Accurate Real-time Identification of IP Prefix Hijacking

Xin Hu Z. Morley Mao
University of Michigan
huxin@umich.edu zmao@umich.edu

Abstract

We present novel and practical techniques to accurately detect IP prefix hijacking attacks in real time to facilitate mitigation. Attacks may hijack victim's address space to disrupt network services or perpetrate malicious activities such as spamming and DoS attacks without disclosing identity. We propose novel ways to significantly improve the detection accuracy by combining analysis of passively collected BGP routing updates with data plane fingerprints *of suspicious prefixes. The key insight is to use data plane information in the form of edge network fingerprinting to disambiguate suspect IP hijacking incidences based on routing anomaly detection. Conflicts in data plane fingerprints provide much more definitive evidence of successful IP prefix hijacking. Utilizing multiple real-time BGP feeds, we demonstrate the ability of our system to distinguish between legitimate routing changes and actual attacks. Strong correlation with addresses that originate spam emails from a spam honeypot confirms the accuracy of our techniques.*

1. Introduction

Analogous to identity theft, IP address hijacking, also known as fraudulent origin attack, is to steal IP addresses belonging to other networks. It is an attack on the routing infrastructure or Internet's control plane. To accomplish this, attackers announce hijacked address prefixes from networks they control, so that they can use the stolen addresses to send and receive traffic. To simplify, we use the term "IP hijacking" to mean hijacking of IP address prefixes.

Attackers may hijack IP address space for two purposes: (1) To Conduct malicious activities such as spamming and DoS attacks without worrying about disclosing their identity through the source IPs. Note that although source IPs can be easily spoofed due to lack of ubiquitous deployment of ingress filtering, establishing a TCP connection still requires using a routable IP address. (2) Intentionally disrupt the communication or reachability of legitimate hosts numbered with the stolen addresses – effectively a more stealthy type of DoS attack. Both types of hijacking can significantly disrupt the stability and security of the Internet. Moreover, stolen IPs were also found to be sold or leased to networks in need of IP address spaces [27]. Note that the symptom of IP hijacking from victim's perspective is similar to other outages, making it nontrivial to diagnose.

Besides malicious intent, IP hijacking can also result from unintentional network misconfigurations. The most notable example is the incident involving AS7007 [9] which accidentally advertised to its upstream provider a short path to numerous prefixes belonging to other networks. Its provider did not filter out these bogus announcements causing a large blackhole for many destinations.

IP hijacking is also known as BGP (Border Gateway Protocol) hijacking, because to receive traffic destined to hijacked IP addresses, the attacker has to make those IP addresses known to other parts of the Internet by announcing them through BGP [41, 28, 21], which is the interdomain routing protocol on the Internet today. A BGP route consists of a prefix and the AS path used to reach that prefix. IP hijacking occurs if an AS advertises a prefix that it is not authorized to use either on purpose or by accident. Because the current BGP protocol implements little authentication and often assumes a significant level of trust between peering ASes, IP hijacking can easily succeed. Furthermore, because a BGP router cannot know routing policies of its neighbors, nor can it accurately evaluate the validity of a routing announcement, this leads to significant difficulties in preventing malicious or misconfigured routing information from propagating through the entire Internet.

An obvious way to *prevent* IP hijacking is to ensure proper configurations of route filters at the links between network providers and their customers to preclude customers from announcing routes for prefixes they do not own. However, this is both difficult and insufficient: (1) Providers do not always know which address blocks their customers are assigned to, due to the prevalence of multihoming. This allows customers to obtain address prefixes from multiple providers. (2) Similar to ingress filtering, as long as there is one provider that does not properly enforce route filtering, IP hijacking becomes possible. (3) Compro-

mised routers in the core Internet can bypass such filters, as route filtering is impossible along peering edges due to lack of information on addresses allocated to customers belonging to one's peer, usually one's competitor.

Given the above difficulties, it is critical to detect and thwart potential IP hijacking attempts. Some of the existing work relying on registry information such as whois database is ineffective due to stale and inaccurate registry data. Other approaches focus on detecting anomalous control plane information – conflicts in origin ASes[1] in the announcements [51] and short-lived nature of routing updates [10]. These suffer from excessive false positives and false negatives, making them impractical for operational use. False positives result from legitimate reasons why seemingly anomalous routing updates occur. False negatives stem from the fundamental observation that the BGP AS-level path may not match the forwarding path [35]. Moreover, using timing as an anomaly indication further undermines online mitigation as the detection needs to wait for the hijacking attempt to disappear.

Our approach to defeating IP hijacking is to first detect, in real time, routing updates that indicate unauthorized announcement of address prefixes. *Our key insight is that a successful hijacking will result in* ***conflicting data plane fingerprints*** *describing the edge networks numbered with the announced address prefix.* This is because during a successful hijacking attack, the same prefix will be announced and used by multiple distinct networks. Thus, we exploit this fundamental property by light-weight fingerprinting that characterizes end-hosts or edge networks to accurately and efficiently ascertain IP hijacking attempts as soon as they occur. Such fingerprints can range from fine-grained host-based information like the host uptime to coarse-grained network information such as firewall policies. Essentially these fingerprints are identifying signature information for the network using the IP address prefix in question. Typically a hijacking attempt cannot succeed in affecting the entire Internet, especially networks topologically close to the network owning the prefix. A real hijacking routing update thus always generates disagreeing fingerprints obtained from different network vantage points.

Our work provides real-time detection of IP hijacking events as soon as they occur instead of post-mortem analysis common in most previous works. Online detection enables timely mitigation responses, for example in the form of requesting help through external channels. Here are our main contributions. We present a comprehensive framework for the attack model of IP hijacking, including attack types previously overlooked and not addressed. We propose detection techniques for each IP hijacking attack type based on several novel techniques such as AS edge popularity checking, AS relationship inference, active probing to collect data plane fingerprints confirming the attacks. Unlike previous work, our approach significantly reduces not only false positives using a variety of anomaly detection and constraint checking techniques on routing data, but also false negatives by successfully detecting previously overlooked IP hijacking types. Overall, we present an efficient, accurate, and general IP hijacking detection framework, readily deployed in today's Internet, requiring no ISP nor end-host cooperation, and validated using empirical data.

The rest of the paper is organized as follows. We first summarize related work in §2, followed by a description of a comprehensive classification of IP address hijacking in §3. §4 proposes our detection techniques for each attack type. To demonstrate the real-time detection capability, we present experimental results in §5. Validation using empirical data are shown in §6. Finally, §7 concludes the paper.

2. Related Work

IP hijacking is an attack on BGP. IETF's rpsec (Routing Protocol Security Requirements) Working Group provides general threat information for routing protocols [5] and in particular BGP security requirements [13]. Prefix origin authentication is one such requirement. Related to it is path authentication. As explained later, malicious AS inserted in the AS path can achieve similar damage as fraudulent origin ASes. A recent survey [11] gives a comprehensive overview on BGP security.

According to RFC1930 [23], a prefix is usually originated by a single AS. MOAS (Multiple Origin AS) conflicts result if multiple origin ASes announce the same prefix. Zhao *et al.* first coined the term MOAS, providing several legitimate explanations for them aside from misconfiguration and hijacking attacks [51]. Their subsequent work [52] suggested the use of BGP community attribute storing a list of originating ASes to detect potential violations. However, such a list is unauthenticated and optional, thus cannot ensure accurate detection of IP address hijacking. To protect routes to specific services such as DNS, Wang *et al.* [48] proposes preferring known stable routes over transient routes. Nevertheless, this approach does not scale to arbitrary routes.

The well-known BGP security architecture S-BGP [45] relies on digitally signed routing updates to ensure integrity and authenticity, assuming the presence of PKIs. Follow-up work such as psBGP [47] and [50] improve the efficiency of S-BGP. Both S-BGP and SoBGP [36] can defend against IP address hijacking attacks. However, their high overhead in terms of computational cost, modification of protocol and additional management overhead prevents their rapid deployment. The Interdomain Routing Validation (IRV) project [20] uses an out-of-band mechanism to

[1] Origin AS is the AS originating the route announcement for a given IP prefix; it is the last AS in the AS path, as each AS prepends its AS number when propagating the route.

validate received routing information by querying the IRV server in the relevant AS. However, it does not prevent an AS from originating a prefix it does not own. The Listen and Whisper scheme [46] also helps identify inconsistent routing advertisement, but does not deterministically detect IP hijacking attacks. Similar to our approach, it takes advantage of data plane information. Complimentary to our techniques, the recent work by Aiello *et al.* [4] investigates the semantics, design, and application of origin authentication services by formalizing address delegation semantics and exploring the use of various cryptographic structures for asserting block ownership and delegation.

