SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2016, Vol. 59 052105:1–052105:17
doi: 10.1007/s11432-015-5490-8

An accurate distributed scheme for detection of prefix interception

Song LI[1], Haixin DUAN[2*], Zhiliang WANG[2], Jinjin LIANG[3] & Xing LI[1]

[1]*Department of Electronic Engineering, Tsinghua University, Beijing* 100084*, China;*
[2]*Institute of Network Science and Cyberspace, Tsinghua University, Beijing* 100084*, China;*
[3]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China*

Received April 14, 2015; accepted September 18, 2015; published online April 8, 2016

Abstract Previous research in interdomain routing security has often focused on prefix hijacking. However, several prefix interception events have happened lately, which poses a new security challenge to the interdomain routing system. Compared to prefix hijacking, prefix interception is much harder to detect, as it avoids black hole by forwarding the hijacked traffic back to the victim. In this paper, we present a novel method to detect prefix interception. Our approach exploits a key observation about prefix interception: during a prefix interception event, the attacker detours the intercepted traffic through its network, which turns it into a new important "transit point" for access to the victim. By collecting data plane information to detect the emerging "transit point" and using control plane information to verify it, our scheme can identify prefix interception in real time. The results of Internet experiments and Internet-scale simulations show that our method is accurate with low false alarm rate (0.28%) and false negative rate (2.26%).

Keywords routing, BGP, hijacking, interception, detection

Citation Li S, Duan H X, Wang Z L, et al. An accurate distributed scheme for detection of prefix interception. Sci China Inf Sci, 2016, 59(5): 052105, doi: 10.1007/s11432-015-5490-8

1 Introduction

The Border Gateway Protocol (BGP) plays a key role in maintaining the Internet routing infrastructure. It is essential for exchanging reachability information between Autonomous Systems (ASes) in the Internet. For many years one of the biggest security threats to BGP was prefix hijacking, which occurs when an AS hijacks routes by advertising bogus prefixes. Prefix hijacking can likely be discovered because it generally leads to unreachability to the victim's network (i.e., black hole) [1]. However, there have been some "prefix interception" incidents recently [2–4], which not only hijack the traffic destined for the victim's prefixes but also forward the traffic back to the victim. Different from prefix hijacking, such prefix interception events are not readily noticed by the victim as it does not cause reachability problem. As a result, they could last for a long time while keeping the end users in the dark. More importantly, as reported by [2–4], the intercepted traffic was diverted through unintended networks or countries, which makes the traffic face risks of being eavesdropped or modified.

* Corresponding author (email: duanhx@tsinghua.edu.cn)

It is not clear whether those prefix interception events are caused by misconfiguration or attacks, but it is certain that they pose a new serious threat to the Internet routing. There are numerous proposals for detecting or preventing prefix hijacking so far, but most of them cannot defend against prefix interception due to some challenges. (1) For the cryptographic based approaches [5–9], previous research has shown that partial deployment will have limited security benefits [10]. Moreover, recent studies [11,12] indicated that even in the case of full deployment, there are some forms of prefix interception attacks that the cryptographic schemes cannot prevent. (2) For the detection proposals [13–16], they often identify prefix hijacking by discovering anomalous BGP routes (control plane) and black holes (data plane). However, prefix interception cannot also be detected by those methods. First, prefix interception is a kind of Man-in-the-Middle (MITM) attack. It intercepts the victim's traffic and then sends it back to its original destination. Thus, prefix interception does not result in black holes. Second, prefix interception is often based on prefix hijacking [17], and hence they share the same anomaly in control plane. As a result, monitoring anomalous BGP routes is not enough to identify a prefix interception, as the actual forwarding path is needed to distinguish prefix interception from prefix hijacking.

In this paper, we present a novel scheme to detect prefix interception. Our key insight about prefix interception is that the intercepted traffic destined for the victim's prefix passes through the attacker AS only after the prefix interception. In other words, there will be a number of ASes in the Internet which are polluted by the ongoing prefix interception. As a result, their traffic destined for the victim's prefix will detour through the attacker's network, which turns the attacker AS into a new important "transit point" for access to the victim's network. On the basis of this observation, we designed a distributed real-time detection scheme, which detects the emerging "transit point" AS to discover an ongoing prefix interception.

Our approach combines the data plane with control plane information to detect prefix interception. For the data plane, we exploit a lot of vantage points to generate traceroute traffic to the protected prefix periodically. By translating the Router-Level traceroute results to directed AS-Level graphs, we continuously monitor the in-degree and out-degree changes of all nodes (ASes). If the in-degree and out-degree of an AS increase dramatically (i.e., it becomes an emerging transit point), the AS is identified as an Upstart-AS. According to the observation above, that the Upstart-AS is detected can be considered as a data plane signature of ongoing prefix interception. For the control plane, we detect BGP routing anomaly caused by prefix interception in real time. Through the joint analysis of detection results in data plane and control plane, our scheme can distinguish prefix interception from other legitimate route change events. Besides, it can provide detailed information about the prefix interception, which includes the suspicious attackers (i.e., the Upstart-AS) and the real interception paths.

This paper makes four main contributions. First, we present a comprehensive attack model of prefix interception (Section 2). Second, we propose a novel distributed real-time scheme for detecting prefix interception (Section 3). Third, we deploy our detection system in the Internet to monitor prefix interception. The Internet experiments show that our detection system is light-weight, and the detection results show that the false positive rate is ideal (0.28%). Finally, we conduct Internet-scale simulations to evaluate the false negative rate of our detection scheme. We show that when the pollution of the prefix interception is over 5%, the false negative rate of our approach is below 2.26% (Section 4).

2 Attack model of prefix interception

Prefix interception generally refers to the Internet traffic interception based on prefix hijacking [17]. However, some other features in BGP protocol, such as valley-free rule [18,19] and AS-Path prepending, can also be exploited to launch a prefix interception attack [11,20]. Commonly, in a prefix interception scenario, an AS redirects the traffic destined for the victim's prefix to its network by advertising an invalid route, and then forwards the attracted traffic back to the origin destination. This behavior can be intentional or unintentional. But it should be emphasized that in any case the "attacker" AS should have at least one route along which the attracted traffic can be sent back to the victim's network [17].

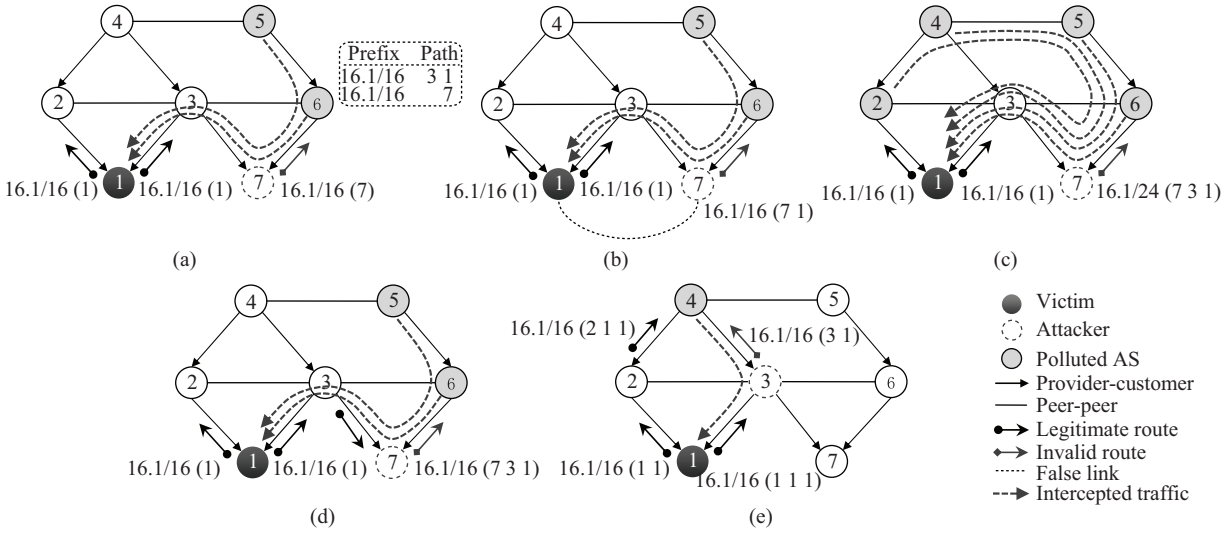

Figure 1 A classification of prefix interception. (a) MOAS-interception; (b) MOAS-Evasion-interception; (c) More-Specific-interception; (d) Valley-Free-interception; (e) ASPP-interception.

