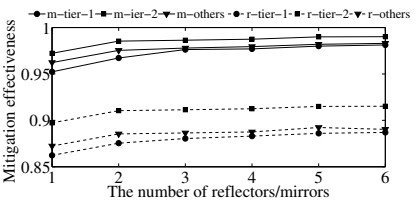


Fig. 10. Multi-provider, reflectors and mirrors.

C. Mitigation Results and Analysis

We now evaluate the algorithm described in section III.

1) Selection Results: Figures 8 and 9 show average mitigation effectiveness for single-provider stubs and Figure 10 shows those for multi-provider stubs. As expected, for both reflectors and mirrors, stub customers of Tier-2 ASes can be better mitigated (e.g. up to 98.2% in Figure 10) than stub customers of other tier ASes with the same number of mitigation towers. Mirror mitigation is always better than reflector mitigation because a successful mirror does not require the path from itself to the target d not to be polluted by hijacking events on d, but a successful reflector does.

To further illustrate the mitigation effectiveness difference between mirrors and reflectors, Figure 11 shows the mitigation effectiveness for single-provider stubs which are customers of Tier-1 ISPs when using $n(n = 2, 4, 6)$ mitigation points consisting of m mirrors $(m = 0, 1, ...n)$ and $n-m$ reflectors. We find that for all cases, the mitigation effectiveness increases as the number of mirrors increases. In addition, the curves tare close to each other when the same number of mirrors are used. This observation seems to suggest that the dominant mitigation effectiveness are contributed by mirrors in these mixed compositions. In other words adding reflectors to a mirror mitigation system has limited marginal benefit.

If the mitigation effectiveness is the only factor that is considered in the tower selection process, then our results suggest that mirrors should be used instead of reflector. However in reality there are other factors limiting the use of mirrors. First, it is more expensive to deploy a mirror than to deploy a reflector due to the

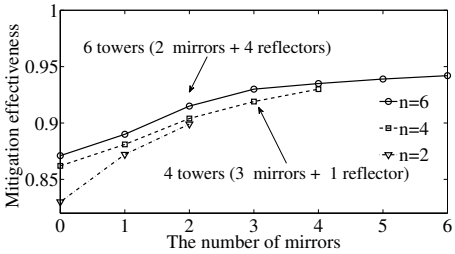


Fig. 11. Combining mirrors and reflectors. Single-home & tier-1 case.

extra system and network resources a mirror requires to serve customers directly from itself. Second, there are certainly more overhead on realtime synchronization of contents and meta-data. All these factors can be easily integrated into the mitigation selection algorithm so that an optimum combination of mirrors and reflectors can be determined to achieve the desire tradeoff between the cost and mitigation effectiveness.

In our analysis, we also find that multi-provider stubs can be better mitigated than single-provider stubs. In both cases, there are several commonalities among the top choice mitigation points. Figure 12 illustrates three cases for using reflectors in mitigating single-provider stub d connected to ASes of different tiers. The top choice reflectors are the lightly shaded ASes. The most noticeable commonality among the reflectors is that they are all Tier-2 ASes with many Tier-1 and Tier-2 neighbors. This is also common to all top choice mirror locations as well. The other two commonalities among these reflectors are that (i) they are relatively close (e.g., one or two hops away) to the target d, and (ii) the path between the reflector and d contains, in decreasing

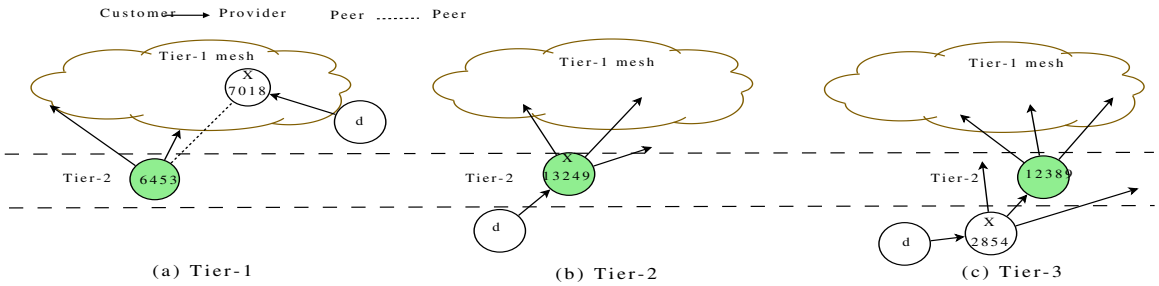


Fig. 12. The examples of reflector selection

preference order, provider, peer, and customer links.