Compared to these related work, our approach focuses on practical, readily deployable mechanisms using data plane information to validate occurrences of IP hijacking in real time. Many operational requirements for secured BGP have not been addressed [8], hindering the deployment of solutions such as S-BGP. In contrast, our solution can be incrementally, easily deployed by end hosts, requiring no additional infrastructure, no modifications to BGP or routers, nor ISP cooperation. Our work improves and utilizes routing anomaly detection techniques, such as those by Kruegel *et al.* [32] for narrowing down suspicious incidents based on edge network fingerprinting. *Essentially we combine anomaly detection of control plane information* i.e., *routing updates with more conclusive conflicting data-plane fingerprints associated with the network in question.*

In the area of routing anomaly detection and complementary to our work is the recent paper by Lad *et al.* [33] which notifies the prefix owners, in real time, of occurrences of new origin ASes. This method nevertheless can be evaded as changes in origin AS is not necessary for hijacking attacks. Recent work by Qiu *et al.* [39] using cooperation among ASes for detection suffers from the same shortcoming. Our approach is more general and identifies all possible hijacking attack types as described in §3. Boothe *et al.* [10] recently presented detection based on heuristics of short-lived MOAS conflicts, similar to [26]. However timing-based methods is not real-time and may be quite inaccurate due to evasion.

Finally, our work benefits significantly from various fingerprinting approaches to characterize end hosts and networks: *e.g.,* OS-based fingerprinting such as nmap [16] and xprobe2 [49], physical device fingerprinting by identifying clock skews [31], timestamp-based information using TCP and ICMP timestamp probing, as well as IP ID probing used for counting hosts behind NAT [7].

3. An Attack Model of IP Hijacking

We first provide a classification of IP hijacking scenarios. The comprehensive attack taxonomy provides the foundation for our discussion on detection, the explanation for attacker's motivations, and possible evasion attempts. Previous taxonomy [33] addressed only a subset of the attacks.

3.1. Type 1: Hijack a prefix

The most direct way to hijack a prefix is to announce the ownership of IP prefixes that belong to some victim ASes. The BGP neighbors subsequently propagate the route, if it is selected as the best path. Combining routing feeds from multiple vantage points will reveal an MOAS conflict [51], *i.e.,* a prefix with conflicting origin ASes. As an example, there are two AS paths to reach prefix P_1, namely $\{AS_1, AS_2, \cdots AS_n\}$ and $\{AS'_1, AS'_2, \cdots, AS_m\}$. An MOAS conflict occurs if $AS_n \neq AS_m$. MOAS is only one possible indication of IP hijacking. There are also valid reasons for MOAS. Therefore detecting MOAS alone serves only as one possible starting point, and we focus on distinguishing IP hijacking from legitimate MOAS cases. We describe two most common legitimate reasons as illustrated in Figures 1 (a),(b), with the attack shown in Figure 1 (c).

- **Multi-homing with static links:** An AS X uses statically configured route to connect to one of its providers, AS Y. AS X uses BGP to connect to another provider. If the same prefix is announced to both providers, it will appear to have two origin ASes: X and Y.
- **Multi-homing with private AS numbers:** A customer may use BGP to connect to its providers with a private AS number due to shortage of AS numbers. Upon receiving the advertised routes, the provider will eliminate the private AS in the AS paths before announcing them externally. If a prefix is announced to both providers, it will appear to originate directly from the providers, resulting in an MOAS conflict.

Other less common valid reasons for MOAS include Internet Exchange Point (IXP) Addresses, address aggregation, and IP anycast [51]. IP hijacking and router misconfigurations can also lead to MOAS conflicts. The fundamental difficulty arises from the lack of authoritative information on address ownership. Therefore, IP hijacking cannot be identified by simply observing MOAS cases alone as in most previous work which suffers from significant false positives. In §4, we develop an accurate algorithm to distinguish IP hijacking using data plane information.

3.2. Type 2: Hijack a prefix and its AS

Despite several valid reasons for MOAS conflicts, they could still be considered as possible abnormal BGP behavior, requiring further investigation. Stealthy attackers can avoid MOAS by advertising a route to the stolen prefix with

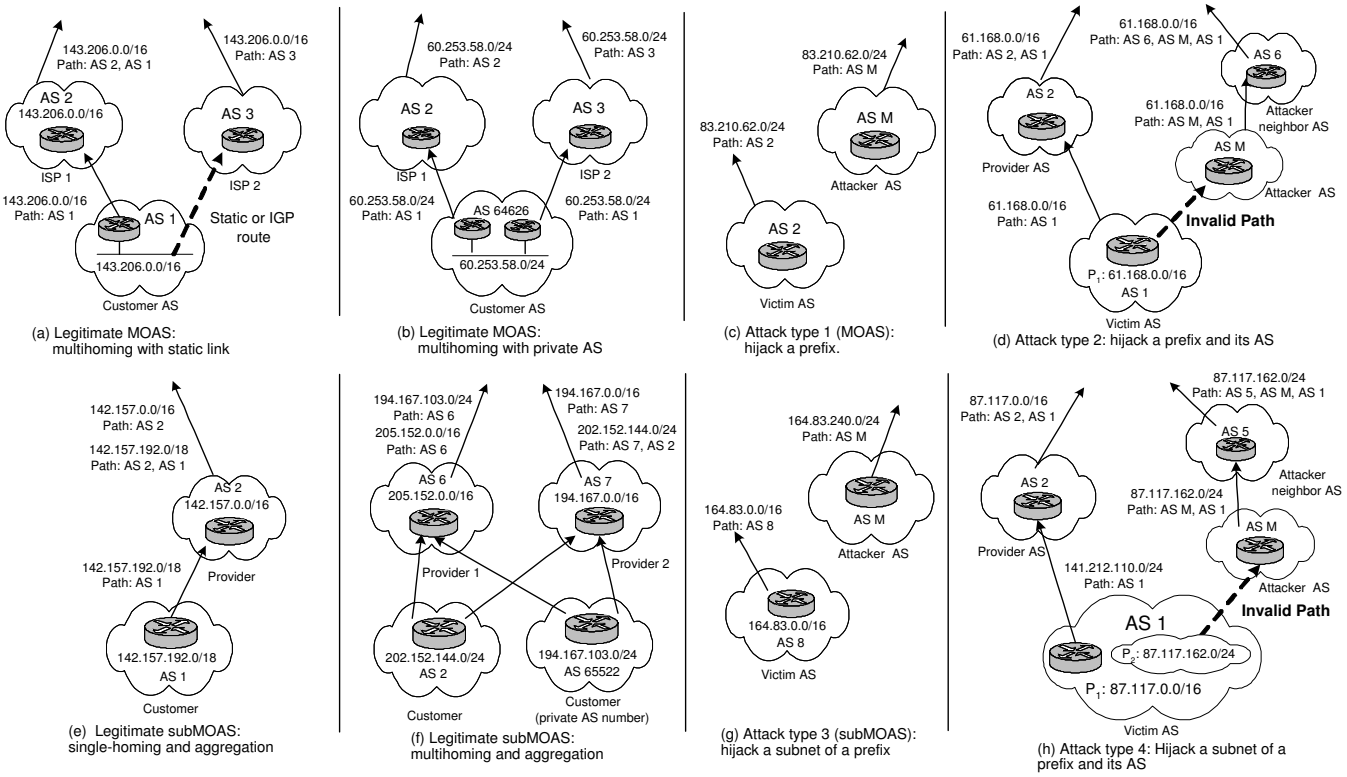


Figure 1. Common legitimate MOAS/subMOAS cases and first four IP hijacking attack types.

an AS path that traverses its own AS to reach the victim AS. It is conceivable that the attacker uses a compromised router to pretend to be the victim AS X by advertising the route with AS path $\{X\}$. However, by default many BGP routers can reject routes with AS paths not starting with the AS number of their neighbor router in the BGP session. To ensure reachability, attackers in AS Y can instead advertise a route traversing its own AS reaching the victim AS X, *i.e.,* with AS path $\{Y, X\}$ for stolen prefixes owned by AS X, as illustrated in Figure 1(d). It is difficult to filter such routes unless a BGP router has accurate knowledge of the BGP topology. By creating false AS edges, attackers can avoid MOAS conflicts, while still achieving the goal of using stolen prefixes to send and receive traffic. Interestingly, some DNS root servers use *IP anycast* for legitimate reasons, matching this attack profile.

3.3. Type 3: Hijack a subnet of a prefix

Another way to avoid MOAS conflicts is to announce a subnet of an existing prefix. For example, an attacker may hijack 129.222.32.0/19 given the existence of 129.222.0.0/16 in the routing table. If there are no other advertisements for this prefix and no filtering for this route, the route is likely to be globally used due to longest prefix based forwarding. For attackers, this approach eliminates the challenging task of making the hijacked route attractive enough to be selected as the best path by other networks. For fear of such attacks, some networks today intentionally deaggregate their address space by announcing many small prefixes such as /24. However, deaggregation severely increases routing table size and may increase routing instability. To capture this routing anomaly, we extend the definition of MOAS to include such origin conflicts involving subnets of prefixes as *subMOAS conflicts*. Similar to MOAS, there are several valid reasons for subMOAS (Figures 1 (e),(f),(g)).