This route can be called a clean route, as the attacker AS should ensure all ASes in the route are not polluted by the invalid route advertised by itself.

In this section, we provide a classification of prefix interception to facilitate the ensuing discussion of our detection scheme. We group prefix interception into the following five categories. To the best of our knowledge, this is the first comprehensive prefix interception taxonomy to date.

(1) MOAS-interception: The attacker attracts the traffic destined for the victim's prefix by advertising the same prefix of the victim AS, following by the forwarding of that traffic back to the victim along a clean route. As illustrated in Figure 1(a), since the bogus route advertised by the attacker will lead to multiple origin AS (MOAS) [21] claiming for the same prefix in the Internet, we call this type of prefix interception MOAS-interception.

(2) MOAS-Evasion-interception: Due to the obvious feature of MOAS in the global routing table, the MOAS-interception is likely to be monitored and detected. To evade detection against MOAS-interception, the attacker can falsify the AS-Path so that it looks like the bogus route was originated from the legitimate AS, while in fact there was no link between the attacker AS and the victim AS. Clearly, this kind of attack will not generate MOAS. We named this attack MOAS-Evasion-Interception. Figure 1(b) shows one case of this interception.

(3) More-Specific-interception: The attacker intercepts traffic by advertising a more specific prefix than the victim advertised. Note that because the more specific bogus route will pollute almost all ASes in the Internet, to ensure a clean route for sending back data, the attacker should set the AS-Path like [22]. For instance, in Figure 1(c), the attacker AS7 has two routes (3 1) and (6 3 1) for the prefix 16.1/16. It selects (3 1) as the clean route and advertises a more specific prefix 16.1/24 with the AS-Path (7 3 1). Clearly, AS3 and AS1 will not accept the bogus route because of routing loops. Therefore, it ensures that the clean route (3 1) is not polluted.

(4) Valley-Free-interception: The attacker intercepts traffic by leaking an invalid route that violates the valley-free rule. For example, in Figure 1(d), AS7 leaks the route learned from its provider AS3 to another provider AS6. According to the common import routing policies (i.e., which route is chosen as the best BGP route) [19], AS6 will prefer the leaked customer route (7 3 1) over the peer route (3 1). As a result, its traffic destined for 16.1/16 is detoured through the attacker AS7.

(5) ASPP-interception: The attacker intercepts traffic by exploiting the BGP AS-Path prepending (ASPP) mechanism. ASPP is often used for traffic engineering purpose. For instance, as shown in Figure 1(e), AS1 announces the routes with 2 and 3 duplication of its own number to AS2 and AS3 respectively. Normally this will make the incoming traffic from AS4 traverse AS2 to reach it. However,

the attacker AS3 can modify the received route by removing the duplicated AS number and send the shorter path to AS4, which will result in traffic of AS4 being diverted through it.

In the five interception types above, the first three are all based on prefix hijacking and hence can be classified as hijacking-based prefix interception. Our subsequent discussion will focus on detecting them, and the term "prefix interception" used in the following sections will refer to the hijacking-based prefix interception. The latter two (Valley-Free-interception and ASPP-interception) are related to routing policy and beyond the scope of this paper. We do not discuss them in detail. However, as we will demonstrate in Section 4, it is important to note that they can also be detected by the detection scheme presented in this paper.

3 System design

As mentioned in Section 1, prefix interception is harder to detect than prefix hijacking. For one thing, prefix interception only diverts the traffic along an unintended path to the victim's prefix. It does not interrupt the traffic and thus is not visible to the end users. As a result, to detect the invisible traffic interception, the actual forwarding path of the packets should be monitored. For another, prefix interception is often performed via prefix hijacking. Consequently, detecting the control-plane anomaly caused by the prefix hijacking is also necessary in order to identify a prefix interception.

In this section, we present a prefix interception detection scheme using both data-plane and control-plane information. The design of our detection system is motivated by the recent prefix interception events [3]. By carefully analyzing those interception events, we found the key feature of prefix interception and designed a detection scheme to discover and identify it.

3.1 Key observation

To gain insight into prefix interception, we gathered the control plane and data plane data of prefix interception events on July 31st, 2013 [3]. First, we collected all BGP updates on the day of events from Route-Views[1)], and then filtered out the hijacked prefixes advertised by the "perpetrator" AS (AS6677). Second, we downloaded traceroute archives on the day and previous day of the events from iPlane [23]. iPlane performs daily traceroutes to a representative set of IP addresses from several vantage points, including PlanetLab[2)] nodes and traceroute servers. We processed the trace archives and extracted traceroutes to those hijacked prefixes performed by the same vantage points on July 31st and July 30th. After resolving IP-Level traceroutes to AS-Level forwarding paths (by IP-to-AS mapping), we compared the AS-Level paths of the two days for each hijacked prefix. As we expected, we discovered some vantage points were impacted by the interception events and thus resulted in difference of AS-Level paths between the normal day and event day.

Figure 2 shows a case of different AS-Level paths for the victim prefix 68.28.53/24. The black path represents the forwarding path on the normal day, and the red path represents the interception path on the event day. From the figure, we can see that there are eight polluted vantage points, whose traceroute packets to the target prefix 68.28.53/24 are misdirected to AS6677 and then forwarded along AS3257, AS1239 to destination AS10507.

If we regard Figure 2 as a directed graph, the in-degree and out-degree of the node AS6677 will both be 8. Note that Figure 2 draws only different AS-Level paths before interception and after interception. This means that if we plot two directed graphs, one graph contains AS-Level forwarding paths from all vantage points to 68.28.53/24 on July 30th (normal day), the other graph includes the same kind of paths on July 31st (event day), then both the in-degree and out-degree of AS6677 will increase from 0 on the former graph to 8 on the latter graph.

Obviously, the significant degree increase of AS6677 is the direct result of prefix interception performed by itself. In other words, a key observation of this prefix interception event is: during a prefix interception

1) University of oregon route views project. http://www.routeviews.org.
2) Planetlab. https://www.planet-lab.org.

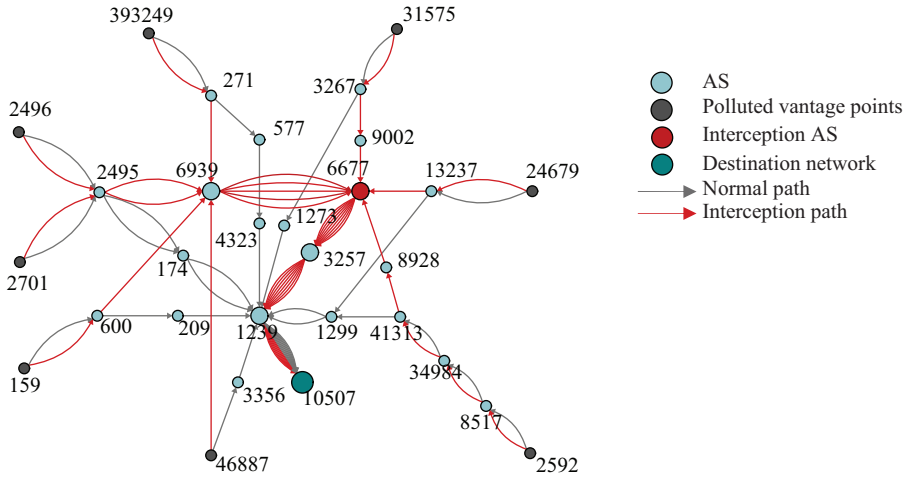

Figure 2 AS6677 intercepts 68.28.53/24.

event, the attacker AS will become a new important transit point on the map for forwarding packets to the victim's prefix, i.e., its in-degree and out-degree will increase significantly on the AS-Level forwarding topology. This key observation gives us a useful enlightenment: we can design an interception detection system based on the monitoring of degree changes of ASes. Or, to be more specific, we can select a lot of vantage points widely distributed across the Internet and perform traceroutes from those vantage points to the target prefix periodically. Once the in-degree and out-degree of an AS calculated from traceroute topology suddenly increased significantly, this AS that we call Upstart-AS can be suspected to be related to a prefix interception with high probability.