2) Mitigation Selection Strategies: Based on our results, we suggest two general strategies for selecting reflectors and mirrors.

1) Find the reflector r which has smallest chance to be polluted by a hijacking event on the target stub customer d. This is complimentary to the detection selection. It is preferable to select a reflector r (i) of which the origin route from d to r is a customer route than a peer route than a provider route; (ii) which is close to d; (iii) which covers as few number of potential hijacker routes learned from providers and peers as possible. Note that this strategy applies only to reflector selection and is not needed for mirror selection.
2) Find the reflector r which will not be easily hijacked (i.e. r is resilient). To achieve the high resilience of r, one need to select a r which reaches as many Tier-1 ASes and other large ISPs as possible via customer routes. In addition, the route to from r to each of these Tier-1 ASes and large ISPs should be short. The similar suggestion regarding resilience is also discussed in [24]. This strategy applies to both reflector and mirror selection.

D. Impact of Incomplete AS topology

As we mentioned in Section IV-A, the AS topology is incomplete, and it may lead to overestimation or underestimation of hijacking events. We now evaluate the robustness of our tower selection algorithms. According to previous study [34], many peer links between lower tiers' ASes can be missing in the inferred AS topology based on public BGP data. We assume that there are $x\%$ of peer links between Tier-3/Tier-4 ASes (i.e. others) are missing and the missing peer links are randomly distributed. To reconstruct the "complete" AS topology, we randomly select $n/(1-x\%)$ pairs of Tier-3/Tier-4 ASes, the two ASes in each of which are not neighbors, where n is the number of inferred peer links in the AS topology. We then add a peer link between ASes in each selected AS pair to the AS topology. We select the towers based on incomplete topology and evaluate the accuracy by simulating the hijack events based on the "complete" topology. Table I shows the average detection and mitigation effectiveness, when the number of towers are fixed as 6. We observe that the effectiveness decease when increasing x. It is because the larger x is, the larger the differences are between the topology used to select the towers and the complete topology. We also find that even missing half of peering links, the algorithm has relative high (more than 86%) coverage, indicating that our algorithm is robust to the missing links.

E. Impact of Route Diversity

In our evaluation, we mainly assume that (1) the hijacker will pollute all its neighbors to maximize the impact. (2) when one router in the AS is polluted, then all routers in this AS will be polluted. In reality, the hijacker may select some of neighbors to propagate the fake AS path announcement. Moreover, it is possible that some of routers in the AS will be polluted, especially when the AS is large e.g. tier-1 or tier-2 ASes. As a result, different routers within one AS may have different views of routes. Figure 13 shows an example. Assume that AS T is the owner of prefix p. Hijacker AS H announces itself as the prefix owner, and propagate the announcement through edge router $H2$. Routers $B1$, $B2$ and $C1$ are polluted. $C2$ is not because both one hop from C to H and T, $C1$ will prefer the routes learned from e-BGP session of T. As a result, AS A is not polluted, AS B is fully polluted, and AS C is partially polluted.

In order to evaluate the impact of these *route diversity* cases on our tower selection algorithm, we conduct the following simulation. We split hijacker AS and each tier-1/tier-2 ASes into two sub-ASes (like in Figure 13). These two parts have equal number of neighbor ASes. The overlap ratio of neighbors is y. Due to the difference of neighbor AS, these two sub ASes may have different view of AS updates. Under this condition, tower selection is more restricted: In order to cover the hijacking events, the detection tower should be in the AS whose both sub-ASes are polluted, e.g. AS B in Figure 13. In terms of mitigation, we assume that mitigation towers

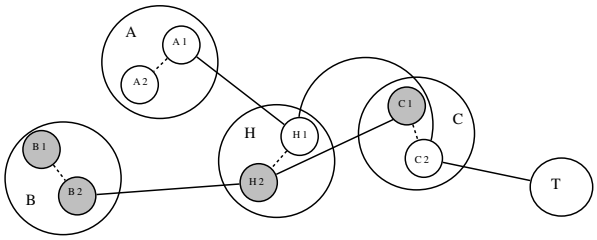


Fig. 13. The examples of route diversity.

should be in the AS whose neither of two sub-ASes are polluted, e.g. AS A in Figure 13.