- **Multi-homing with static links:** Similar to MOAS, except that the static routing between the two ASes is configured to reach a subnet prefix, or the other session announces the subnet.
- **Multi-homing with private AS numbers:** For load balancing and redundancy, a customer may multi-home and announce overlapping prefixes to its providers. If private AS number is used, the prefix and its subnet will appear to have the provider's AS as the origin AS, resulting in subMOAS conflicts.
- **Aggregation with single-homing or multi-homing:** A customer C obtains a prefix P from its provider A, who may aggregate P into a larger prefix and advertise only the aggregate with origin AS A to reduce routing table size. If the customer advertises P to its other

provider B, who usually cannot aggregate. A subMOAS conflict results: the bigger prefix with origin AS A and its subnet P with origin AS C. Similarly for single-homing, the provider A announces both the aggregate with origin AS A and P with origin AS C.

3.4. Type 4: Hijack a subnet and its AS

This is the most stealthy hijacking attack shown in Figure 1(h), combining the advantages of both the second and third attack types to avoid both MOAS and subMOAS conflicts. Because of longest prefix matching, attackers can exclusively receive traffic destined to the hijacked prefix. For example, an attacker hijacks a subnet P' of prefix P owned by AS_1. Assume attacker's AS is AS_2. He announces the AS path $\{AS_2, AS_1\}$ for prefix P'. If attacker's neighbors cannot validate whether AS_2 really has a connection to AS_1, they will propagate this route. Since P' is more specific than P, most routers adopt it.

3.5. Type 5: Hijacking on a legitimate path

Instead of forwarding the traffic to the expected next-hop network, the attacker intercepts traffic and originates traffic using the address block of the downstream network.

In the first four attack types, attackers attempt to announce an attractive route, so that routers in different networks on the Internet, even given alternative routes, will still select the hijacking route as the best route. One of the steps in route selection process is preferring routes with the shortest AS path [41]. Note that given the shortest AS path preference, networks topologically close to the victim AS are less likely impacted as they tend to choose the correct routes which are usually shorter than the hijacking routes. Based on the same reasoning, routing tables of networks close to the attacker's AS announcing the hijacking route are more likely polluted. For the fifth attack type, the attacker does not need to announce a new route but merely violate the rule of forwarding traffic. We do not focus on this attack type, but our techniques can also identify it by simply performing traceroute-like probing to show that traffic stops within the malicious network.

Based on the above taxonomy, we highlight two important attack strategies to improve hijacking success and avoid detection. Such understanding helps devise detection techniques. The first strategy is announcing a subnet of an existing prefix, resulting in two advantages. First, if the hijacking route is not filtered[2], each router receiving such route will select it as the best path regardless of its AS path length. Second, simple MOAS-based routing anomaly detection will overlook this type of attack. Note that attackers do not have the incentive to announce a supernet or *covering prefix* (using past terminology [33]), as it makes the hijacked route less attractive. Such announcement is only useful if there exists address blocks within the supernet not covered by existing route announcements. Essentially, it involves allocated but unannounced routes, and can be identified in a similar fashion as unallocated routes through a bogon-like list. We leave this as future work.

Existing work on detecting IP hijacking usually relies on MOAS detection. Aside from false positives caused by legitimate reasons for MOAS, they also suffer from 1false negatives, because attackers could avoid MOAS conflicts using attack type 2 and 4. This is attacker's second strategy with the disadvantage that the announced AS path is longer and may not be selected as the best path. However, announcing a subnet combined with this strategy, as illustrated in type-4 attack, overcomes this disadvantage, creating the most devious attack. We next propose detection techniques.

4. Real-time Detection of IP Hijacking

The focus of our detection algorithm is to distinguish the unique characteristics of IP hijacking attacks based on dataplane properties of the network using the suspected prefix. Operationally it is critical to have a highly accurate detection scheme with low false positives and negatives. The fundamental difference between IP hijacking and valid routing updates lies in the ownership of the IP prefix. For valid MOAS and subMOAS conflicts, despite the disagreeing origin ASes, there is only one owner for the prefix, corresponding to a unique network numbered with the prefix. Traffic sent from anywhere on the Internet destined to the prefix will arrive at the same network location. In the case of IP hijacking, the attacker illegally takes control over the prefix. Traffic sent from different network locations, depending on routing policies, may arrive at either the true owner or the hijacked owner. Such a conflict must exist, as traffic sent from networks topologically close to the true owner or from within the owner network must always arrive at the correct network. This holds even in the case for subMOAS, as IGP routing within the true owner network is unaffected. If hijacking is successful, as evidenced in the suspicious routing updates, networks advertising such updates will choose the hijacked route and reach the attacker network instead. To summarize, the consistency of the destination network is the major criteria underlying our detection algorithm.

4.1. Fingerprint-based consistency checks

When IP hijacking occurs, a given IP address in the hijacked prefix may be used by different end hosts. Similarly, two distinct networks can use the same IP prefix. Therefore we can check the consistency of destination hosts by

[2]In general, prefixes smaller than /24 are likely filtered to limit the size of routing tables [6] based on the longest prefix matching rule [41].

verifying whether their properties match. Note that we do not require end-host cooperation, which can readily provide strong cryptographic authentication information. Instead, we propose a general approach using fingerprints to characterize properties of networks and hosts of the IP prefix. We can generally focus on either host-based or network-based fingerprints. End host properties such as the Operating System (OS), the actual physical device, host configurations (*e.g.,* firewall rules), host software, host services, *etc.* can all constitute host fingerprints serving as signatures to help detect inconsistency. Network characteristics including network configurations like firewall policies, resource properties like bandwidth information, characteristics of routers connecting the network, *etc.* can provide distinguishing network-level fingerprints.

There are several considerations in choosing properties for detecting inconsistency implying real IP hijacking. One challenge is that many networks have firewalls preventing external networks from probing internal hosts. We discuss later in §5.4 how these difficulties are eliminated with assistance from potential victim networks. Probing cost, in terms of network overhead, and probing duration need to be considered. Another consideration is accuracy caused by inherent errors in measurement due to limited precision and external influences. Combining multiple fingerprints can lower both false positives and false negatives. Aside from measurement errors, false positives can also result from legitimate changes in such fingerprints. For example, load balancing in server farms and responses specific to the source IP address (such as those generated from firewalls) may possibly result in conflicting fingerprints. However these uncertainties can be identified beforehand by comparing fingerprints from multiple probing places, so that hosts with such uncertainties are excluded from fingerprinting checks. False negatives may result from distinct networks or hosts with identical fingerprints. Using multiple fingerprints and choosing discriminating properties such as host uptime and physical device fingerprints [31] significantly reduce the likelihood of such coincidences.

Intuitively, attackers can use two methods to *evade detection*: faking the similar fingerprints and forwarding all probing packets to victim networks. Faking network or host properties is challenging given the use of diverse properties, especially if properties are host-specific and vary continuously, *e.g.,* clock skew, uptime, and IP ID number, or those associated with available resources such as bandwidth, since faking more resources than available is challenging. Forwarding all packets to victim networks is also infeasible. First, there is little incentive for attackers to spend precious bandwidth for forwarding, as it may disrupt their attack activities. Moreover, such forwarding can be easily detected with traceroute-like tools, as it creates abnormal patterns in packets' forwarding path. The abnormal forwarding path can also be identified simply by comparing its actual AS path with the path of prefixes from the same origin AS with the same announced AS path but are unlikely hijacked. According to our experiment where we randomly choose and traceroute to a pair of prefixes with the same AS paths from 3026 distinct ASes, we find that 95.6% of prefix pairs have matching AS forwarding paths. Hence, packet forwarding will cause the attacker's AS path to exhibit unusual deviation from the normal paths, thus making evasion very difficult to succeed.

As initial examples, below we discuss the use of host OS, IP ID, TCP and ICMP timestamp based fingerprints. Note that other fingerprinting techniques can be easily incorporated to improve the accuracy, *e.g.,* bandwidth estimation [24, 42, 34] and physical fingerprints [31, 19].

Host OS properties: Attackers are likely to use a dissimilar OS or configure the OS differently in terms of open ports compared to legitimate users of the network. Even if the host is configured in the same way, the IP addresses used within the prefix may be different. Using popular remote OS probing tools like Nmap [17, 16] and xprobe2 [49], such host information can provide identifying fingerprints.

IP Identifier probing: IP header includes a 16 bit identifier (IP-ID) field, designed to be unique for each IP datagram to help IP fragment reassembly. A common implementation is "global" IP ID, *i.e.,* incrementing IP ID by one for every outgoing packet, regardless of its destination. Similar to Bellovin's work on using IP IDs to count hosts behind NAT [7], we use them to verify whether two machines are the same. We continuously send probe packets simultaneously to the same IP suspected to be hijacked from two locations. In the case of no hijacking, packets reach the same machine. Because of the global incremental properties of most implementations, the IP IDs in reply packets exhibit roughly alternating incrementing patterns. If the address is hijacked, probe packets reach distinct machines, and IP ID in reply packets appear unrelated. Several difficulties exist: Some implementations randomly set the IP ID field, reset it to 0, or set it to be uniquely increasing for each destination.

