Route monitoring for cloud security

Karan Hamirshi Balu
karan.balu@tecnico.ulisboa.pt

Instituto Superior Técnico

Advisors: Professor Miguel Pupo Correia
Professor Miguel Filipe Leitão Pardal

Abstract The Border Gateway Protocol (BGP) plays a critical role in providing connectivity to hosts across the world. Unfortunately, due its lack of reliability and security, invalid routes may be generated through mis-configurations or forged maliciously by attackers that can, this way, hijack traffic. Some detection systems for route hijacking have been deployed, yet the majority of them require non public information, high resources or can easily be avoided by attackers. Our document presents a monitoring solution that effectively and reliably detects a route hijack based solely on the data plane information, and with enough redundancy to prevent attacker countermeasures. This report discusses the current solutions and their limitations, talks about techniques with different objectives that can be used for our purpose, and proposes an evaluation method to validate our solution.

Keywords: Route hijacking, BGP attacks, Network security, Traceroute

Contents

1 Introduction

The Internet is a network composed by many interconnected networks. Administrative network domains are called *Autonomous Systems (AS)*, and the routing between these autonomous systems is handled by the *Border Gateway Protocol (BGPv4)* [35].

An *Internet Protocol (IP) prefix*, which is an identifier for a sub network, is associated to an AS. If some AS wants to provide connectivity between its IP prefixes and other ASes it will announce those prefixes to those ASes. Each AS contains one or more routers configured with BGP, known as BGP speakers and represented in Figure 1. Each speaker contains forwarding tables that provide the information necessary to forward a packet based on the destination and the prefix available in the table. BGP speakers send UPDATE messages to other BGP speakers in order to announce or withdraw routes.

A considerable limitation of BGP is its lack of reliability which sometimes leads to serious irregularities and outages. An example of one such outage occurred June 12th, 2015, when Telekom Malaysia started to announce, accidentally, about 176,000 prefixes to Level3, a multinational Internet service provider that operates a Tier-1 network, whom in turn accepted these and propagated them to their peers and customers. Telekom Malaysia got overwhelmed by the amount of traffic and hit its capacity limit, ultimately leading to a severe packet loss which resulted in a significant Internet slowdown [33].

An incident, that demonstrates the insecurity associated to BGP, happened on August 2013 where the Hacking Team company helped the Italian police regain control over computers that were being monitored by them. The Hacking Team worked with an Italian Web host, called Aruba, announcing to the global routing system 256 IP addresses that it did not own. This was the first known case of an ISP performing a *route hijacking attack* [10].

These security problems mainly come from the potential to interfere with route announcements in order to corrupt BGP routing policy.

Attackers can exploit this security limitation in order to claim ownership of victim prefixes and announce them to their upstream providers. Providers that do not verify the origin of the announcements may end up injecting these to the global routing system, which leads network packets to reach an incorrect destination.

This work does not intend to substitute the use of best practices to configure BGP, or several prevention mechanisms that have already been proposed [5,16], but to present a *route hijacking detection system* using a set of monitoring techniques like, traceroute, measuring latencies and IP traceback mechanisms that can effectively and reliably monitor the routes that packets are taking, ultimately

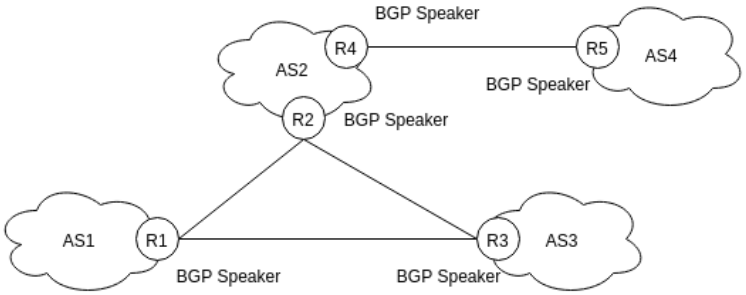

Figure 1: BGP network example

leading to a conclusion about the existence of a route hijack.