To evaluate the impact of partial propagation, we compare the detection/mitigation effectiveness of the towers selected by original simulation environment (APX) and the new one (OPT), under the new and more "real" propagation cases. We fix the number of towers as 6 and tune the parameter y. Table II shows the results. The small value of y means the higher diversity of route views, which means that selection of detection tower and mitigation tower are more restricted, making it harder to to select the towers. We find that the smaller y is, the the smaller the detection/mitigation effectiveness is. We also find that the effectiveness of APX is slightly lower than OPT when $y = 0.1$ and $y = 0.9$. When $y = 1.0$, the effectivenesses are the same because two sub-ASes have identical view. Given that we have no idea that the real partial propagation looks like, we will still use original methodology.

F. Case Study: How Large ISPs May Improve Protection Effectiveness

We now use a case study to illustrate the value that the TOWERDEFENSE system may offer to large ISPs.

A large ISP often has multiple ASes. Thus it is tempting for such an ISP to simply deploy detection and mitigation points at its own ASes for protecting the ISP's customers. Such a deployment strategy may also seem effective because such ISPs networks often span across large geographic areas or even multiple continents. Our case study is about a large Tier-1 ISP. Despite the fact that this ISP has 20 ASes of its own, Figure 14 shows that the detection and mitigation effectiveness (averaging over all of its direct stub customers) are very low when only the ISP's own 20 ASes are used, with no additional towers(i.e., 0 on X-axis).

We first investigate how our tower selection algorithms can help improve this Tier-1 ISP's deployment strategy. First, when we start from scratch, 3 ASes[9] are enough to achieve the same effectiveness as using all 20 existing ASes can achieve. Second, in addition to using self-owned ASes, external ASes can be identified to help improve protection quality quickly. Figure 14 shows how protection quality significantly increases as the number of external ASes are used for deploying detection and mitigation towers.

Next, we use the same Tier-1 ISP as an example to show that TOWERDEFENSE service can be incrementally deployed. Based on the public topology data as of June 2009, in total this AS has 823 stub customers, including 390 single-provider customers and 433 multi-provider stub customers. Initially we randomly select one customer and we pretend this is the first customer signing up for prefix hijacking protection service. We deploy 6 towers using the methods as described before. Next, we randomly choose another customer and pretend that this is a new customer signing up for the service. It may or may not be necessary to add new tower or towers to maintain the overall protection effectiveness to be not lower than its current vale. Figure 15 shows how the number of towers increases as more and more customers sign up for the service. The gradual slopes of lines indicate that such service can be incrementally deployed as the number of customers increases. Even when a majority of its customers (800 out of 1266) have signed up one by one for the TOWERDEFENSE service, at most 20 towers (9 for detection, 11 for either mirror or reflector) are needed, as shown in the figure when the value on x-axis is 800.

The above case study shows that TOWERDEFENSE is very deployable from individual provider point of view. Assuming that TOWERDEFENSE can be deployed by many providers, we now investigate what kind of providers have more incentive to build offer such service (i.e., they are more cost-effective in terms of achieving a certain level of detection/mitigation effectiveness when new stub customers are subscribed to the service). In the evaluation, we randomly select some ISPs in different tiers (5 from Tier-1, 10 from Tier-2 and 20 from Tier-3). For each chosen provider, similarly to the previous case study, we start with 6 towers at the beginning. When new stub customers start to register the service, the provider have to keep adding new towers to maintain the originally effectiveness. To satisfy all stub customers, we compute the number of towers needed out of providers' ASes. We use the *customer per tower ratio* as the metric to measure the cost-effectiveness of deploying TOWERDEFENSE. We find that the average ratios are 36.4, 15.2, and 2.5 for Tier-1, Tier-2, and Tier-3 ISPs, respectively. The results indicate that the Tier-1 providers are most cost-effective and hence have more incentive to deploy and provide TOWERDEFENSE service.

V. INTERNET EXPERIMENTS

We evaluate TOWERDEFENSE performance by constructing synthetic hijacking attacks using Internet mea-

[9]The selected detection, reflector, and mirror towers can be different. Hence more than one ASes might be needed.