TCP timestamp probing: The TCP timestamp option specified by RFC 1323 [29], used for measuring round-trip times, can give estimates of the time when the machine was last rebooted. TCP timestamp is set based on a machine's internal clock which is reset upon system reboot [31]. This clock runs at a certain frequency ranging from 1Hz to 1000Hz. Knowing this frequency and the TCP timestamp, we can infer the uptime. If the inferred uptime obtained from different locations for the same IP is sufficiently diverse, a hijacking attack may have succeeded.

ICMP timestamp probing: Sending ICMP timestamp requests to the target machine will solicit the ICMP timestamp replies containing the system time of the target machine reported in millisecond. Because not all the machines connected to the Internet are synchronized with NTP, we

Attack type	Routing updates monitored	Detection techniques
1. (Hijack prefix)	MOAS updates	Fingerprint-based consistency check (FP check)
2. (Hijack prefix, AS)	All updates	Edge, geographic, and relationship (EGR) constraints, FP check
3. (Hijack subnet prefix)	SubMOAS updates	Customer-provider (C-P) check, reflect-scan
4. (Hijack subnet prefix, AS)	New, nonsubMOAS prefixes	Edge, geographic, and relationship constraints, reflect-scan
5. (Hijack a legitimate path)	Not triggered by updates	Fingerprint-based consistency check

Table 1. Summary of detection techniques.

can expect two different machines likely to have noticeable differences in their clock time.

Note that none of the above methods guarantees to distinguish two different machines, but their combination reduces false positives and negatives. In what follows, we discuss the techniques of detecting IP hijacking attacks for each of first four attack types summarized in Table 1.

4.2. Type 1: Detect prefix hijacking

This type of IP hijacking has the characteristic of MOAS conflicts as shown in Figure 1(c). The essence of our attack detection is to check whether the prefix originated by multiple ASes has consistent data-plane signatures. To verify this, we send probing packets to the same IP in the suspect prefix and use the previously discussed fingerprint-based consistency checks. The process is outlined here: 1) For each prefix involved in MOAS conflicts, find its AS paths from BGP data. 2) Find a live host in the prefix serving as the probing target. 3) Select probe locations so that packets reach conflicting origin ASes. 4) Perform probing using techniques described in §4.1. 5) Analyze obtained fingerprints to identify mismatches.

One challenge is to select probe locations such that probe traffic reaches different origin ASes. We use the current best AS paths from public BGP data to guide the selection. For example, assume prefix P_1 announced by both AS_1 and AS_2 has two AS paths reaching it: $\{AS_5, AS_3, AS_1\}$ and $\{AS_6, AS_4, AS_2\}$. Probe locations are chosen to be as close to the origin AS as possible – AS_1 is preferred over AS_3. Traffic may not conform the expected AS paths, because of inconsistency between the data and the control plane or disagreeing AS paths within the same AS caused by tie-breaking. Thus after selecting the probe locations, we verify that traffic arrives at the intended AS. This is nontrivial, as translating a router IP from traceroute to AS numbers may result in multiple ASes [35]. Furthermore, traceroute may not reach the destination. We use either of the following two criteria to ensure that packets with high probability reach the origin AS, *e.g.,* AS_1. 1) The traceroute IP-level path contains a router whose IP address is originated by AS_1 only. 2) The traceroute IP-level path contains a router whose IP is originated by the nearest possible AS before reaching AS_1, *e.g.,* AS_3. In addition, AS_3 should not appear within the AS path originated by other conflicting origin ASes for the prefix, *e.g.,* AS_2.

4.3. Type 2: Detect prefix and AS hijacking

We now address the second attack type shown in Figure 1(d) and described in §3.2. Attackers avoid MOAS and subMOAS conflicts by retaining the correct origin AS and creating at least one fake AS edge. For example, attackers append the correct origin AS after its own AS in the AS path, creating a fake AS edge between its network and the victim network. Thus the AS path is inconsistent with the data plane. Our approach still relies on data-plane fingerprinting, but we enhance it by first using the following checks to reduce false positives, especially given that any update may be a possible attack in this category. Unlike the previous approach [33], our techniques are applicable independent of the position of the fake edge within the AS path.

- **Edge popularity constraint:** We identify fake AS edges by computing AS edge *popularity*. If an AS edge has never been observed in previous route announcements or few prefixes use routes traversing this edge, it is highly suspicious.

- **Geographic constraint:** Similar to the above constraint, a fake AS edge can connect two geographically distant networks. BGP peering sessions between two ASes almost always occur between routers physically colocated. Thus, an AS edge corresponding to two distant networks signals an alarm.

- **Relationship constraint:** Extending the path constraint in previous work [32], we identify obvious violations of routing policies within the AS paths using inferred AS relationships [18].

We elaborate two improvements for the geographic constraint checking. First, rather than using data from registries such as whois, which provides only a single location for each AS, we exploit more fine-grained prefix locations. Freedman *et al.* [14] showed that roughly 97% of all prefixes announced by stub ASes were announced from the same location. Second, we build up a location set for each

AS consisting of all distinct locations of its originated prefixes. The distance between ASes is the minimum distance between every pair of locations in their sets.

4.4. Type 3: Detect prefix subnet hijacking

This attack shown in Figure 1(g), elaborated in §3.3, occurs when the attacker hijacks a subnet of victim's prefix by announcing it as originating from its *own* AS, resulting in a subMOAS conflict. This approach is more stealthy, as it does not create obvious MOAS conflicts and is also preferred by attackers as more networks will adopt the hijacked route. Our detection scheme first identifies subMOAS conflicts and then excludes the cases directly involving ASes with customer-provider relationships using the *customer-provider check* explained below. Finally, we use fingerprint checks to analyze the remaining cases.

The customer-provider check operates based on the assumption that providers will not intentionally hijack customer's routes due to lack of economic incentives and the ease of discovering such attacks through traceroute-like probing. Similarly, customers are incapable of hijacking provider's routes because traffic needs to first traverse the provider, and providers can easily detect such routing announcements. Given this justification, we developed a simple yet very robust and accurate technique for inferring customer provider relationships elaborated in the extended version of this writing [25].

The customer-provider check does not deal with conflicts involving ASes with non-customer-provider relationship, *e.g.,* Figure 1(g). Thus, we still need to resort to fingerprinting for the remaining cases, but the biggest challenge is that the longest prefix match rule causes all traffic be routed to more specific hijacked prefix regardless of the probe location unless we can find the probe location inside the victim AS, so that the fingerprinting packets will be routed using IGP or the probe location is inside the customer or provider of the victim AS that use static links to connect to the victim AS and are thus unaffected by hijacking.

Given limited probe locations, neither condition is easily satisfied. We devise a new probing technique called *reflect-scan* for fingerprinting the victim network. Our method is inspired by the TCP Idlescan technique [15] implemented in Nmap [17]. The basic idea is to make use of predictable IP ID increment in IP packet and IGP routing within the victim AS which is unaffected by polluted BGP routes. We use IP spoofing to solicit traffic inside the victim AS. As an example, let us assume a typical hijacking scenario where AS_1 has a large prefix P_1, *e.g.,* 195.6.0.0/16. AS_2 is malicious and hijacks subnet P_2 of P_1, *e.g.,* 195.6.203.0/24. Our probing technique works as follows (depicted in Figure 2):

1. Find a live host (H_2 or H_2', *e.g.,* 195.6.203.3) in the hijacked prefix P_2 with predictable IP ID values (*e.g.,* increment by 1) and has little outgoing traffic. Later we relax this requirement, but for ease of explanation, let's assume the host has no outgoing traffic.

2. Find a live host (H_1, *e.g.,* 195.6.216.26) with IP in P_1 but not in P_2. More generally H_1 can be any live host in any prefix except P_2 originated by AS_1.

3. Assume that due to hijacking, there exists a host H_2' in attacker's network AS_2 and a host H_2 in the victim's network AS_1 with the same IP 195.6.203.3. Since H_1 and H_2 are in the same AS, packets from H_1 to 195.6.203.3 is routed using IGP, *e.g.,* OSPF and reach the correct host H_2. In contrast, if probing packets are sent from outside of AS_1, they are routed using the polluted BGP routes and reach H_2' instead, since P_2 is more specific than P_1.

4. Step 1-2: Send probe packets to 195.6.203.3 and record its current IP ID value. Remember because our probing comes from outside of AS_1, in the case of hijacking, traffic is routed to the potentially hijacked prefix and the IP ID value is that of attacker's machine, *i.e.,* H_2'.

5. Step 3-5: Send a SYN packet to an open port of H_1 (195.6.216.26) with a spoofed source IP of H_2 (195.6.203.3). H_1 should reply with SYN/ACK to the spoofed source. Because IP address of H_1, 195.6.216.26 and 195.6.203.3 are inside the same AS, the response should reach H_2 in AS_1. After receiving this unsolicited SYN/ACK, H_2 sends back a RST and increases its IP ID value by one.