3.2 Upstart-AS

We gave a loose description of Upstart-AS above. To define it in more detail, we first discuss the degree distribution of ASes in the directed graph formed by the AS-Level forwarding paths from all vantage points to a single IP address. There are several studies [24, 25] showing that the Internet topology can be described by power laws. The Internet graphs used in those studies were mainly inferred from the global BGP routing tables or traceroutes from a single source to a large number of destinations. The graph in our studies, on the contrary, is formed by performing traceroutes from several vantage points to a single destination. Thus, the graph is much smaller than the Internet graph. Moreover, we are mainly concerned with the in-degree distribution of ASes. This is because the out-degree of a node (AS) is equal to its in-degree, except the source (vantage points) and destination.

3.2.1 *Pareto distribution*

We downloaded the trace archive on March 28th, 2014 from iPlane dataset. To make the results reliable, we randomly selected 1000 destination IP addresses from the set of iPlane probe targets. For each destination IP address, the traceroute data were translated to AS-Level paths and collected to form a directed graph. Thus, we got 1000 graphs. And for each graph, we sort the nodes in descending order of in-degree. After removing the nodes whose in-degree is equal to 0 (these nodes are vantage points), we discovered that the distribution of in-degree in most graphs approximately follows the Pareto distribution. That is, a larger portion of in-degrees is owned by a smaller percentage of the nodes.

Figure 3(a) shows complementary cumulative distribution function (CCDF) of in-degree in three representative graphs. From Figure 3(a), we can see that the three lines are nearly straight on log-log scales. This matches the characteristic of Pareto distribution. Pareto distribution has a well-known application—"80-20 rule", which means 20% of the population control 80% of the wealth. Inspired by this, we counted the percentages of in-degrees of the top 20% ASes for the 1000 graphs. Figure 3(b) and (c) shows the probability distribution and cumulative distribution of the 1000 results. It is clear that

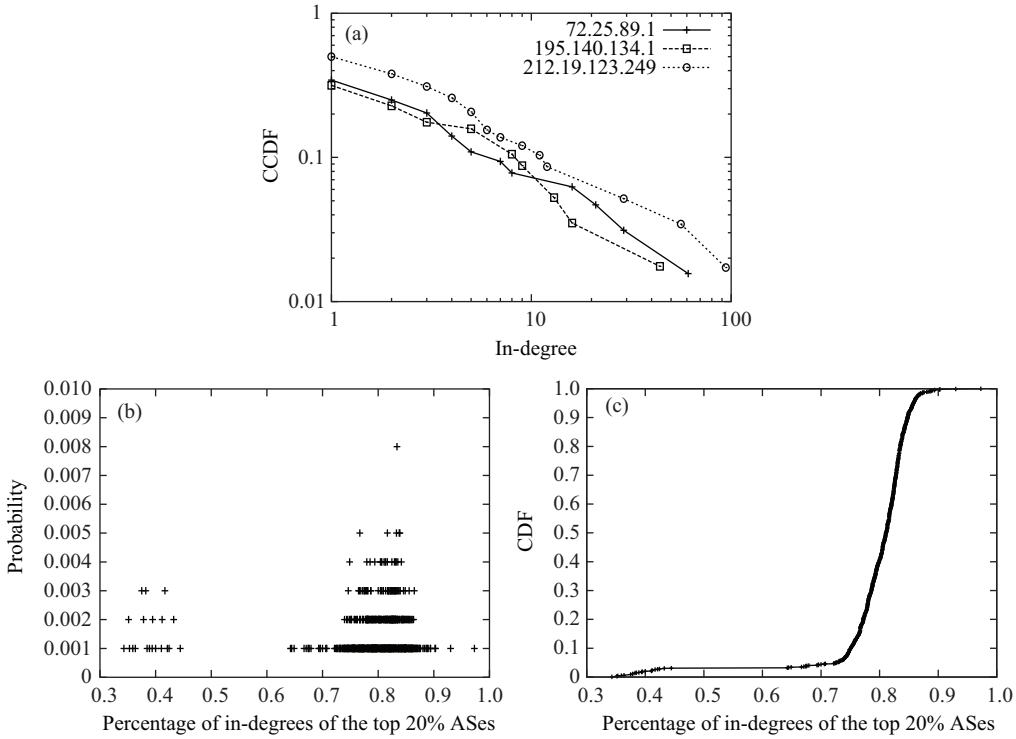

Figure 3 Pareto distribution of in-degree. (a) CCDF; (b) probability; (c) CDF.

most of the results are located between 70% and 90%. We also calculated the mean of them, which is 79.41%. This means the top 20% of ASes have nearly 80% of the total in-degrees in a statistical sense. Based on this statistic, we give the detailed definition of Upstart-AS below.

3.2.2 *Definition of Upstart-AS*

For a directed AS-Level graph, if we regard in-degree as "wealth", as mentioned above, there is a gap between the rich ASes and poor ASes, and the dividing line is the minimum in-degree of the top 20% of ASes. Note that those all other ASes that do not appear on the graph can be classified into poor ASes, as their in-degree and out-degree are both equal to 0.

Under the circumstances of monitoring in-degree and out-degree change periodically, we define Upstart-AS as follows:

• For a poor AS, if its in-degree and out-degree increases are greater than or equal to the last minimum in-degree of the top 20% of ASes (threshold 1), the AS is considered an Upstart-AS.

• For a rich AS, if its in-degree and out-degree increases are greater than or equal to the last mean in-degree of the top 20% of ASes (threshold 2), the AS is also considered an Upstart-AS.

The reason we set threshold 1 for poor ASes to the minimum in-degree of the top 20% of ASes is once the in-degree of a poor AS increases to over threshold 1, the poor AS has become a rich AS in a short time (one period), and this can be identified as an anomaly. Similarly, setting threshold 2 is based on the following consideration: the threshold 2 represents the average in-degree of the rich ASes. If the in-degree growth of one rich AS exceeds threshold 2, the AS can be also considered as an anomalous AS.

For instance, Figure 4(a) and (b) show the rank of ASes in the graph destined for 68.28.53/24 on July 30th and July 31st. On July 30th, AS6677 did not appear on the graph. And consequently, its in-degree and out-degree were equal to 0. On July 31st, AS6677 appeared in the rich ASes. Its in-degree and out-degree became 8 and the increases of them were also 8, which was greater than the threshold 1 (value was 3) on July 30th. Thus, AS6677 is a typical Upstart-AS.

Note that in Figure 4, according to the definition of Upstart-AS, AS6939 is also considered as an Upstart-AS. From Figure 2, we can see that this is because AS6939 is one of the main neighbors (transit

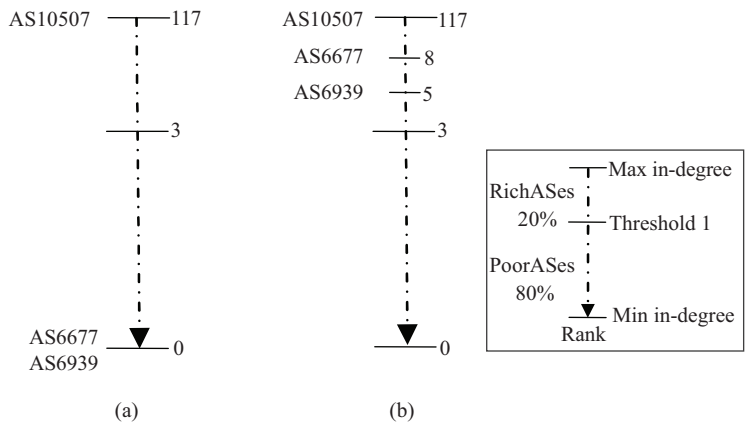

Figure 4 Rank of ASes for 68.28.53/24 on (a) July 30th, and (b) July 31st.

providers) of AS6677. Once AS6677 intercepts traffic, AS6939 will be impacted and forced to transit the intercepted traffic to AS6677. As a result, its in-degree and out-degree will increase correspondingly and thus lead to it also being detected as an Upstart-AS.