TABLE I
ROBUSTNESS OF TOWER SELECTION, FACING INCOMPLETE TOPOLOGY

x	0	10	20	30	40	50
Detection tower	.902	.895	.891	.883	.875	.867
Mirror	.953	.950	.941	.933	.929	.920
Reflector	.923	.918	.914	.908	.903	.891

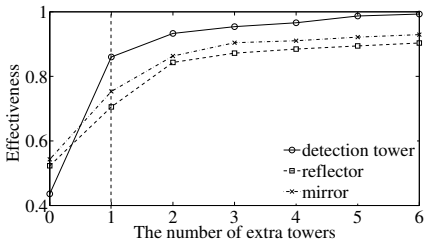


Fig. 14. Effectiveness vs. # of extra towers.

TABLE II
ROBUSTNESS OF TOWER SELECTION, FACING ROUTE DIVERSITY

y	0.1		0.9		1.0	
	APX	OPT	APX	OPT	APX	OPT
Detection tower	.822	.853	.848	.863	.902	.902
Mirror	.906	.927	.932	.944	.953	.953
Reflector	.882	.902	.903	.914	.923	.923

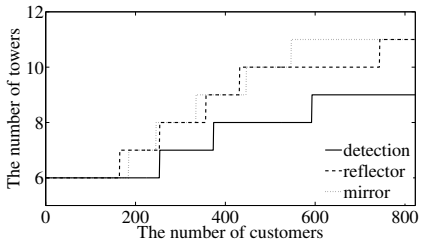


Fig. 15. The incremental deployment.

surements on Planetlab [35].

A. Experimental Methodology

We conduct our experiments in the following steps. First, we identify a set of target prefixes used in the experiments. Then, we select candidate Planetlab nodes to serve as the base of our experimental infrastructure. Each node can serve as detection tower, mitigation tower, traffic source, or hijacker in various attack scenarios. Next, for each target prefix, we select detection towers and mitigation towers among candidate Planetlab nodes using TOWERDEFENSE methodology. As a comparison, we also implemented monitor selection schemes studied in [36]: (1) **random based**: monitor nodes are selected randomly and (2) **greedy link based**: at any time, the next detection tower is selected with the largest number of unobserved links, given the set of already detection selected towers. Oppositely, the next mitigation tower is selected with the largest number of observed links, given the set of already selected mitigation towers. Finally, we construct all possible attack scenarios among candidate Planetlab nodes and evaluate the performance of TOWERDEFENSE.

Protected Target Selection. We select target prefixes from four different groups: (i) Multiple Origin ASes (MOAS) prefixes, (ii) Single Origin AS (SOAS) prefixes with large traffic volume, (iii) prefixes of popular Web sites, and (vi) prefixes of popular online social networks. Combining prefixes from four groups, we have a total of 343 target prefixes. We manually identify the service provider of these target prefixes. 57 of them are served by CDN providers, while the rest are served by ISPs. For each of the 201 target prefix with multiple providers, we randomly select one as the service provider which provides defense service to the prefix using TOWERDEFENSE. More details of target selection can be found in [33].

Planetlab Nodes Selection. We manually select 73 Planetlab nodes in 36 distinct ASes at different geographical regions. More specifically, relying on the DNS name, we select half of US nodes, which covers both coasts and the middle area; and half from other countries, which cover different continents. These 73 Planetlab nodes serve as the base for our experiments, i.e., potential hijackers, traffic sources, and detection/mitigation towers are selected from these nodes. The reasons of not selecting all Planetlab nodes are: (1) some nodes are co-located (e.g. multiple nodes in one university campus). They do not provide much gain in using TOWERDEFENSE; (2) some nodes are unstable or heavily loaded. We exclude them from our experiments.

Measurement Data Gathering. In our experiments, each selected Planetlab node measures its paths to all live IP addresses in all selected target prefixes via *traceroute*. In addition, each Planetlab node also measures its paths to other Planetlab nodes. We obtain AS-level paths of above measured paths by mapping IP addresses to their ASes based on the IP-to-AS mapping published at iPlane [37].

Constructing Synthetic Prefix Hijacking Events. Assuming that "prefer customer routes" and "valley-free routing" are used as interdomain routing policies, we use the same method as in Section III for determining whether an AS is impacted by the hijacking event. For a target prefix d, we first select detection towers and mitigation towers from the 73 Planetlab nodes using greedy algorithms described in Section III. The selection process is based on the assumption that the traffic source and hijacker can be potentially anywhere on the Internet.

We now construct synthetic prefix hijacking events on the Planetlab nodes. In particular, we first select one Planetlab node as the source s, another Planetlab node as the hijacker h, which attempts to hijack the target prefix d. Then we construct attack scenario using the

TABLE III
EFFECTIVENESS OF TOWERDEFENSE OVER PROTECTED TARGETS.