The rest of the document is organized as follow. Section 2 describes the goals associated to this work. Section 3 explains relevant works in the context of our work. Section 4 presents a proposal for a route hijacking detection system. Section 5 provides a methodology of how the results obtained will be evaluated, and finally Section 6 presents the conclusions.

2 Goals

This work has the main objective to present a route hijacking detection system that efficiently monitors the route that packets take, from source to destination. The techniques employed by the solution shall be, by themselves, redundant enough to avoid countermeasures that attackers might execute in order to bypass the detection system. Therefore, the solution has to meet the following requirements:

- *Accurate* – The solution should detect hijacking with low false positive and negative ratios;
- *Efficient* – The overhead associated to the measurements has to be kept as low as possible;
- *Scalable* – The detection system has to scale well i.e., the number of routes being monitored shall not comprimize the accuracy of the system;
- *Easy to deploy* – The system shall not depend on vantage points and privileged information like BGP messages that limit its ability to be used in practice;
- *Resilient* – The techniques that constitute the system have to be redundant enough to prevent attackers from bypassing the detection system;
- *User notification* – The system shall correctly identify the network where the hijacking process takes place, notifying the user.

This work intends to produce the following results: design and implement a system that detects a route hijacking attack, thoroughly evaluate the prototype developed, and finally report what was done in an article and in a dissertation.

3 Related Work

In this section we will give some contextual information about concepts and techniques that are relevant to our work. In Section 3.1 we will begin by providing a more detailed description on what route hijacking is and how BGP can be exploited to conduct these kind of attacks. From Section 3.2 to 3.6 we will describe relevant notions, algorithms and systems for detecting route hijacking using control/data plane information, by applying traceroute, by estimating latencies, with IP traceback mechanisms, and by leveraging avoidance routing. Finally, Section 3.7 provides a summary of all the works discussed.

3.1 Route hijacking

The Border Gateway Protocol does not ensure that BGP routers use the AS number it has been allocated, or that the AS holds the prefixes it originates. So a router can be configured to advertise a prefix from an address space belonging to another AS in an action known as IP prefix hijacking [5]. This action can happen in the following forms:

- Hijack the entire prefix - Where the hijacker announces the exact prefix of the victim. Meaning that a same prefix have two different origins.
- Hijack only a sub-prefix - The offender announces a more specific prefix from an already announced prefix. For example if the victim is announcing 200.200.0.0/16 the attacker announces 200.200.200.0/24. Due to the longest prefix matching rule, ASes that receive these announcements may direct traffic toward the wrong AS. This type of hijacking is associated to blackhole attacks, where the malicious AS drops all the packets received.
- Interception hijack - The attacker announces a fake route to an AS, that forwards traffic of the victim to the original server. The contents of the intercepted traffic can be analyzed/changed, before sending it to the legitimate destination. Figure 2 illustrates this type of attack.

Schlamp et al. described an attack where an offender claims ownership of an entire AS in [30]. To perform an AS hijacking attack, the attacker pretends that he owns the AS of the victim. These types of attacks are harder to detect because unlike the prefix hijacking attack, there are no signs of duplicate origin announcements, the only change that does occur is the formation of a new link to the upstream provider from the victim AS. According to the authors, to perform this attack, the offender needs to have a router configured with BGP and prove the ownership of the victim AS, to an upstream provider, by controlling Internet Registrars (RIR) databases where the information about ownerships are stored. The authors' conclude the paper by suggesting an early detection system that combines multiple data sources and verifies the expiration date of the domain of the autonomous systems. A warning is sent to ASes in which an expiry date is close. If a domain expires, an attacker can re-register that domain claiming the ownership.