3.3 Detection scheme

While we have revealed that prefix interception will always generate Upstart-ASes, another important question is if the Upstart-ASes can only be caused by prefix interception. As we will demonstrate in the next section, other factors such as legitimate route change can also bring about Upstart-ASes. Therefore, detecting Upstart-ASes alone cannot identify a prefix interception. To solve this problem, our detection scheme further checks the control plane information. As illustrated in Section 2, the hijacking-based prefix interceptions (MOAS-interception, MOAS-Evasion-interception and More-Specific-interception) are performed by advertising hijacking prefixes, and hence they will cause corresponding control plane anomalies (i.e., anomalous Origin AS, anomalous adjacent AS and anomalous subprefix). Therefore, this control plane signature can be used to distinguish prefix interception from other legitimate route change events.

Figure 5 demonstrates the framework of our detection scheme. Our scheme consists of two main modules: Upstart-AS detection module (UADM) and routing anomaly monitoring module (RAMM). The UADM detects Upstart-ASes for the target prefixes periodically, and the RAMM monitors anomalous BGP routes for the same prefixes continuously. Once the UADM detects Upstart-ASes for one prefix, the Upstart-ASes are output to the RAMM. The RAMM will then check those Upstart-ASes to see if they are contained in the anomalous routes of that prefix, which were detected in the last period. If there are some Upstart-ASes that appear in the detected anomalous BGP routes, the detection scheme raises a prefix interception alarm and identifies those Upstart-ASes as the suspicious attackers of the prefix interception. Instead, if there are no matches in the anomalous routes for all the Upstart-ASes, an undetermined alarm will be raised with the detected Upstart-ASes.

It should be mentioned that the undetermined alarm does not necessarily mean it is a false alarm. The reason for the undetermined alarm may be various: legitimate route change, Valley-Free-interception, ASPP-interception, and others. They can all generate Upstart-ASes but all do not cause obvious routing anomalies in control plane. Without more detailed routing policy information, it is hard to identify whether they are prefix interceptions or not. Therefore, they are classified as undetermined alarms.

3.3.1 *Upstart-AS detection module*

We now formulate the UADM in detail. Suppose there are M prefixes to be protected against interception attack. First, for each prefix, we select one live IP address to represent it, and hence we have M IP addresses as probing targets. Next, we select N vantage points distributed in different N ASes. From all these N vantage points, traceroutes destined for M target IPs are performed simultaneously with a

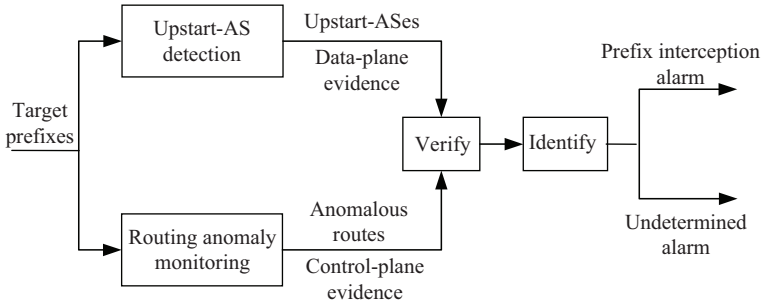

Figure 5 Framework of our detection scheme.

period of T. Certainly, the period T is set to be long enough so that traceroutes to all target IPs have finished at the start of the next period.

Once the traceroute task has been done, each vantage point uploads the traceroute results to a data server. The data server then translates all $N \times M$ traceroutes data to AS-Level paths. This can be achieved by looking up IP-to-AS mapping table. We use Route-Views data to generate IP-to-AS mapping table. And to ensure the correctness of the mapping, the table is updated periodically.

After getting the $N \times M$ AS-Level paths, we cluster them according to target IP (prefix). For each target prefix x_i $(1 \leqslant i \leqslant M)$, there are N AS-Level paths for it, which can form a directed AS graph $D_i = (V, A)$ $(1 \leqslant i \leqslant M)$. In the graph, the node set V consists of ASes included in the AS-Level paths, and the arrow set A consists of ordered AS pairs that are adjacent hops in paths.

Let $D_{i,t} = (V, A)$ denote the graph of prefix x_i at time t. $I_{v,i,t}$ $(v \in V)$ and $O_{v,i,t}$ $(v \in V)$ denote the in-degree and out-degree of ASes in graph $D_{i,t} = (V, A)$. Then ASes in graph $D_{i,t} = (V, A)$ are ranked based on $I_{v,i,t}$ and classified into poor AS set $P_{i,t}$ and rich AS set $R_{i,t}$. In addition, the threshold 1 $\delta_{1,i,t}$ and threshold 2 $\delta_{2,i,t}$ are also calculated:

$$\delta_{1,i,t} = \min\{\ I_{v,i,t}\ \} \quad v \in R_{i,t}\ , \tag{1}$$

$$\delta_{2,i,t} = \left(\sum_{v\ \in\ R_{i,t}} I_{v,i,t} \right) /\ |R_{i,t}|\ . \tag{2}$$

Now, at time $t + T$, new traceroute results are gathered to the data server. To determine if there are prefix interceptions that occurred in the last period, we observe the difference between $I_{v,i,t+T}$, $O_{v,i,t+T}$ and $I_{v,i,t}$, $O_{v,i,t}$. If there exists $v \in P_{i,t}$ such that

$$I_{v,i,t+T} - I_{v,i,t} \geqslant \delta_{1,i,t} \ \text{ and } \ O_{v,i,t+T} - O_{v,i,t} \geqslant \delta_{1,i,t}\ , \tag{3}$$

or there exists $v \in R_{i,t}$ such that

$$I_{v,i,t+T} - I_{v,i,t} \geqslant \delta_{2,i,t} \ \text{ and } \ O_{v,i,t+T} - O_{v,i,t} \geqslant \delta_{2,i,t}\ , \tag{4}$$

the system will report an event that Upstart-ASes are detected for the prefix x_i, and put those ASes meeting the above conditions into Upstart-ASes set.

3.3.2 *Routing anomaly monitoring module*

Running simultaneously with the Upstart-AS detection module, the routing anomaly monitoring module continuously detects the following three main types of anomalous routes:

(1) Anomalous origin AS: For the target prefix, there is a new origin AS discovered in BGP updates. This anomaly can be caused by MOAS-interception.

(2) Anomalous adjacent AS: For the target prefix, there is a new AS found adjacent to the origin AS in BGP updates. The MOAS-Evasion-interception can lead to this anomaly.

(3) Anomalous subprefix: For the target prefix, there are new BGP routes with the subprefix of it in the global routing tables. This anomaly is used to detect More-Specific-interception.

Our routing anomaly monitoring module uses the Route-Views repository and the RIPE RIS[3)] database to extract normal routing information related to the target prefix. The information includes its origin ASes, adjacent ASes and prefix length. With the normal routing information, we exploit BGPmon[4)] to gather real time routing information of the prefix, and make a comparison between them. So long as the three kinds of anomalies are detected, the system filters out the anomalous BGP routes and stores them for verifying the detected Upstart-ASes.

4 Evaluation

Our prefix interception detection system has been running on the Internet since April 2014. The UADM was deployed on 307 PlanetLab nodes which belong to 150 ASes, and the RAMM performs passive monitoring of the BGP updates on our local network. Generally, there are two important requirements for an anomaly detection system: efficient and accurate. In this section, we evaluate the performance of our detection system by the two measures.

4.1 Efficiency

Because the RAMM is nearly real time, for evaluating the efficiency of our system, we are primarily concerned about the probing latency and the amount of traffic generated by the UADM. To measure the probing latency, we randomly selected 10, 100 and 1000 prefixes as the probing targets from the global routing table. Traceroutes to x.x.x.1 of each prefix were performed from 100 Planetlab nodes. Figure 6 shows the distributions of the probing time with the number of concurrent traceroutes in each node not being greater than 40. The average time for probing 10, 100 and 1000 prefixes are 24, 55 and 396 s.