	TowerDefense		Ramdon		Greedy-link	
	AVG	STD	AVG	STD	AVG	STD
Detection tower	.943	.013	.632	.104	.842	.062
Reflector	.816	.023	.432	.203	.719	.107
Mirror	.846	.022	.443	.228	.684	.127

same methods described in [20].

We repeat experiments for all possible selections of h, s, and d, except for cases where d's AS is on the AS path from s to h because the hijack will never succeed in these cases. In addition, since some paths were not traceroute-able, we had to discard combinations that require these paths.

B. Detection Tower Selection Effectiveness

We use the detection method proposed in [20], which uses hop count and path divergence information obtained from the data plane. In addition, we use a fixed number of detection towers (i.e., 6)[10] in the Planetlab experiments and compute the average detection effectiveness for each target prefix.

Table III compares the effectiveness of detection tower selection using TOWERDEFENSE algorithm, random and greedy-link based algorithm [36]. We observe that our algorithm yields the highest detection effectiveness. Greedy-link algorithm is better than Random algorithm, but not as good as our algorithm because its optimization goal is to maximize link visibility of AS topology, rather than hijacking probability.

Though we use the detection method proposed in [20] in our experiments, TOWERDEFENSE can adopt any of the existing detection methods [2], [15]–[20], [31]. The only exception is iSpy [22]. iSpy is a data plane prefix hijacking detection method that is designed to be used by the target prefix itself. Another important difference between TOWERDEFENSE and iSpy is that TOWERDEFENSE carefully chooses a small number of detection towers and probes from the detection towers to the target prefix, while iSpy probes from the target prefix to every transit AS on the Internet. Figure **??** compares the effectiveness of TOWERDEFENSE and iSpy with varying probing costs under default settings. We observe that when the number of probe paths is small, TOWERDEFENSE can achieve much higher detection ratio (the percentage of detected hijacking events) than iSpy. For example, TOWERDEFENSE can achieve over 90% detection ratio by using 5 detection towers, while iSpy can achieve about 50% detection ratio if 5 random transit ASes are probed. Both methods benefit from adding more detection towers or probing more transit ASes. When all 73 Planetlab nodes are used as detection towers, TOWERDEFENSE can achieve 99.87% detection ratio. The corresponding figure for iSpy is about 90% when 73 transit ASes are probed. On the other hand, we also observe that iSpy can achieve 99.54% detection ratio when all (thousands of) transit ASes are probed. This implies that TOWERDEFENSE is much more cost effective than iSpy, though both methods can achieve comparable detection ratio when probing cost is not a concern.

[10]Gains of adding additional towers become marginal, similar to simulation in Section IV.

C. Mitigation Tower Selection Effectiveness

Recall that TOWERDEFENSE uses two types of mitigation towers: reflectors and mirrors. We use three metrics in evaluating the effectiveness of the mitigation tower selection: (i) mitigation effectiveness; (ii) reduction in the impact of hijacking event; and (iii) change in AS path lengths for impacted traffic.

Mitigation Effectiveness. For each target prefix, we select a fixed number of mitigation towers (i.e., 6) among candidate Planetlab nodes and compute the mitigation effectiveness for each possible attack scenario. Table III compares average mitigation effectiveness achieved by both mirrors and reflectors selected using TOWERDEFENSEalgorithm described in Section III with random and greedy-link algorithms. Similar to detection results, we observe that our algorithm is the best, because our algorithm is tailored to optimizing the mitigation effectiveness. More specifically, we observe that the average mitigation effectiveness is about 80% with 6 carefully selected reflectors (Note that the ratio is lower than that in Section IV because we have very limited number of candidate selections in Planetlab). The average mitigation effectiveness is close to 90% with 6 mirrors.

Hijacking Impact Reduction. We measure the impact of a hijacking event by the percentage of ASes from which the path to the target prefix is polluted by the hijacker. We compare the impact of a hijacking event before and after using mitigation towers. Figure 17 shows the hijacking impact reduction when 6 mitigation towers are used in TOWERDEFENSE. We observe that the use of reflectors or mirrors significantly reduced the impact of hijacking events (e.g., from 65% $\sim$ 90% to 10% $\sim$ 15%). Again, the reduction of hijacking impact by using mirrors is more significant than that of using reflectors.