6. Step 6-7: Reprobe 195.6.203.3 and obtain the current IP ID value of H_2 or H_2' (depending on whether there is a hijacking attack). If the IP ID value in the reply is only increased by 1, it has not sent any packets. Very likely it did not receive H_1's SYN/ACK packet (Figure 2(a)), indicating a possible hijacking attack.

As demonstrated by the Figure 2, the target host with IP 195.6.203.3 responds differently depending on whether the subMOAS is caused by hijacking. If there is no hijacking, the target host (in this case H_2) receives reply SYN/ACK packets from H_1, causing its IP ID number to be incremented by the number of spoofed packets received (Figure 2(b)). Otherwise, the IP ID value of the target host (H_2') does not increase (Figure 2(a)). We now relax the restriction that H_2 needs to be idle to improve the robustness of the reflect-scan test. In reality, during our probing, H_2 may also send out other packets not triggered by our probing. To reduce false negatives, we repeat the test, send multiple spoofed packets, and use information of average increase rate of H_2's IP ID value to detect hijacking.

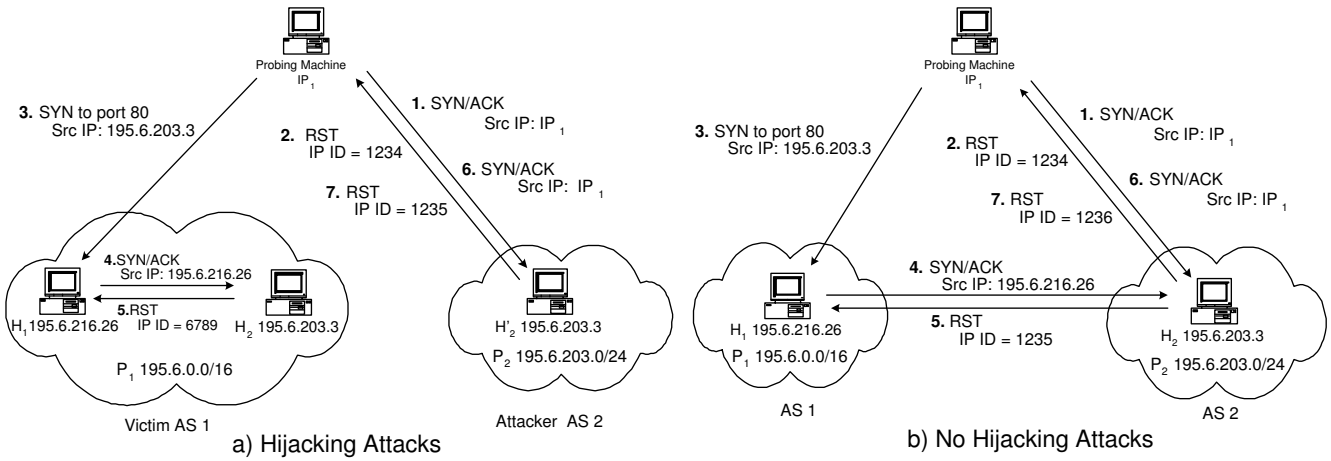


Figure 2. Reflect-scan: detection of hijacking subnet of a prefix (type-3 and type-4 attacks).

4.5. Type 4: Detect subnet, AS hijacking

The most devious attack type as illustrated in Figure 1(h), discussed in §3.4, occurs when the attacker hijacks a subnet *and* retains the correct origin AS. Similar to type-2 attack, there is no MOAS nor subMOAS conflicts. To detect this attack type, we continuously monitor new prefixes that are subnets of existing prefixes in the routing tables. If they do not cause a subMOAS conflict, they may fall into this category. We apply similar checks for type-2 attacks: edge popularity constraints, geographic constraints, and relationship constraints to reduce false positives and then apply reflect-scan probing to deal with the remaining cases that violate any of the checks. Note that we can still achieve real-time monitoring given that the space of suspicious cases for this attack only includes new prefixes not present in the current routing tables.

5. Implementation of Real-Time Monitoring

One of the most important properties of our system is real-time monitoring. As hijacking sometimes lasts only for a short time period to avoid detection, a real-time detection system is essential to defend against malicious attacks in a timely manner, reduce the damage incurred, and identify the culprit. We demonstrate next how we achieve the real-time capability in our prototype system.

5.1. System architecture

We developed a prototype system aimed at online detection of anomalous BGP routing updates and selective lightweight active probing to gather data-plane fingerprints for identifying hijacking attacks. Figure 3(a) illustrates the architecture of the prototype. It consists of three modules.

1. **Monitor Module** processes BGP updates in real time to identify potential IP hijacking. The classifier in this module depicted by Figure 3(b) classifies each update into two types: valid and anomalous. For the latter case, it groups them into four hijacking types described in §3. Then both the type and the update information (*i.e.,* prefix and AS path) are fed into the Probing Module for further analysis.

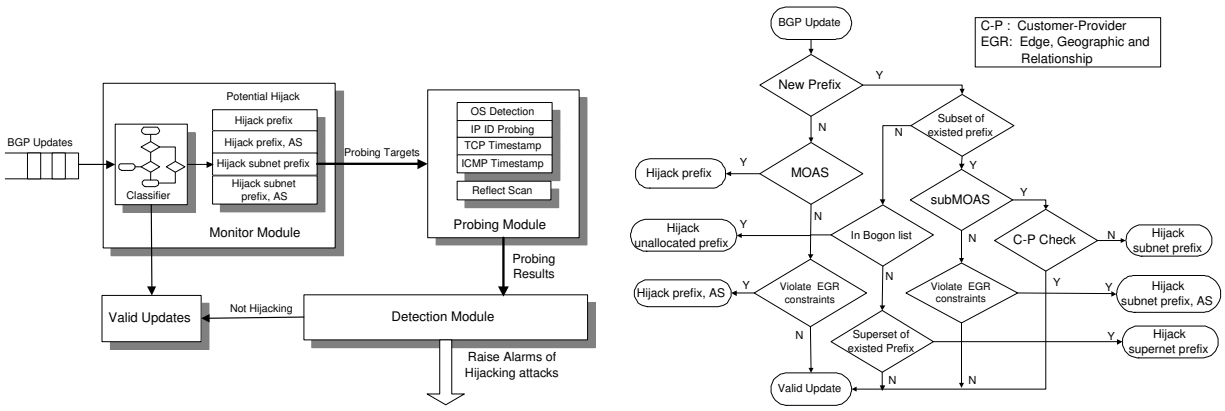
2. **Probing Module** takes input from the Monitor Module and selects corresponding probing techniques. It chooses the appropriate probing locations and launches probing (*e.g.,* OS detection, IP ID reflect-scan) to the target prefix. Probe results are sent to the Detection Module.

3. **Detection Module** analyzes and compares the probe results to identify suspicious updates.

5.2. Experimental methodology

BGP data set: We use BGP update data primarily from two sources: University of Oregon RouteViews Server [2] which peers with 57 BGP routers in 46 ASes and our own route monitor peering with 7 BGP routers in 7 distinct ASes. RouteViews data has better coverage; however, its update files have a two-hour lag. Thus, we obtain real-time BGP updates from our own monitor. Because of the larger number of feeds in RouteViews data, we use it to evaluate our system's scalability and efficiency in processing large volumes of updates. For update-triggered response, we use data from our own monitor to study timely responses to anomalous updates.

Probe location selection: We use the Planetlab testbed [1] (consisting of 642 machines in 179 different ASes including 3 tier-1 ISPs) as the candidate probing



(a) (b)

Figure 3. (a) System architecture for real-time detection of IP prefix hijacking attacks, (b) Classification of hijacking types.

Attack Type	Anomalous updates	Max rate /15 min	Avg rate /15 min
1	Hijacking a prefix (MOAS conflicts)	0.42	0.08
2	Hijacking a prefix and its AS	28.17	1.60
3	Hijacking a prefix subnet (subMOAS)	2.92	0.16
	After Customer-provider check	0.86	0.09
4	Hijacking a prefix subnet and its AS	3.74	0.33
	After EGR constraint check	0.15	0.01

Table 2. Anomaly rate of updates/BGP feed (1 day of RouteViews data).

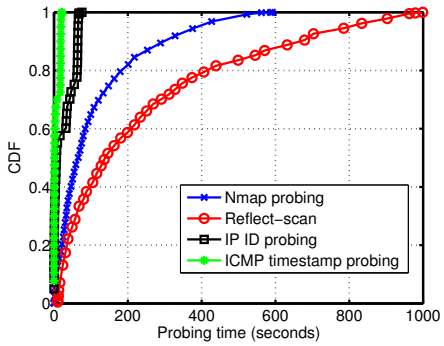


Figure 4. The probing time distribution.

places for both type-1 and type-2 attacks. Note that reflect-scans can be conducted anywhere as long as IP spoofing is permitted. Altogether we are able to find probe locations for 89% MOAS cases and 75% type-2 attack cases.