Second, the traffic generated by the 100 probing nodes was measured. Note that the amount of outbound traceroute traffic is determined by the hops that the node takes to reach the target prefix. The more the hops are, the more the traffic is. We set the size of traceroute packet to 64 bytes (i.e., minimum Ethernet frame size). The average outbound traffic for a node to probe 10, 100 and 1000 prefixes are 43 kB, 484 kB and 4.8 MB.

Finally, we measured the total possible inbound probing traffic into one target prefix. To study the inbound probing traffic, we configured our one host as the probing target of the 100 Planetlab nodes. The probing traffic into our host was measured for ten cycles. The average inbound probing traffic is 66 kB and the average inbound bandwidth is 25 kbps. It can be inferred that the probing traffic will grow linearly as the probing nodes increases. That is, if we can exploit 1000 Planetlab nodes as vantage points, the probing traffic to one target in one cycle will be about 660 kB.

From the results above, we can see that both the probing time and the traffic generated by the UADM are not large. This confirms that our detection system is efficient and light-weight.

4.2 False positive rate

In order to evaluate the detection accuracy of our scheme, we carried out Internet experiments to test the false positive ratio of the system. We selected 10 sites (shown in Table 1) as protected targets (prefixes) from Alexa top 20 sites and set the probing period to 10 min. For illustrative purposes, we provide the detection results of one month, from April 17th to May 16th, to demonstrate the detection accuracy of our system.

During the month, our detection system performed 4293 probing for each site, and reported 0 prefix interception alarm and 123 undetermined alarms out of the total 42930 detections. This indicates that no definite hijacking-based prefix interceptions are detected. However, it's not certain that all the 123 undetermined alarms are false alarms. They could also be Valley-Free-interception or ASPP-interception, as discussed in Subsection 3.3. To determine if they are false or not, we further carefully analyzed all the alarms with the data-plane information (traceroute data captured by our system), control-plane

3) Ripe ris. http://www.ripe.net/data-tools/stats/ris.
4) Bgp monitoring system. http://bgpmon.netsec.colostate.edu.

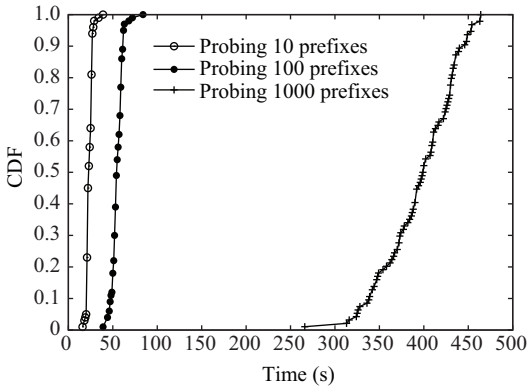

Figure 6 Distributions of the probing time.

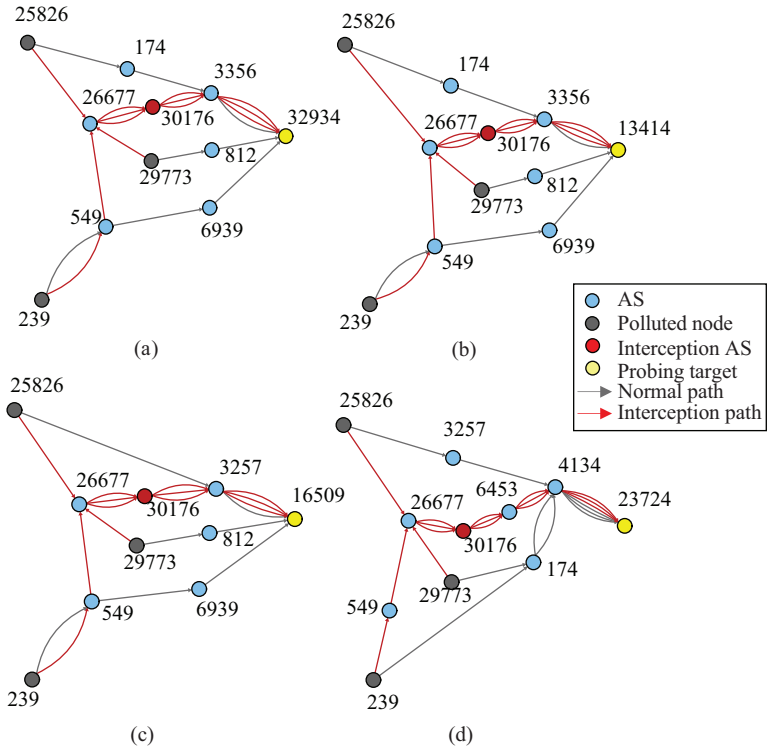

Figure 7 Valley-Free-interceptions. (a) Facebook; (b) Twitter; (c) Amazon; (d) Baidu.

information (BGP updates) and routing polices queried from IRR[5]. We finally figured out 4 Valley-Free-interception alarms and 119 false alarms.

4.2.1 *The true interception alarms*

On May 9th, 2014, the detection system raised an undetermined alarm with the Upstart-ASes set {26677, 30176} for the Twitter site at 18:50 UTC. And at 19:00 it raised again the alarms with the same Upstart-ASes for another three sites: Facebook, Amazon and Baidu. Figure 7 shows the 4 alarm events. From the figure, we can see the abnormal AS-Level forwarding path segments captured by the UADM: (26677 30176 3356), (26677 30176 3257) and (26677 30176 6453).

We believe that the 4 alarms were true Valley-Free-interception alarms based on the following evidences. (1) By looking up the ground truth relationships [26] and extracting the relationships from IRR database and BGP Community attribute [27], we know that AS3356, AS3257 and AS6453 are the providers of

5) Irr-internet routing registry. http://www.irr.net.

Table 1 False alarms

10 Sites (Prefixes)	False alarms				False alarm rate
	Load balancing	Policy-based route change	Network congestion	Route flapping	
Google.com (173.194.41/24)	3	0	2	0	5 (0.12%)
Facebook.com (173.252.96/19)	0	1	1	0	2 (0.05%)
Yahoo.com (98.138/16)	0	0	0	0	0 (0.00%)
Baidu.com (220.181.96/19)	6	2	0	1	9 (0.21%)
Taobao.com (42.120/16)	0	3	18	0	21 (0.49%)
Live.com (65.52/14)	0	2	36	0	38 (0.89%)
Twitter.com (199.16.156/22)	0	0	0	0	0 (0.00%)
Amazon.com (176.32.96/21)	2	3	0	0	5 (0.12%)
Sina.com.cn (202.108/18)	1	3	1	22	27 (0.63%)
Weibo.com (114.134.80/24)	0	8	4	0	12 (0.28%)
Total	12	22	62	23	119 (0.28%)

AS30176, and AS26677 is the peer of AS30176. Hence, according to the export routing policies [18,19], it is a violation of the Valley-free rule that AS30176 advertises routes received from its provider (AS3257, AS3356 and AS6453) to its peer (AS26677). (2) When examining the historical BGP updates in the same time period, we found that at 18:46 AS30176 started advertising large numbers of routes (17550 in total) to some of its neighbors. This usually means a route leak event caused by misconfiguration. (3) If the routes are normal, they will reasonably exist for some time. However, we have not observed them since May 9th 2014.

4.2.2 *The false interception alarms*

Apart from the above 4 true interception alarms, we confirmed that all the other 119 alarms are false. The reasons for the false alarms can be classified into four kinds: load balancing, legitimate route change, network congestion and route flapping. Table 1 shows the statistics of the 10 sites as well as the summary statistics. The total false alarm rate is 0.28%, and more than half of the false alarms were caused by network congestion. Besides, some protected sites, such as Live.com and Sina.com.cn, have more false alarms than other sites, which pulls up the average of false positive ratio. Figure 8 illustrates the four types of false alarms. Below we explain how they generated false alarms for the protected sites.

First, Figure 8(a) shows a practical example of load balancing. AS2381 is a transit AS between AS40387 and AS15169. And from the detailed router topology, we can see that R1 and R5 are the load balancing routers. According to [28], traceroute from AS40387 to AS15169 may get anomalous result paths because of the load balancing between them. Two of anomalous paths gathered by our probing module were (R1, R3, R4, R7, R9, R8) and (R1, R2, R5, R7, R6, R8), whose TTL sequences both exactly correspond to (4, 5, 6, 7, 8, 9). After mapping IP to AS, the two paths became (40387 15169) and (40387 2381 15169 2381 15169). Therefore, when calculating the degree of nodes in the two paths, the in-degree and out-degree of AS2381 increased to 2 in the latter path. This triggered a false interception alarm with the Upstart-AS AS2381.