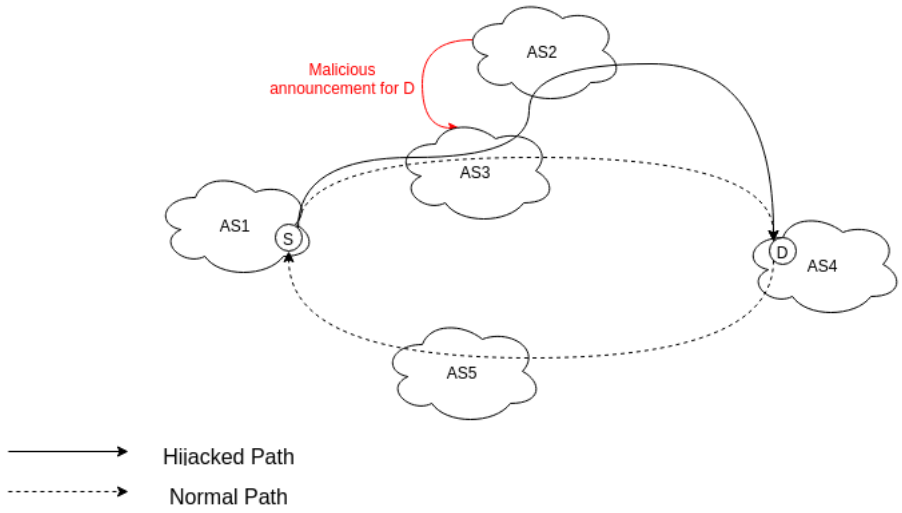

Figure 2: Example of an interception attack

Autonomous systems are bound by business relationships, therefore network operators specify routing policies which affect which BGP routes are chosen. BGP UPDATE messages contain route attributes, that are used by BGP routers to compare the announcements received. Some of the most important route attributes are the local preference, the AS path length and the origin type. A BGP router selects a route with a maximum value of local preference and a minimum value for the AS path length.

A survey [9] was conducted in order to obtain some information about the routing policies in place, in which almost 100 responses from network operators were obtained. The questions asked, involved mainly the usability of models of routing policies, like the Gao and Rexford model, and the criteria of BGP decision process (steps which help decide the route to choose). In the Gao and Rexford model ASes that buy transit services, to obtain access to other parts of the Internet, are called customers, ASes that provide these services are named as providers, and finally ASes at the same level are known as peers. The model assumes the following conditions:

- By having a choice the ASes always choose to route traffic to neighbouring customers instead of a neighbouring peer, or provider. This preference is due to the monetary gain obtained by choosing customer routes and the way to explicitly define it is by using a criteria known as local preference.
- ASes only export providers or peers routes to neighbouring customers. This implies that an AS only exports traffic if it was paid to do so.

According to the responses, 68 per cent applied both conditions and 19 per cent only applied the first condition. Reasons registered for the non usability

of the export condition include secret agreements and the thought that exports restraining techniques may end up breaking routing. These evidences show that it is difficult to predict the paths that packets take due to the heterogeneity of routing policies in different ASes.

Autonomous systems can modify forwarding attributes for their own convenience. For example, they can reduce an AS path in order to look more attractive, or put additional AS hops at the end to make a hijacked route look like it was originated by the proper AS, or simply adding the victim AS in the AS path, and once the advertisement reaches the victim AS the BGP looping system will drop the misleading announcement [5].

BGP security today, mainly consists of filtering suspicious BGP announcements. Announcements that contain loopback addresses, or addresses that are not owned by the AS that announced it. The problem of this approach is that detecting invalid route announcements is more challenging when the offending AS is several hops away. Therefore, having a global view of correct routing information would make it much easier to detect invalid routes.

An accurate routing registry would have prefix ownership, AS-level connectivity and routing policies enabled in each AS, helping ASes in verifying the legitimacy of the advertisements that they receive. The drawbacks of this model mainly include, the lack of desire of ISPs to share their proprietary routing policies, moreover the registry itself is often untrusted due to its power to manipulate the route information at will.

Ultimately, the factors that complicate the adoption of security solutions is the sharing of valuable information like the correct mapping of IP addresses and ASes in public Internet registries turning these more reliable.

3.2 Detection mechanisms

Several approaches have been proposed to detect route hijacking. These approaches can be divided in two categories: systems that rely on the information provided by the control plane, namely BGP feeds and update messages, and systems that only use information from the data plane, like the actual path that packets take. In this section we present detection systems from the two categories.