Changes in AS Path Lengths. In TOWERDEFENSE, the impacted traffic is re-routed to or through mitigation towers. We compare the AS path lengths of the impacted traffic before and after using mitigation towers for each target prefix. Figure 18 shows that the average AS path lengths increases 1.7 AS hops and 0.6 AS hops when reflectors or mirrors are used, respectively. Note that a

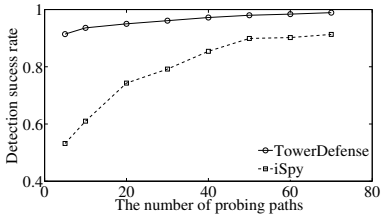


Fig. 16. The effectiveness of detection tower selection, compared with iSpy

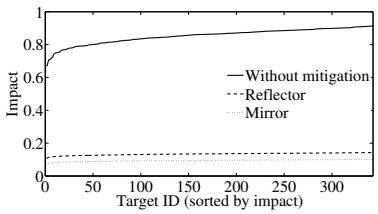


Fig. 17. Hijacking impact reduction using mitigation towers.

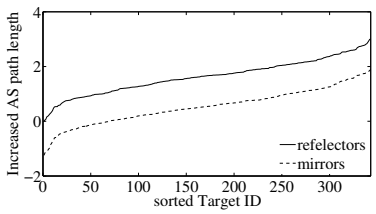


Fig. 18. The change of AS path lengths using mitigation towers

negative value means a decrease in AS path lengths. This is observed for some target prefixes when mirrors are used, when some mirrors are placed in the upstream providers of the target prefix.

VI. Related work

A number of solutions have been proposed to proactively defend against prefix hijacking [2]–[14], but the placement and deployment problems are not the focuses of these work. These approaches also need to change router software, router configurations, network operations, or introduce public key infrastructures, and most of them also need explicit collaboration with others, which make immediate deployment very difficult. For example, in the mitigation approach in [14], victim AS needs to collaborate with its previous-arranged "Lifesaver" ASes to remove the bogus route and promote the genuine route.

The hijacking *detection* approaches [15]–[20], [31] use control-plane and/or data-plane vantage points to detect hijacking. However, most of them depends on existing routing information tapping points (e.g. Route Views [28] and RIPE [29] or regulated traffic access(e.g. PlanetLab [35]), which are often not optimum for hijacking detection.

Instead of using vantage points, iSpy [22] allows a prefix owner to detect hijacking attacks on its own prefix by probing a large number of transit ASes on the Internet. As shown in Section V-B, iSpy requires much more probing overhead than TowerDefense and only works for a specific type of hijacking attacks known as "blackholing".

VII. Conclusion

In this paper, we propose the practical deployment strategies for battling against IP prefix hijacking, which we call TowerDefense . We advocate that the best way to move forward prefix hijacking protection is to offer such a protection as a new type of service by existing service providers, and propose a simple heuristic for the placing detection and mitigation agents. Through extensive simulations and large scale experiments, we show that with a small number of detection and mitigation agents deployed at locations selected by our selection algorithms, TowerDefense can achieve high detection and mitigation success ratios. Our case study of one Tier-1 ISP as TowerDefense provider also shows that high success ratios can also be achieved when detection and mitigation points are incrementally deployed.

References

[1] Y. Rekhter, T. Li, and S. Hares, "Border Gateway Protocol 4," Internet Engineering Task Force, RFC 4271, Jan. 2006.

[2] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz, "Listen and Whisper: Security Mechanisms for BGP," in *Proc. USENIX NSDI*, Mar. 2004.

[3] W. Aiello, J. Ioannidis, and P. McDaniel, "Origin Authentication in Interdomain Routing," in *Proc. of ACM CCS*, Oct. 2003.

[4] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure Path Vector Routing for Securing BGP," in *Proc. ACM SIGCOMM*, Aug. 2004.

[5] J. Ng, "Extensions to BGP to Support Secure Origin BGP," April 2004, ftp://ftp-eng.cisco.com/sobgp/drafts/draft-ng-sobgp-bgp-extensions-02.txt.

[6] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE JSAC Special Issue on Network Security*, Apr. 2000.

[7] K. Butler, P. McDaniel, and W. Aiello, "Optimizing BGP Security by Exploiting Path Stability," in *Proc. ACM CCS*, Nov. 2006.