Second, Figure 8(b) shows an instance of legitimate route change. For some unknown reasons, on May 14th, 2014, AS32934 modified its routing policies and switched one of its upstream providers from AS3356 to AS6762. This impacted our six vantage points and changed their paths to AS32934. From Figure 8(b), we can see that the in-degree and out-degree of AS6762 increased to 6 after this policy-based route changed. Of course, it was detected as an Upstart-AS by our detection system and led to an undetermined alarm.

Third, Figure 8(c) shows an example of network congestion. When networks in AS4134 were congested, the three probing nodes in the figure could not receive ICMP response packets from AS4134. Thus the degree of AS4134 was zero. When the congestion finished, traceroutes from the three probing nodes to the target site (Baidu.com, AS23724) also recovered, and the in-degree and out-degree of AS4134 became

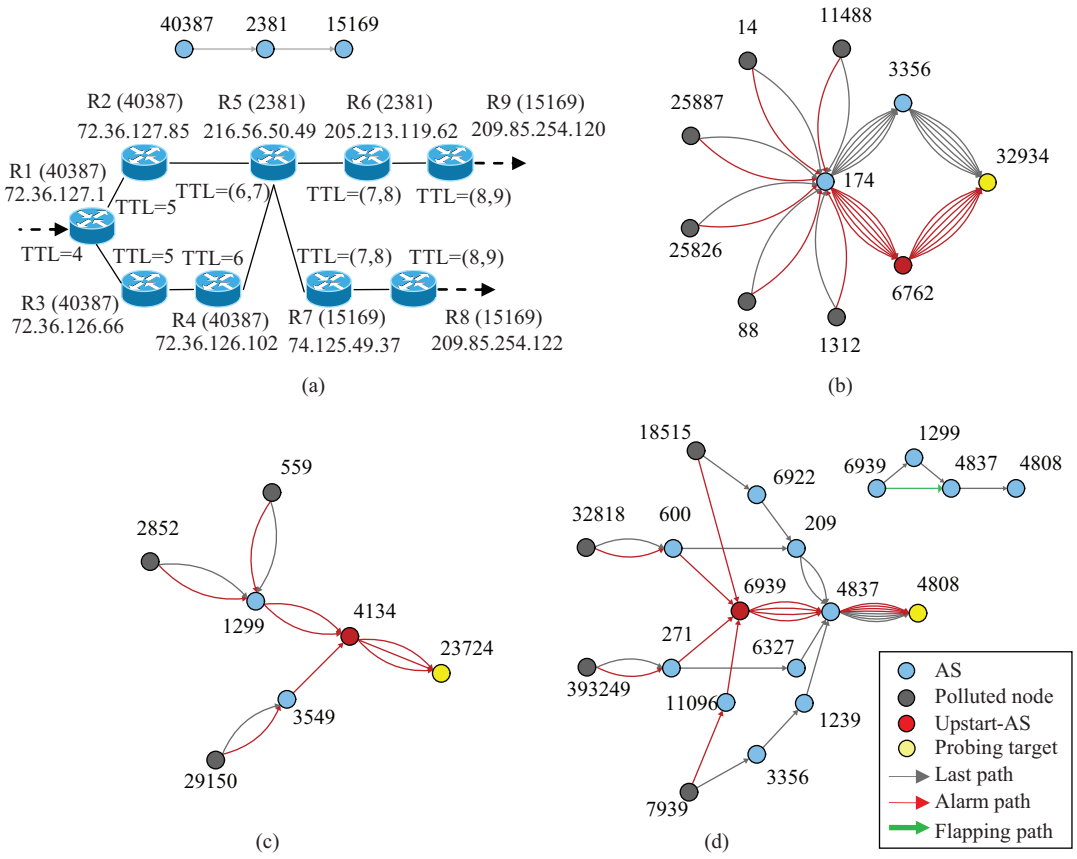

Figure 8 Four types of false alarms. (a) Load balancing; (b) legitimate route change; (c) network congestion; (d) route flapping.

3. As a result, the detection system raised an undetermined interception alarm for this instance.

Fourth, Figure 8(d) shows an instance of route flapping. On May 10th, 2014, the link between AS6939 and AS4837 continuously flapped. As a result, AS6939 chopped and changed the advertised BGP route for the protected site (sina.com.cn, AS4808) between (6939 1299 4837 4808) and (6939 4837 4808). In our detection system, there were four vantage points impacted by this route flapping. As we can see from Figure 10(d), the black paths were the AS-Level forwarding paths of the four vantage points when AS6939 advertised the route with longer AS-Path (6939 1299 4837 4808), and the red paths were their changed forwarding paths when AS6939 advertised the route with shorter AS-Path (6939 4837 4808). Obviously, each flapping (back to the shorter path) caused the in-degree and out-degree increases of AS6939, which made the detection system generate an undetermined interception alarm.

4.3 False negative rate

As mentioned above, the UADM was deployed in 150 ASes located in different Internet regions. These 150 vantage points may not be able to detect some interception events when the interceptions are low impact. We now evaluate the false negative ratio of our detection system by simulating in C-BGP [29].

4.3.1 *Simulation methodology*

We made Internet-Scale simulations by loading AS-Level topology[6)] into the C-BGP. The topology we used consists of 45427 ASes. We first randomly selected 100 ASes as the victim ASes announcing legal prefixes from all of the 45427 ASes. Next, for each victim AS, we also chose 100 attacker ASes for it. Note that to ensure the attacker AS has at least one clean route, the 100 attacker ASes are randomly chosen

6) The caida ucsd as-relationships - 20131101. http://data.caida.org/datasets/as-relationships/serial-1/.

Table 2 Simulation results

Interception types	Simulation instances		
	Total	Unsuccessful interceptions	Successful interceptions
MOAS-interception	10000	5664(56.64%)	4336(43.36%)
MOAS-Evasion-interception	10000	4398(43.98%)	5602(56.02%)
More-Specific-interception	10000	187(1.87%)	9813(98.13%)

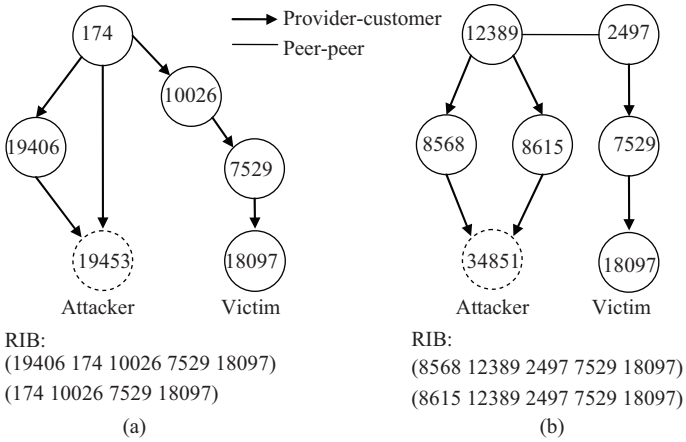

Figure 9 Two cases of More-Specific-interception. (a) Case #1; (b) Case #2.

from a complete set of ASes with at least two routes to the victim AS. Finally, to get real evaluations, we set the same 150 ASes used in our real Upstart-AS detection module as the vantage points in our simulation.

After setting up the simulation, we simulated all the three hijacking-based prefix interceptions as follows. First, the victim AS advertises a test prefix owned by it, which makes all ASes add a BGP route to the test prefix. Second, the attacker AS launches the special type of interception attack on the test prefix, which will possibly pollute the routing tables of some ASes. Third, traceroutes to the test prefix are performed from the 150 vantage points and the results are saved for the next step. Finally, the Upstart-AS detection algorithm is used to detect interception attacks from the results of the previous step. It should be mentioned that there are two questions in the second step, which are how to select clean route and how to advertise the false prefix. For the first question, because there are at least two routes to the victim in the routing table of the attacker AS, we randomly select one of them as the clean route. For the second question, we make the attacker AS advertise the false prefix to all its neighbors, aside from the neighbor providing the clean route for it.