3.2.1 Control plane based An IP prefix should only be generated by a single AS. A Multiple Origin Autonomous System (MOAS) conflict occurs when a prefix is simultaneously originated by more than one AS. These conflicts may indicate a prefix hijacking [38]. The following systems use this fact to build prefix hijacking detection systems.

Kruegal et al. [20], construct a topology model, which contains a mapping between IP addresses and their origin ASes. The authoritative AS for a IP prefix is directly extracted from the BGP UPDATE messages. Any occurrence of a MOAS conflict is signaled. The drawback of this method is the need to update the

topology model every time a change of IP address ownership occurs in the network.

Prefix Hijacking Alert System (PHAS) [21] uses public repositories like Route Views and European IP Networks (RIPE) to examine BGP routing data. When a new origin AS is associated to a certain IP prefix, the owner of that prefix is notified. To join the system, the owner of the prefix has only to register in the PHAS server. This server is a single point of failure and also the system does not have protection against false registrations.

Pretty Good BGP [16] maintains, in routers, a certain amount of historical routing data to determine what routes to prefix should be considered normal. Routes with dubious origins are avoided unless there are no suitable alternative routes. This increases the overhead in routers and sufficiently equipped adversaries can force the system to accept hijacked routes.

Hu and Mao [12], provided a mechanism to detect prefix hijacking in real time based on fingerprinting techniques. The idea is to build a fingerprint to a particular network prefix based on the operating systems of machines with a given prefix, the identifier field of the IP packet and TCP and ICMP timestamps. As soon as there is a MOAS, probes are sent to all origins for the generation of fingerprints. If the fingerprints differ then it will mean that the announcements came from different hosts. This approach mainly relies on the capability to capture BGP updates, if the updates are delayed the detection will be compromised.

Argos [32] is a system that detects route hijacking, when the traffic is blackholed. The main idea is that if a prefix is hijacked then that prefix will be unreachable from different areas of the Internet, so the system will start by capturing MOAS conflicts and then correlate BGP path information, from public traceroute servers, and reachability via ping. The result of this correlation will indicate if there is a prefix hijack. This system cannot detect interception attacks.

3.2.2 Data plane based The systems, described in this section, rely only on the data plane so they have the advantage of being implementable by virtually anyone, whereas control plane mechanisms require information that is not typically available (e.g., BGP data). Furthermore since these systems are not constrained by the availability of BGP information, they are capable of showing a bigger accuracy than the systems presented in Section 3.2.1.

Zheng et al. [39], uses a set of monitors to detect prefix hijacking attacks in real time. These vantage points monitor a certain prefix from topologically diverse areas, which leads to an increase of accuracy and resilience to counter measures performed by attackers. The monitorization performed is based on two key observations. The first one is that the hop count from a certain source to a certain destination is generally constant and the second one is that the path from a source to a reference point, which is a router that is close to the target

prefix but does not belong to the same network, is almost always a sub-path of the path between a source and the target prefix. This way, each monitor keeps track of the network location of the target prefix by measuring the hop count, and if past measurements greatly differ from new measurements then this is the first indication of a prefix hijacking attack, and the system proceeds to the path disagreement process. All monitors have their reference points, keeping the paths that packets take from the vantage point to the references points and the path from the vantage point and the target prefix. A second indication of a prefix hijacking attack is signaled if the path from the monitor to the reference point stops being a sub-path of the path from the vantage point and the target prefix. The use of vantage points limits the scalability and therefore is a weakness of the system.

Zhang et al. [37], developed the first technique for detecting prefix hijacks, without the need of an infrastructure and purely data plane based. This approach is owner-centric, which means that each network must deploy the system in order to detect a prefix hijacking attack of their own prefixes. The key observation behind this system is that, in an ongoing prefix hijacking attack, replies from probes sent by some victim network to various networks will be routed to the attacker instead of the victim network, which leads to unreachablity events to the victim. To successfully detect these cases, the authors' proposed using cyclic probing of transit ASes, in which the IPs and the corresponding ASes are stored in the database and if multiple cases of sudden unreachability to different ASes are verified than the prefix hijacking alarm is activated. The drawback of this approach is that the attacker only has to forward the replies, of the probes, to the victim in order for the attack to stay undetected.