Live IP addresses: Live IP addresses for probing are collected by combining locally collected DNS and Web server logs. We also use reverse DNS to look up authoritative DNS servers and mail servers of various domains. In addition, we conduct light-weight ping sweeps for a very limited address range. Currently our list contains 1,165,845 unique IP addresses allowing us to find target hosts for 70.3% of all prefixes in MOAS conflicts, 55.2% for type-2 attacks, 71.0% for subMOAS conflicts, and 90.1% for type-4 attacks.

Geographic information of prefixes: In our current implementation, we use the NetGeo [12] database, developed by CAIDA to map IP addresses and AS numbers to geographic locations, providing detailed longitude and latitude values for 98.4% of all 198,146 prefixes. We plan to explore other techniques [37].

5.3. Real-time detection

To understand our system performance of real-time detection, we measure BGP update rate, detected anomaly rate, the probing time of different attack types, and the memory usage of the prototype. We use RouteViews data for its better coverage. We simulate update processing by feeding RouteViews Data into the Monitor Module.

Update rate: The update rate determines the workload of our system. We take one week's updates (from 04/01/2006 to 04/07/2006) from RouteViews and calculate the average update rate for each BGP feed over a period of the seven days. The maximum update rate is 12 updates/second, the minimum rate is less than 1 update/second, and the average rate is about 2.45 updates/second. Because the classification process does not involve active probings, even a desktop machine can easily handle many BGP feeds.

Anomaly rate: The anomaly rate is the number of suspicious updates per unit time after classification. This determines the rate of active probing to detect hijacking attacks. Table 2 show the anomaly rate for each of four hijacking types using one day of RouteViews data. As illustrated in the figure, the average anomaly rates for all attack types are usually small leading to relatively low overhead. Given that all the probing can be done in parallel, our system can easily

scale to monitoring an even larger number of BGP feeds.

Probing time: For each anomalous BGP update, the system performs active probing to identify IP hijacking. In the current implementation, we adopt four probing techniques: Nmap scan, IP ID probing, ICMP timestamp probing, and reflect scan. Based on one week's experiments, probing duration distribution is shown in Figure 4. In general, the probing takes less than 10 minutes, with the average time of less than 3 minutes for Nmap, and less than 4 minutes for reflect-scan (mostly due to the overhead of finding idle hosts and open ports). Our prototype implementation can easily handle much a higher anomaly rate.

5.4. Deployment and Operational issues

Our system requires neither end-host cooperation nor modification of existing protocols, making it easily and incrementally deployable in the current Internet. We highlight two important operational issues here. First, although any network can deploy our system to detect all potential hijacking attacks on the Internet, in general each AS starts probing only when it suspects its own network is under hijacking attacks. Thus no flash crowd will occur toward a victim network. On the other hand, our system can also be deployed on a few centralized servers monitoring hijacking attacks for the entire Internet and notifying victims via some out-of-band channels [33]. Second, asking victim ASes to probe the suspicious prefix can offer additional advantages. Fingerprinting may be limited by firewalls in some destination networks blocking external probing packets. *However, if the probing is initiated from inside the network to identify if external networks can reach its own network correctly, the probing packets are usually permitted by the firewalls, facilitating the collection of conflicting fingerprints.* Thus, ASes that are suspicious of being hijacked (informed by the monitoring module)can select one probing location inside its own network and another location near the culprit AS to collect confirming evidence of the hijacking attack.

6. Evaluation

We next describe results in data probing and evaluate the effectiveness of the detection system by illustrating interesting results collected during a two weeks time period, including validation using IP anycast of root DNS servers, a special legitimate case of IP address hijacking, as well as correlation with known spam source IPs.

6.1. Monitoring results

We now present some interesting results obtained from over 253 hours of real-time monitoring across two weeks (We do not have results for the remaining hours due to network problems with the BGP monitor.) The type and number of anomalies are summarized in Table 3. The rate is averaged over all 7 feeds monitored. We implemented probing for IP-ID and ICMP timestamp on Planetlab using Scriptroute [44] and reflect-scan using hping [43]. Probing to the same IP across different paths are conducted roughly simultaneously.

Suspicious MOAS conflicts and type-2 attacks: Since we use similar probing techniques to identify suspicious MOAS conflicts (type-1 attacks) and type-2 attacks, we present them together here. We group the observed suspicious fingerprinting results into the following categories.

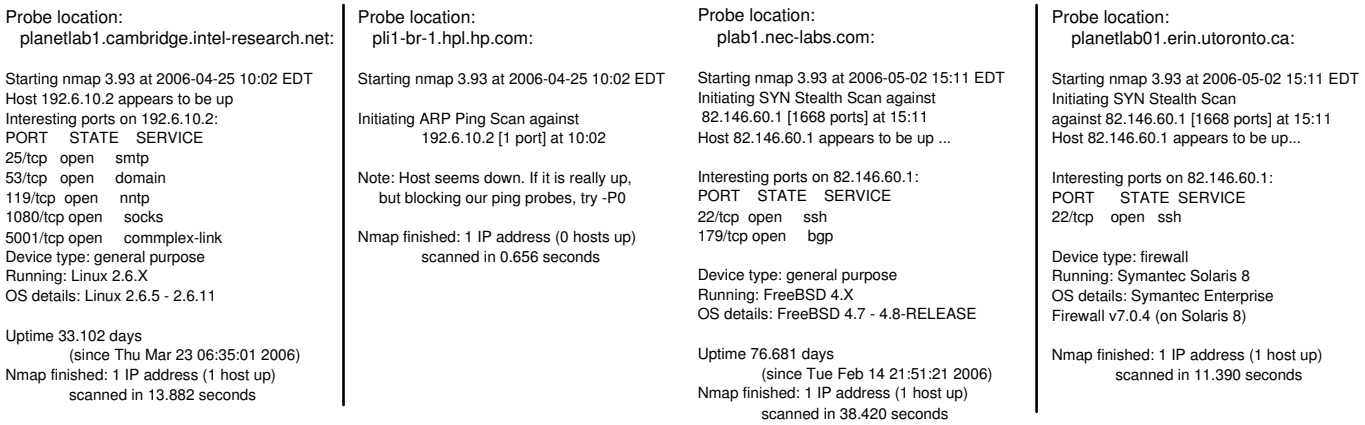
- **Different liveness:** If the host appears alive from one location, but unresponsive somewhere else, it may be a real hijacking attack barring intermediate network problems and special firewall policies (Figure 5(a)).
- **Different Operating Systems:** Figure 5(b) is a suspicious type-2 attack with different Nmap-inferred OS.
- **Different open ports:** Figure 5(b) exhibits inconsistency in open services: BGP (port 179).
- **Different TCP timestamps (uptime):** The host probed from one location may support TCP timestamp, but not from another location, *e.g.,* Figure 5(b). We also observed significantly different uptime values (Figure 7(a)).
- **Different ICMP timestamps (local time):** Figure 6(a) indicates significantly different ICMP timestamp values.
- **Different IP IDs:** For systems with globally incrementing IP-ID patterns, there is a significant difference in IP ID return values or patterns, *e.g.,* Figure 6(a).

Suspicious subMOAS conflicts and type-4 attacks: For suspected subMOAS (type-3) and type-4 attacks, we use reflect-scan to identify hijacking incidents. The following is a found example of a suspicious subMOAS conflict with the probing results using reflect-scan shown in Figure 6(b). Prefix 193.140.140.0/24 is announced by AS15390 at 21:27 on April 25th, 2006, which has a subMOAS conflict with prefix 193.140.0.0/16 owned by AS8517.

1. 193.140.140.8 (H_2) in the subnet 193.140.140.0/24 is selected as the idle host, because its IP ID increases regularly by one and has the open port 21.
2. We send SYN/ACK packets to port 21 of H_2 to verify that H_2 responds with RST.
3. The live host 193.140.0.2 (H_1) in the larger prefix 193.140.0.0/16 but not in the subnet is chosen as the *reflect host* with an open port 514.

Anomalous update type	Total number	Average rate (/15min)	Suspicious updates (after fingerprinting)
Hijack a prefix (MOAS conflicts)	3685	0.52	332
Hijack a prefix and its AS	17205	2.43	594
Hijack a subset of a prefix (subMOAS conflicts)	3380	0.47	151
Hijack a subset of a prefix and its AS	1195	0.17	85

Table 3. Suspicious updates detected during 2 weeks' monitoring after various constraint and fingerprint checking.



(a) Different liveness of the target host in an MOAS conflict
192.6.10.0/24 is announced by AS 2856 and AS 786.

(b) Difference in response fingerprints of suspicious type 2 attack
82.146.60.0/23 is announced by AS 25486. The first hop <8804 2548> is used only by 6 prefixes and the edge distance is 8968 kilometers

Figure 5. Conflicting fingerprints of Nmap probing (a) type-1 attacks, (b) type-2 attacks. The first line indicates the probe location.