4.3.2 *Simulation results*

We simulated 10000 interception instances for each type of hijacking-based prefix interception attack. Of all the 30000 instances, we filtered out 4336, 5602 and 9813 successful interception instances respectively. Note that a successful interception instance means that there is at least one AS, whose traffic destined for the victim AS does not pass through the attacker AS before the prefix interception attack, but does go through it after the prefix interception. The unsuccessful interception instances in the simulation can be classified into two categories: (1) there is no AS whose traffic destined for the victim AS is intercepted by the attacker. This is often caused by the failure of prefix hijacking. (2) the clean route is polluted. Table 2 shows the simulation results in detail.

It is a little surprising that about 2% of More-Specific-interception instances are unsuccessful. We expected that all More-Specific-interception instances would be successful. Figure 9(a) shows one case of unsuccessful More-Specific-interception instance. The attacker is AS19453, a customer AS multi-homed to AS19406 and AS174. There are two routes to AS18097 in its routing table, but it unfortunately selected (19406 174 10026 7529 18097) as the clean route. According to the attack method of More-

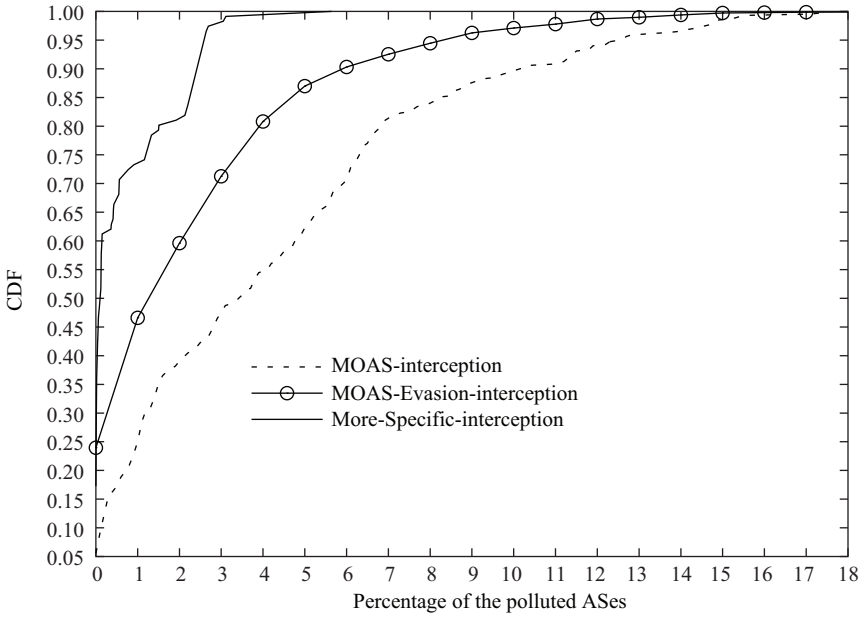

Figure 10 The percentage distribution of the polluted ASes of the false negative instances.

Table 3 Statistics of the detection rate and false negative rate (FNR)

Interception types	Pollu.<=5%			Pollu.>5%			Detection rate (%)	FNR (%)
	Instances	Detected	FN	Instances	Detected	FN		
Interception[1]	425	158(37.18%)	267(62.82%)	3911	3742(95.68%)	169(4.32%)	89.94	10.06
Interception[2]	1893	670(35.39%)	1223(64.61%)	3709	3487(94.02%)	222(5.98%)	74.21	25.79
Interception[3]	124	9(7.26%)	115(92.74%)	9689	9688(99.99%)	1(0.01%)	98.82	1.18
Total	2442	837(34.28%)	1605(65.72%)	17309	16917(97.74%)	392(2.26%)	89.89	10.11

[1]MOAS-interception. [2]MOAS-Evasion-interception. [3]More-Specific-interception.

Specific-interception, it advertised a more specific route to AS174 with the AS-Path (19406 174 10026 7529 18097). Obviously, this route was rejected by AS174 because of the routing loop. Therefore, no AS was polluted by the route and then the prefix interception failed. We analyzed all the unsuccessful More-Specific-interception instances and found that this is the case with them.

Next, of all the successful interception attacks, some instances polluted a large part of the Internet, and other instances only impact a few ASes. Generally speaking, the former can be more easily discovered by the UADM than the latter. Table 3 shows the detection results of different pollution levels. We divided all the successful interception instances into two kinds: instances with less than 5% of polluted ASes and instances with more than 5% of polluted ASes. When the percentage of polluted ASes is less than or equal to 5%, the detection rates for the three types of interception attacks are a little low. By contrast, when the percentage is more than 5%, the detection rates increase significantly, which become 95.68%, 94.02% and 99.99% respectively; the total false negative rate also declines to ideal level (2.26%).

For each type of interception, its detection rate and false negative rate are determined by the proportions of the above two kinds of interception instances. For example, for the MOAS-Evasion-interception, the former kinds of instances are nearly one-third of all the successful instances. This clearly raises the false negative rate for detecting it. For the More-Specific-interception, the former is about 1% of the total, thus the false negative rate is very small. Nevertheless, it is surprising again that there are 124 More-Specific-interception instances with less than 5% of polluted ASes. As an example, in Figure 9(b), AS34851 has two routes to AS18097. Whichever it chooses as the clean route, there will be only one AS (AS8568 or AS8615) being polluted.

Finally, to gain insight into the false negative instances, we study the percentage distribution of the polluted ASes of them. As shown in Figure 10, for the three types of interception attacks, there are nearly

62%, 85% and 99% of the false negative instances whose percentages of the polluted ASes are below 5%. These instances can be considered low impact and they are hard to be detected. From the results, we can see that the low impact of the interception instance is the main causes of the false negative.

Note that there are still small parts of false negative instances with the percentages of polluted ASes being greater than 5% (but less than 18%). The impacts of these instances are not too low, but also not high. As a result, some of the 150 vantage points are polluted, but the polluted number is not enough for making the degree increase of the attack AS exceed the alarm threshold. Consequently, these small parts of instances are not detected by the Upstart-AS detection module.

5 Discussion and future work

(1) System optimizations. The evaluation in the previous section shows that our system is efficient and accurate. However, there are some optimizations that can be used to improve the performance of the system further. First, we can use Paris-traceroute [28] to replace the default traceroute in Linux. Paris-traceroute is a good tool to address the problem of anomalous traceroute caused by the load balancing. It can obtain a more precise forwarding path that the packets actually follow than the typical traceroute tool. Therefore, using Paris-traceroute will definitely eliminate the false alarms resulting from load balancing. Second, it is best to deploy two or more data servers in different networks so that when one network interrupts, the other can continue to receive the traceroute results from the vantage points.

(2) Vantage points. There are two main factors that need to be considered when choosing the vantage points: one is the locations of the vantage points; the other is the total of them. For the former, an important principle is to ensure their broad distribution in the Internet. For the latter, it is clear that too few vantage points are bad for the Upstart-AS detection module, because it will lead to a high false negative ratio. However, too many vantage points are not likely good for the detection system; it will reduce the efficiency of the system while the detection accuracy is improved. On the other hand, there are limited Internet resources that can be used as vantage points. Therefore, we plan to study that how many vantage points are optimum for the Upstart-AS detection module in the future.

(3) Threshold. The detection threshold can be adjusted according to different algorithms. For example, other than the mean value, the threshold 2 can also be the median in-degree of the rich ASes. It should be mentioned that different threshold will result in different detection accuracy. A lower threshold will raise the false positive rate and reduce the false negative rate. A higher threshold is just the opposite. In our experiments and simulations, when the threshold 2 was set to the mean value, the false positive ratio and false negative ratio are 0.28% and 2.26% (Pollu. >5%). And when taking the median value, they would be 0.73% and 2.25% (Pollu. >5%).

6 Related work

Most previous work [5, 8, 13, 15, 30] on interdomain routing security has focused on prefix hijacking detection or prevention. As discussed in Section 1, they are not able to address the challenge of prefix interception. Compared with prefix hijacking, prefix interception is more complicated and stealthy, which leads to the difficulty of detecting it.