3.2.3 Common disadvantages In this section we present two works that expose forms of prefix hijacking that are not completely covered by the systems presented earlier.

To successfully deal with sub-MOAS, which is same as MOAS but in consequence of a sub-prefix hijack, Schlamp et al. proposed a validation scheme to classify these events in [31]. The scheme can be divided in four steps, in which the first one corresponds to extracting all sub-MOAS events from the BGP routing tables and update messages. From the second step to the fourth, the objective is to remove all the legitimate sub-MOAS events by, verifying the ownerships of the prefixes, by infering the business relationships between the parties involved in the event. The idea is that an attacker would not want to hijack his own upstream provider so all sub-MOAS events in which the victim is the attacker upstream are marked as legitimate. Finally the last filter uses the SSL/TLS scan, where sub-MOAS are considered legitimate if public keys, of affected hosts, remain the same before and during the occurrence of a sub-MOAS event. The authors' data-sources were able to cover 60 % of the sub-MOAS events and by applying

the filters, 46 % were legitimized.

Vervier et al. discussed a technique of executing BGP hijacks using IP addresses which were never announced before [34]. This paper alerts about the existence of an automated infrastructure capable of finding allocated but unannounced IP address space. The IP addresses found are then claimed by the attacker for a brief period of time, and used for spam campaigns in a technique known as BGP spectrum agility. In this technique, portions of IP space are announced briefly, spam is sent and then the spammer removes the routes associated with the addresses that were announced.

These short lived hijacks are an effective way for spammers to circumvent current spamming defenses, like blacklists, by hopping from one hijacked IP prefix to another.

The authors' tested current BGP hijack system, namely Argos, in order to verify its effectiveness on the uncovered hijack incidents. They found out that this system is blind to hijacks of registered though unannounced IP address space. The reason is that most BGP hijack detection systems work by building a model of the Internet AS-level topology and then using it to validate any routing change. Since there is no state for the IP address blocks, any new route announcement is accepted as legitimate.

3.3 Traceroute

Traceroute is an utility that helps network operators in analyzing the latency and the path that packets are taking until they reach their destination. This mechanism can be useful for a route hijacking detection system.

Traditional traceroute, presented by Jacobson in 1988 [13], sends multiple Internet Control Message Protocol (ICMP) probes (ICMP Echo Requests), from a specific host machine, with incremental Time-To-Live (TTL) value. Routers that receive these probes, decrement the TTL by one; when the TTL reaches 0, the router typically replies with a ICMP TTL exceeded error. This way it is possible to know the IP address of each router and by subtracting the time that the error message arrived from the time the probe was sent. It is also possible to know the Round Trip Time (RTT) associated to each router. Figure 3 illustrates this technique.

Since ICMP probes have the disadvantage of being blocked by many firewalls [23], modern versions of traceroute implementations use UDP (User Datagram Protocol) or TCP (Transmission Control Protocol) probes. TCP probes have the advantage of not being easily blocked by firewalls, because of the difficulty in differentiating TCP SYN probes to port 80 from normal web requests but has the disadvantage of requiring root privileges in order to be sent. UDP probes are more easily blocked by firewalls but do not need root privileges [23,15].

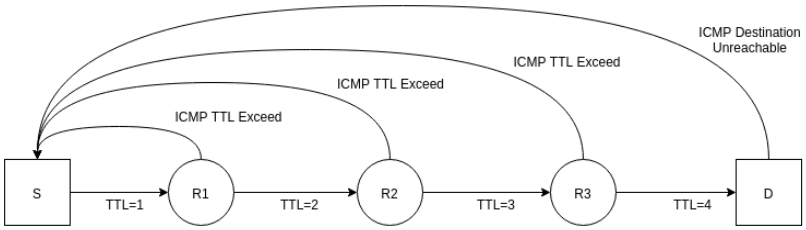

Figure 3: Example of traceroute in action

3.3.1 Anomalies Regular traceroute fails when there are devices, which perform load balancing based on packet headers [15]. Routers performing load balancing can propagate their traffic per-flow, per-packet or per-destination policy. By transmitting per-flow, the router sends packets of the same flow to the same interface. A flow is characterized by the following header fields of the IP packet: source IP address, destination address, protocol, source port, destination port, IP Type of Service (TOS) and checksum fields. So basically, different flows means different routes and the problem with this is that UDP and ICMP probes from classical traceroute have their header fields used in load balancing varied for the purpose of matching the corresponding responses from routers to the probes sent.