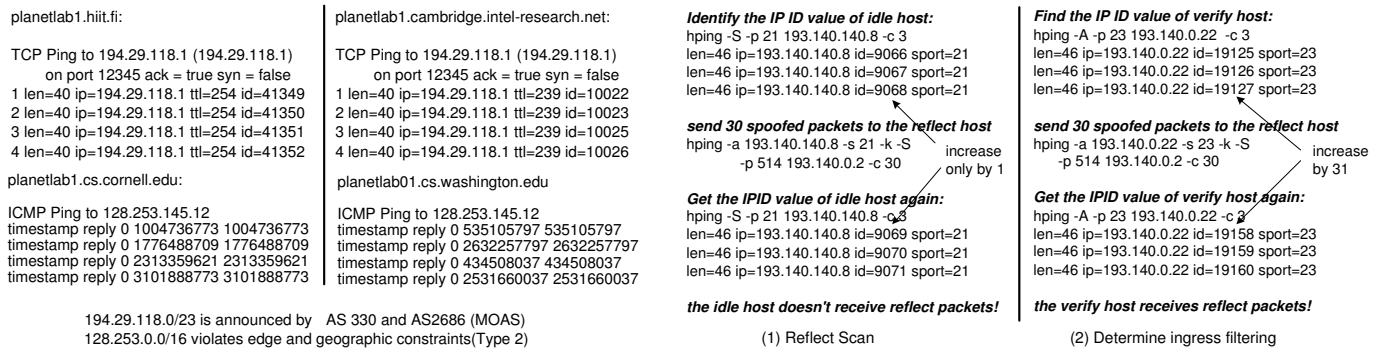


Figure 6. (a) Different IP ID values and ICMP timestamp values (potential type-2 attacks). (b) A reflect-scan example (type-3).

4. Compare the idle host H_2's IP ID values before and after sending spoofed packets to reflect host H_1 with source IP of H_2. We found the idle host did not receive 30 reflected packets, which may be dropped or delivered somewhere in AS8517 (Figure 6(b)(**1**)).

5. To verify that the test did not fail due to ingress filtering[3] which may cause the idle host not to receive spoofed packets, we select another idle host 193.140.0.22 similar to H_1 to be the *verify host*.

6. We do the similar test to check for ingress filtering. By comparing the IP ID value of the verify host before and after sending spoofed packets using verify host as the source IP to the reflect host, we find that it receives all reflected packets indicating the lack of ingress filtering in AS8517 (Figure 6(b)(**2**)).

[3]If AS8517 has ingress filtering that filters out incoming traffic with source IP from inside the AS, the spoofed packet cannot reach the reflect host, and no reflect packets will be generated.

Since we are confident that reflected packets are sent to the idle host (step 6) and the idle host responds to SYN/ACK packets (step 2), the idle host's IP ID value should be increased, if it received them. Thus, we can conclude that this case fails reflect-scan and is highly suspicious as a real hijacking attack.

6.2. Validation using IP anycast

For load balancing and robustness considerations, a number of root name-servers are deployed using IP anycast [22]. IP anycast, defined in RFC 1546 [38], is an internetwork service where multiple severs support the same service under the same IP address. Currently, 5 out of all 13 DNS root servers (C, F, I, J and K) are using IP anycast, each with multiple servers in different locations [30, 3]. IP anycast for root DNS is achieved by announcing the same prefix and AS number from multiple locations on the Internet, identical to hijacking both the prefix and its AS (type-2 attack). However, this is a valid case; thus, we use it for validation.

Across two weeks' monitoring, our system successfully captured suspicious updates from four root servers (F, I, J and K), with the exception of the C-root server (c.root-server.net in prefix 192.33.4.0/24 with origin AS2149). After investigating the updates for the C-root server, we find that it only have one upstream provider AS174 which is a large tier-1 ISP. Since AS174 also has a location near to AS2149, the updates for C-root server do not violate the geographical constraint and therefore cannot be captured using that constraint alone. Figure 7 illustrates an example of the F-root server (f.root-servers.net) detected by our system. The IP address of the F-root server is 192.5.5.241 in prefix 192.5.5.0/24 announced by AS3557. Figure 7 clearly shows that probing from two different Planetlab nodes actually reaches two distinct machines, validating our fingerprinting approach.

6.3. Validation using spam source IPs

Hijacked IP prefixes are believed to be often used by spammers to send spam. Ramachandran *et al.* [40] recently found that non-trivial amount of spam was sent from short-lived, possibly hijacked IP prefixes by analyzing network-level behavior of spammers using spam collected via a "spam sinkhole" or a honeypot-like spam domain. To validate our work, we correlate our identified suspicious hijacking attempts with the source IPs of the spam data in [40] for the same two week time period.

Table 4 summarizes the correlation results, where "matched prefixes" indicate prefixes appearing in both data sets bounded by a time window of 3 days. To understand the time-related spam behavior, the number of matched prefixes within some time window after detecting suspicious hijacking attempts is also shown, with the CDF of time intervals illustrated in Figure 8. Even though the actual percentage of hijacked prefixes used for spamming and the percentage of spam sources using hijacked prefixes are not known, this result still shows non-negligible correlation between highly likely hijacked prefixes and spam sources, indicating a potential spamming mode of exploiting routing infrastructures.

Attack Type	Number of suspicious prefixes	Number of matched prefixes	Number of matched prefixes within the time window		
			1 hour	6 hours	1 day
1	332	28	19	25	25
2	594	91	34	74	87
3	151	10	4	8	10
4	85	11	5	10	11

Table 4. Correlation between detected suspicious prefixes and spam sources.

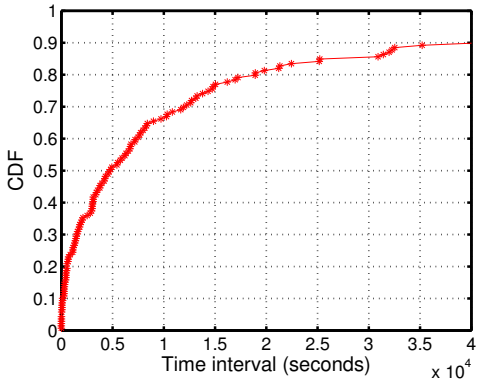


Figure 8. CDF of time intervals between identification of suspicious updates and the arrival of spam.

6.4. Reducing false positives and negatives

Compared to most of previous work in detecting IP hijacking, which solely relies on identification of MOAS conflicts as the indication, our system successfully reduces both the false positive and the false negative rate. Because of a wide range of valid reasons for MOAS and subMOAS, alarming every MOAS or subMOAS conflict will cause excessive false notifications, which may overwhelm network administrators and also hide important alarms for real hijacking attacks. In contrast, our scheme provides more definitive evidence for suspicious hijacking attacks by checking the fundamental difference, *i.e.,* data plane inconsistencies, between valid and hijacking cases and therefore greatly reduces the likelihood of false positives. For ex-

crt1.planetlab.umontreal.ca:

Starting nmap 3.93 at 2006-05-03 21:42 EDT

Interesting ports on 192.5.5.241:
PORT STATE SERVICE
53/tcp open domain

Device type: general purpose
Running: FreeBSD 5.X
OS details: FreeBSD 5.3

Uptime 11.573 days
(since Sat Apr 22 07:56:43 2006)

Nmap finished: 1 IP address (1 host up)
scanned in 26.225 seconds

planetlab-1.eecs.cwru.edu:

Starting nmap 3.93 at 2006-05-03 21:42 EDT

Interesting ports on 192.5.5.241:
PORT STATE SERVICE
53/tcp open domain

No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi)

Uptime 14.963 days
(since Tue Apr 18 22:35:51 2006)

Nmap finished: 1 IP address (1 host up)
scanned in 23.554 seconds

(a) Difference of Nmap fingperints of F root server

crt1.planetlab.umontreal.ca:

TCP Ping to 192.5.5.241 on port 12345
ack = true syn = false
1 len=40 ip=192.5.5.241 ttl=56 id=29577
2 len=40 ip=192.5.5.241 ttl=56 id=29578
3 len=40 ip=192.5.5.241 ttl=56 id=29579
4 len=40 ip=192.5.5.241 ttl=56 id=29580
5 len=40 ip=192.5.5.241 ttl=56 id=29581

crt1.planetlab.umontreal.ca:

ICMP Ping to 192.5.5.241 (192.5.5.241)
timestamp reply 0 2487465 2487465
timestamp reply 0 2487539 2487539
timestamp reply 0 2487625 2487625
timestamp reply 0 2487697 2487697
timestamp reply 0 2487769 2487769

planetlab-1.eecs.cwru.edu:

TCP Ping to 192.5.5.241 on port 12345
ack = true syn = false
1 len=40 ip=192.5.5.241 ttl=251 id=60654
2 len=40 ip=192.5.5.241 ttl=251 id=47890
3 len=40 ip=192.5.5.241 ttl=251 id=61606
4 len=40 ip=192.5.5.241 ttl=251 id=624
5 len=40 ip=192.5.5.241 ttl=251 id=59346

planetlab-1.eecs.cwru.edu:

ICMP Ping to 192.5.5.241 (192.5.5.241)
1no response
2no response
3no response
4no response
5no response

(b) Difference in IP ID and ICMP timestamp probing

Figure 7. Probing signatures for the F-DNS root server (legitimate type-2 case).

ample, from Table 3, during the experiment period, 3685 MOAS conflicts occurred (more than 1 MOAS conflict per hour), which may be too frequent to be handled. After the fingerprinting check, only 332 highly suspicious cases are left for further investigation – a huge reduction in false positives. On the other hand, MOAS-based detection schemes also suffer from significant false negatives because attackers can evade MOAS conflicts (and thus the detection) altogether by simply choosing the remaining 3 types of attack schemes in section 3. Although our system cannot completely remove false positives and false negatives that may stem from changing fingerprints, server farms, and faking fingerprints, it successfully minimizes such possibilities with a variety of verification schemes, *e.g.,* using multiple fingerprinting techniques (Note that although not adopted in our system, resource-based or physical device based fingerprinting can be incorporated easily for more accuracy), edge popularity check, relationship check, *etc.*, making our system more efficient and incrementally deployable on the current Internet.