So far, there have been a few proposals [17, 20, 31] that detect prefix interception by exploiting data-plane information or control-plane information (or both) collected from a lot of vantage points. Ref. [17] uses global routing table to find the set of next-hop ASes for the target prefix, and then checks the data-plane trace to discover those anomalous paths traversing two consecutive next-hop ASes. Since only MOAS-Evasion-interception can cause those anomalous data-plane paths, apart from it, the scheme in [17] cannot detect other forms of prefix interception. Similarly, the detection algorithm in [20] is especially designed for the AS-Path prepending based prefix interception and not applicable to other interceptions. Ref. [31] only relies on data plane measurements to discover black hole hijack, imposture and interception attacks. However, although it shows the capability of detecting them, it cannot distinguish whether the

attack is a prefix hijack or a prefix interception. More importantly, it cannot point out the suspicious attackers as our detection scheme does.

Our Upstart-ASes detection system used traceroute and IP-to-AS mapping to obtain the AS-Level forwarding paths; this relates to some existing work such as [15, 23]. Furthermore, we studied the in-degree distribution of the directed AS-Level graph formed by those AS-Level paths, and found that it follows the Pareto distribution. This part of work is inspired by other previous researches [24,25]. Finally, the routing anomaly monitoring module in our scheme builds on a lot of control-plane based hijacking detection proposals [16, 32, 33].

7 Conclusion

This paper presents a distributed detection system to identify prefix interception. The design of our detection system is based on the observation about prefix interception: a prefix interception will turn the attacker AS into an important transit point for access to the victim. Our approach consists of two main parts: Upstart-AS detection module and routing anomaly monitoring module. While the Upstart-AS detection module continuously probes the target prefix from several vantage points and calculates the in-degree and out-degree of ASes in the traceroute topology to find Upstart-ASes, the routing anomaly monitoring module detects anomalous BGP routes in real time. By using the anomalous routs to verify the detected Upstart-ASes, our detection system can identify prefix interception accurately and efficiently.

Different from the few existing proposals for detecting prefix interception, our detection scheme can address almost all forms of prefix interception attacks. In addition, our method has several other advantages: (1) It can provide detailed information about prefix interception, which includes the suspicious attackers and the real interception paths. (2) It is efficient with low detection latency and probing overhead. (3) It can detect prefix interception with high accuracy. (4) It is easy to deploy. (5) It is scalable, being able to include more vantage points. (6) It can be customized to protect the key sites.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61472215). We thank Cristel Pelsser for her helpful comments. We are also grateful to Randy Bush for his help.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Karrenberg D. Youtube Hijacking: a Ripe Ncc Ris Case Study. RIPE NCC Technical Report. 2008
2 Hiran R, Carlsson N, Gill P. Characterizing large-scale routing anomalies: a case study of the China telecom incident. In: Proceedings of the 14th International Conference on Passive and Active Measurement, Hong Kong, 2013. 229–238
3 Cowie J. The New Threat: Targeted Internet Traffic Misdirection. Dyn Research Technical Report. 2013
4 Madory D. Uk Traffic Diverted Through Ukraine. Dyn Research Technical Report. 2015
5 Kent S, Lynn C, Seo K. Secure border gateway protocol (s-bgp). IEEE J Sel Area Commun, 2000, 18: 582–592
6 Ng J. Extensions to BGP to support secure origin BGP (soBGP). IETF Draft draft-ng-sobgp-bgp-extensions-02. 2004
7 van Oorschot P C, Wan T, Kranakis E. On interdomain routing security and pretty secure bgp (psbgp). ACM Trans Inf Syst Secur, 2007, 10: 11
8 Lepinski M, Kent S. An Infrastructure to Support Secure Internet Routing. IETF RFC 6480. 2012
9 Xiang Y, Shi X, Wu J, et al. Sign what you really care about-secure bgp as-paths efficiently. Comput Netw, 2013, 57: 2250–2265
10 Lychev R, Goldberg S, Schapira M. BGP security in partial deployment: is the juice worth the squeeze? ACM SIGCOMM Comput Commun Rev, 2013, 43: 171–182
11 McPherson D, Osterweil E, Amante S, et al. Route-Leaks & MITM attacks against BGPSEC. IETF Draft draft-ietf-grow-simple-leak-attack-bgpsec-no-help-04. 2014
12 Li Q, Hu Y C, Zhang X. Even rockets cannot make pigs fly sustainably: can BGP be secured with BGPsec? In: Proceedings of the NDSS Workshop on Security of Emerging Networking Technologies, San Diego, 2014
13 Hu X, Mao Z M. Accurate real-time identification of IP prefix hijacking. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy, Oakland, 2007. 3–17
14 Zhao X, Pei D, Wang L, et al. Detection of invalid routing announcement in the Internet. In: Proceedings of the International Conference on Dependable Systems and Networks, Bethesda, 2002. 59–68

15 Zhang Z, Zhang Y, Hu Y C, et al. Ispy: detecting ip prefix hijacking on my own. ACM SIGCOMM Comput Commun Rev, 2008, 38: 327–338
16 Xiang Y, Wang Z, Yin X, et al. Argus: an accurate and agile system to detecting IP prefix hijacking. In: Proceedings of the 19th IEEE International Conference on Network Protocols, Vancouver, 2011. 43–48
17 Ballani H, Francis P, Zhang X. A study of prefix hijacking and interception in the Internet. ACM SIGCOMM Comput Commun Rev, 2007, 37: 265–276
18 Gao L. On inferring autonomous system relationships in the Internet. IEEE/ACM Trans Netw (ToN), 2001, 9: 733–745
19 Gill P, Schapira M, Goldberg S. A survey of interdomain routing policies. ACM SIGCOMM Comput Commun Rev, 2013, 44: 28–34
20 Zhang Y, Pourzandi M. Studying impacts of prefix interception attack by exploring bgp as-path prepending. In: Proceedings of the IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), Macau, 2012. 667–677
21 Zhao X, Pei D, Wang L, et al. An analysis of BGP multiple origin AS (MOAS) conflicts. In: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, San Francisco, 2001. 31–35
22 Pilosov A, Kapela T. Stealing the Internet: an Internet-Scale Man in the Middle Attack. Defcon Technical Report. 2008
23 Madhyastha H V, Isdal T, Piatek M, et al. iPlane: an information plane for distributed services. In: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, Seattle, 2006. 367–380
24 Faloutsos M, Faloutsos P, Faloutsos C. On power-law relationships of the internet topology. ACM SIGCOMM Comput Commun Rev, 1999, 29: 251–262
25 Siganos G, Faloutsos M, Faloutsos P, et al. Power laws and the AS-level internet topology. IEEE/ACM Trans Netw (TON), 2003, 11: 514–524
26 Luckie M, Huffaker B, Dhamdhere A, et al. AS relationships, customer cones, and validation. In: Proceedings of the 2013 Conference on Internet Measurement, Barcelona, 2013. 243–256
27 Xia J, Gao L. On the evaluation of AS relationship inferences [Internet reachability/traffic flow applications]. In: Proceedings of the Global Telecommunications Conference, Dallas, 2004. 1373–1377
28 Augustin B, Cuvellier X, Orgogozo B, et al. Avoiding traceroute anomalies with paris traceroute. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, Rio de Janeriro, 2006. 153–158
29 Quoitin B. Uhlig S. Modeling the routing of an autonomous system with C-BGP. IEEE Netw, 2005, 19: 12–19
30 Wählisch M, Maennel O, Schmidt T C. Towards detecting BGP route hijacking using the RPKI. ACM SIGCOMM Comput Commun Rev, 2012, 42: 103–104
31 Zheng C, Ji L, Pei D, et al. A light-weight distributed scheme for detecting IP prefix hijacks in real-time. ACM SIGCOMM Comput Commun Rev, 2007, 37: 277–288
32 Lad M, Massey D, Pei D, et al. Phas: a prefix hijack alert system. In: Proceedings of the 15th Conference on USENIX Security Symposium, Berkeley, 2006. 153–166
33 Karlin J, Forrest S, Rexford J. Pretty good BGP: improving BGP by cautiously adopting routes. In: Proceedings of the 14th IEEE International Conference on Network Protocols, Santa Barbara, 2006. 290–299