The resulting anomalies that are caused by load-balancing are the following:

- *Loops* - Same node appears at least two times. Probes that are sent to different paths get to a same node originating multiple responses.
- *Cycles* - The signature of this anomaly is the appearance of at least two times the same address, say k, separated by at least one address different from k.
- *Diamonds* - This anomaly occurs when multiple probes are deployed to one hop, and, due to load-balancing, traceroute ends up displaying false links.

Aside from load balancing, that according to [8] is the main cause for anomalies in traceroute results, Multi Protocol Label Switching (MPLS) and path asymmetry also cause anomalies.

MPLS is a mechanism in which routers forward packets based on labels instead of network addresses. Traceroute can still work in routers supporting MPLS, because TTL field can be copied from the original IP header to the MPLS header [7]. Yet there are some routers that do not do this and because of this, these routers are not found by traceroute [15].

Finally due to path asymmetry, RTT values can suddenly become very high giving the idea of a congested hops but in this situation it just means that the sending path is different from the return path [15].

3.3.2 Solutions / implementations In this section we present some solutions that minimize the impact caused by the anomalies. We will begin by describing a new tool that deals with load balancers, continue with an algorithm to calculate the reverse traceroute that helps diagnose RTT anomalies and finally we will provide some extensions that can be used with traceroute that gives us the AS number of a router reliably and prevent denial of service of traceroute traffic.

Paris traceroute [3] evades the problems posed by load balancers by keeping the header fields used by load balancers constant, ultimately leading probes to a same route even in a presence of a per-flow load balancer. But it still needs to match responses packets, which is done by varying header fields that are not used by load balancers. For UDP probes, it uses the checksum field. For ICMP probes, Paris traceroute varies the Sequence Number and the identifier field.

According to evaluations made by Augustin et al. [3], anomalies like loops, cycles and diamonds are significantly mitigated.

The reverse traceroute [18] is a distributed system supported by vantage points (servers) which gather information to build the reverse path. The gathered information is useful for the interpretation of RTT anomalies provided by the execution of traceroute. The steps for the reverse traceroute are the following.

1. A number of vantage points perform traceroute to a source S, creating an atlas of paths;
2. Source S executes a RR (Route Record) ping to a destination D. This ping has the route record option enabled, therefore 9 hops are recorded;
3. Considering that the destination D is within 8 RR hops, at least one hop from the returned path has been registered. The rest of the returned path can be built in an iterative manner;

If in step 3, S is not within 8 RR hops then there is a need to determine a vantage point that satisfies the condition mentioned. Only users with significant amount of resources, may use this approach to calculate the reverse traceroute since it requires vantage points.

One extension to traceroute that provides useful information, for example to delimit network boundaries, is the AS-number lookup. This can be done by accessing directly databases like RIPE or Route-Views, for IP-to-AS mapping. Yet according to [24], these databases have incomplete and out of date information so the authors' purpose to improve the mapping between IP and AS by comparing BGP information and traceroute paths from multiple vantage points. Then the traceroute tool periodically downloads the latest IP-to-AS mapping and uses it to show the AS path associated with each traceroute probe the user launches.

These techniques mainly depend on the ability to collect information.

The final extension presented here shows a way to prevent malicious nodes from treating traceroute and normal traffic differently. This method is described in [25] and works like classical traceroute except for the following characteristics:

– Hosts responding to the secure traceroute packets, provide the address of the next hop for the packet. This way the element that initiated the operation always knows the expected next hop.

– Before sending the traceroute packets, the tracer node establishes a secure channel for the next expected hop with the purpose of mentioning the signature of the traceroute packets. The signature could include the origin and the destination address of the packets plus a constrain of a value of a certain field of the packet.