[8] B. R. Smith and J. J. Garcia-Luna-Aceves, "Securing the Border Gateway Routing Protocol," in *Proc. Global Internet*, Nov. 1996.

[9] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, "Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing," in *Proc. NDSS*, Feb. 2003.

[10] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Protecting BGP Routes to Top Level DNS Servers," in *Proc. IEEE ICDCS*, 2003.

[11] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Dection of Invalid Routing Announcement in the Internet," in *Proc. IEEE/IFIP DSN*, June 2002.

[12] J. Karlin, S. Forrest, and J. Rexford, "Pretty Good BGP: Protecting BGP by Cautiously Selecting Routes," in *Proc. IEEE ICNP*, Nov. 2006.

[13] S. Y. Qiu, F. Monrose, A. Terzis, and P. D. McDaniel, "Efficient Techniques for Detecting False Origin Advertisements in Inter-domain Routing," in *Proc. IEEE NPsec*, Nov. 2006.

[14] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical Defenses Against BGP Prefix Hijacking," in *Proc. ACM CoNext*, Dec. 2007.

[15] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Topology-based Detection of Anomalous BGP Messages," in *Proc. RAID*, Sept. 2003.

[16] "RIPE myASn System," http://www.ris.ripe.net/myasn.html.

[17] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A Prefix Hijack Alert System," in *Proc. USENIX Security Symposium*, Aug. 2006.

[18] X. Hu and Z. M. Mao, "Accurate Real-time Identification of IP Prefix Hijacking," in *Proc. IEEE Security and Privacy*, May 2007.

[19] G. Siganos and M. Faloutsos, "Neighborhood Watch for Internet Routing: Can We Improve the Robustness of Internet Routing Today?" in *Proc. IEEE INFOCOM*, May 2007.

[20] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A Light-Weight Distributed Scheme for Detecting IP Prefix Hijacks in Real-Time," in *Proc. ACM SIGCOMM*, Aug. 2007.

[21] T. Qiu, L. Ji, D. Pei, J. Wang, J. Xu, and H. Ballani, "Locating Prefix Hijackers using LOCK," in *Proc. USENIX Security Symposium*, Aug. 2009.

[22] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "Ispy: Detecting IP Prefix Hijacking on My Own," in *Proc. ACM SIGCOMM*, Aug. 2008.

[23] Y. Zhang, Z. Zhang, Z. M. Mao, and Y. C. Hu, "HC-BGP: A Light-weight and Flexible Scheme for Securing Prefix Ownership," in *Proc. DSN-DCCS*, 2009.

[24] M. Lad, R. Oliveira, B. Zhang, and L. Zhang, "Understanding Resiliency of Internet Topology Against Prefix Hijack Attacks," in *Proc. IEEE/IFIP DSN*, June 2007.
[25] "Tower defense," http://en.wikipedia.org/wiki/Tower_defense.
[26] V. Cardellini, M. Colajanni, and P. S. Yu, "Redirection algorithms for load sharing in distributed web-server systems," in *Proc. IEEE ICDCS*, May. 1999.
[27] A. Shaikh, R. Tewari, and M. Agrawal, "On the Effectiveness of DNS-based Server Selection," in *Proc. IEEE INFOCOM*, Apr. 2001.
[28] "University of Oregon Route Views Archive Project," http://www.routeview.org.
[29] "RIPE RIS Raw Data," http://www.ripe.net/projects/ris/rawdata.html.
[30] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Transactions on Networking*, 2001.
[31] H. Ballani, P. Francis, and X. Zhang, "A Study of Prefix Hijacking and Interception in the Internet," in *Proc. ACM SIGCOMM*, Aug. 2007.
[32] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points," in *Proc. IEEE INFOCOM*, Apr. 2002.
[33] T. Qiu, L. Ji, D. Pei, J. Wang, and J. Xu, "TowerDefense: Battling Against Prex Hijacking as a Service," Georgia Institute of Technology, Tech. Rep. GT-CS-09-08., May 2009.
[34] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (in)Completeness of the Observed Internet AS-level Structuree," in *IEEE/ACM Trans. Networking*, 2010.
[35] "PlanetLab," http://www.planet-lab.org.
[36] Y. Zhang, Z. Zhang, Z. M. Mao, Y. C. Hu, , and B. Maggs, "On the Impact of Route Monitor Selection," in *Proc. ACM IMC*, 2007.
[37] "iPlane," http://iplane.cs.washington.edu/.