7. Discussions and conclusions

We discuss several limitations with our work and plans for future improvement. First, our system is triggered based on anomalous routing updates. However, hijacking may not be visible on the control plane, as the data plane is not guaranteed to be consistent with advertised routes. We plan to explore continuous monitoring and performance-triggered probing to augment the current approach. We also plan to analyze in more detail the accuracy of fingerprinting techniques. A second more serious limitation is that probing will be limited by limited availability of vantage points and increasing deployment of firewalls. We plan to explore the coverage based on the probing location and network-based fingerprints. Note that our system can be deployed either by individual networks or by a centralized system. In the latter case, we have demonstrated the scalability of the system, but we did not address the issue of reliably notifying the victims. This is challenging as the victim may not be easily reached due to the impact of IP hijacking. Work by Lad *et al.* [33] suggests the use of diverse paths, without providing absolute guarantee.

In summary, we present a framework for accurate, real-time IP address hijacking detection. Our work is based on the novel insight that a real hijacking attack will result in conflicting data-plane fingerprints describing the hijacked network. Using this key difference, we can significantly reduce both false positives and false negatives and more confidently identify IP hijacking without sacrificing efficiency. This is the first work exploiting the consistency between data-plane and control-plane information to identify IP hijacking attacks. Our system can be incrementally deployed without modifying any infrastructure nor requiring support from networks. We have demonstrated the effectiveness and efficiency of a prototype system using real data.

References

[1] PlanetLab. `http://www.planet-lab.org/`.

[2] University of Oregon Route Views Archive Project. `http://www.routeviews.org`.

[3] J. Abley. Hierarchical Anycast for Global Service Distribution. ISC's Technical Note, 2003.

[4] W. Aiello, J. Ioannidis, and P. McDaniel. Origin Authentication in Interdomain Routing. In *Proc. CCS*, 2003.

[5] A. Barbir, S. Murphy, and Y. Yang. Generic Threats to Routing Protocols. IETF Draft: draft-ietf-rpsec-routing-threats-07, April 2004.

[6] S. Bellovin, R. Bush, T. G. Griffin, and J. Rexford. Slowing routing table growth by filtering based on address allocation policies. 2001.

[7] S. M. Bellovin. A Technique for Counting NATted Hosts. In *Proc. Second Internet Measurement Workshop*, November 2002.

[8] S. M. Bellovin, J. Ioannidis, and R. Bush. Position Paper: Operational Requirements for Secured BGP. In *DHS Secure Routing Workshop*, March 2005.

[9] V. J. Bono. 7007 Explanation and Apology. NANOG email on Apr 26, 1997.

[10] P. Boothe, J. Hiebert, and R. Bush. How Prevalent is Prefix Hijacking on the Internet. NANOG36 Talk, February 2006.

[11] K. Butler, T. Farley, P. McDaniel, and J. Rexfod. A Survey of BGP Security Issues and Solutions. Technical Report TD-5UGJ33, AT&T Labs - Research, 2004.

[12] Caida. NetGeo - The Internet Geographic Database.

[13] B. Christian and T. Tauber. BGP Security Requirements. IETF Draft: draft-ietf-rpsec-bgpsecrec-04, March 2006.

[14] M. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Geographic Locality of IP Prefixes. In *Internet Measurement Conference*, October 2005.

[15] Fyodor. Idle Scanning and related IPID games. `http://www.insecure.org/nmap/idlescan.html`.

[16] Fyodor. Remote OS detection via TCP/IP Stack Fingerprinting. `http://www.insecure.org/nmap/nmap-fingerprinting-article.html`, 2002.

[17] Fyodor. Nmap free security scanner. `http://www.insecure.org/nmap/`, 2006.

[18] L. Gao. On Inferring Autonomous System Relationships in the Internet. In *Proc. IEEE Global Internet Symposium*, 2000.

[19] R. Gerdes, T. Daniels, M. Mina, and S. Russell. Device Identification via Analog Signal Fingerprinting: A Matched Filter Approach. In *Proc. NDSS*, 2006.

[20] 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*, February 2003.

[21] S. Halabi and D. McPherson. *Internet Routing Architectures*. Cisco Press, Indianapolis, Indiana, second edition, 2000.

[22] T. Hardy. RFC 3258 - Distributing Authoritative Name Servers via Shared Unicast Addresses. RFC 3258, April 2002.

[23] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registation of an Autonomous System(AS). RFC 1930, 1996.

[24] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping*, 2003.

[25] X. Hu and Z. M. Mao. Accurate Real-time Identification of IP Hijacking. Technical Report CSE-TR-516-06, University of Michigan, June 2006.

[26] G. Huston. Auto-Detecting Address Hijacking? Presentation at RIPE-50, May 2005.

[27] C. Hutzler and R. da Silva. The Relationship Between Network Security and Spam. NANOG 29 Meeting, October 2003.

[28] J. W. S. III. *BGP4 Inter-Domain Routing in the Internet*. Addison-Wesley, 1999.

[29] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. RFC 1323, May 1992.

[30] D. Karrenberg. Distributing K-Root Service by Anycast Routing of 193.0.14.129. RIPE 268, 2003.

[31] T. Kohno, A. Broido, and K. C. Claffy. Remote Physical Device Fingerprinting. In *Proc. the 2005 IEEE Symposium on Security and Privacy*, 2005.

[32] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Topology-Based Detection of Anomalous BGP Messages. In *Proc. Recent Advances in Intrusion Detection: 6th International Symposium, RAID*, 2003.

[33] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: a Prefix Hijack Alerting System. In *Proc. USENIX Security*, August 2006.

[34] B. A. Mah. pchar: A Tool for Measuring Internet Path Characteristics. `http://www.kitchenlab.org/www/bmah/Software/pchar/`.

[35] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz. Scalable and Accurate Identification of AS-Level Forwarding Paths. In *Proc. IEEE INFOCOM*, March 2004.

[36] J. Ng. Extensions to BGP to Support Secure Origin BGP (soBGP). IETF Draft: draft-ng-sobgp-bgp-extensions-01.txt, November 2002.

[37] V. N.Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proc. ACM SIGCOMM*, 2001.

[38] C. Partridge, T. Mendez, and W. Milliken. Host Anycasting Service. RFC 1546, 1993.

[39] S. Qiu, F. Monrose, A. Terzis, and P. McDaniel. Efficient Techniques for Detecting False Origin Advertisements in Inter-domain Routing. In *Proc. Workshop on Secure Network Protocols*, 2006.

[40] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proc. ACM SIGCOMM*, 2006.

[41] Y. Rekhter and T. Li. A Border Gateway Protocol. RFC 1771, March 1995.

[42] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. PathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Passive and Active Measurement Workshop* , La Jolla, CA, April 2003.

[43] Salvatore Sanfilippo. Hping. `http://www.hping.org/`, 2006.

[44] N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A Public Internet Measurement Facility. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems*, 2002.

[45] Stephen Kent and Charles Lynn and Karen Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE J. Selected Areas in Communications*, 2000.

[46] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz. Listen and Whisper: Security Mechanisms for BGP. In *Proc. first Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[47] T. Wan, E. Kranakis, and P. van Oorschot. Pretty Secure BGP (psBGP). In *Proc. NDSS*, 2005.

[48] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Protecting BGP Routesto Top Level DNS Servers. In *Proc. IEEE International Conference on DistributedComputing Systems (ICDCS)*, 2003.

[49] F. Yarochkin, M. Kydyraliev, and O. Arkin. Xprobe2, 2006.

[50] M. Zhao, S. Smith, and D. Nicol. Aggregated Path Authentication for Efficient BGP Security. In *Proc. CCS*, 2005.

[51] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *Proc. ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[52] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Detection of Invalid Routing Announcement in the Internet. In *Proc. DSN*, 2002.