– Nodes receiving these secure traceroute packets respond to the tracer node with a agreed upon marker and a secure message authentication code.

This method assumes the existence of public key infrastructure (PKI) that enables a secure key exchange between an investigating and a investigated node. Since this approach is more expensive than normal traceroute, it must be limited when possible by initiating the method near the destination.

3.4 Measurement of latency

By continuously measuring latencies between two end hosts it is possible to determine if packets timings are normal or not, which may indicate a traffic hijacking attack.

One popular way to measure latency is by using the ping tool, which leverage ICMP [29] to calculate the difference of the time in receiving echo reply packets and the time of sending echo request packets. Yet, according to Pelsser et al. [27], latencies returned by ping are highly varied due to load balancing performed by routers which leads packets to different flows. To cope with this problem, the authors' came up with a variant of Paris traceroute like tool, named Tokyo-ping, which can estimate consistent delays even in the presence of load balancing. The trick is to keep the header fields constant like Paris traceroute, but unlike Paris traceroute the flow-id is kept constant in the return path, which guarantees same return path to all measurement packets.

Although the evaluation performed indicated that classic ping displays a considerable more amount of jitter than the tool developed. The inability of executing the tool in Linux systems is a significant drawback.

A simpler approach was used in 1993 when Bolot et al. [4] presented a study to determine delays and loss behavior of packets from end-to-end and in different time scales by changing the interval between probe packets. The tool used for the measurement is called NetDyn and the idea behind the tool is to send regular UDP packets at a destination through an intermediate node. A packet includes three timestamps fields to be filled by the source, intermediate node and the destination. This tool does not deal with time synchronizations of clocks,

therefore in all the experiments the source and the destination are the same machine. The source registers a timestamp *ts* and a sequence number in the packet before sending it to the intermediate node. The intermediate node writes its own sequence number and a timestamp and sends back the packet. Finally the destination marks its timestamp *td* and calculates the RTT by subtracting *td* with *ts*. Results showed that probe packets are lost randomly, except when the Internet traffic intensity is very high. Although the results do not mean much due to the year that the study was published, the technique to conduct the study is still relevant.

The next system, unlike the technique already described, measures latency without the active cooperation of end-users. King [11] is a latency measuring tool that does not require deploying an infrastructure, because it uses Domain Name System (DNS) servers to calculate an approximation of the latency of communication between two end-hosts. The main idea behind King is supported by two observations, the first one is that end hosts are close to their DNS name servers and the second one is that recursive DNS queries can be used to calculate latency between two name servers. So basically if a client c wants to know the estimated latency between c and some target end-host t, it must first send a recursive DNS query to an authoritative name server asking for the resolution of the name t, then this authoritative name server will ask the authoritative name server of t. This server will send a response to c, thus having the latency from c to the authoritative server of t. Next by calculating the latency between c and his authoritative name server and subtracting these two latencies found, c obtains the measured latency between c and t.

All approaches presented so far are capable of providing the RTT but not the *One Way Delay (OWD)*. The OWD represents the amount of time that traffic takes from source to destination. Calculating the OWD can be hard there needs to be a strict time synchronization and access to both end - hosts. So many applications that do need these times, estimate them as being RTT / 2. To investigate this assumption, Pathak et al. [26] performed experiments using owping, a one way measuring tool that works by marking the time to a packet before sending it, then at the destination the time in the packet is subtracted from its current time. These experiments were performed on 180 research and education networks (GREN) and 25 commercial nodes on Planet Lab [1]. The traces collected consist of traceroutes, obtained with Paris traceroute, and OWD measurements. The number of Internet paths that were continuously monitored were 10000. The metrics used for the measurements are AS-level path asymmetry and router-level path asymmetry. The results found indicate that, in commercial networks delay asymmetry is very noticeable, that asymmetry in the router level happens more times than asymmetry in the AS-level, and that router-level asymmetry does not imply delay asymmetry where as delay asymmetry implies router-level asymmetry.

Another important question that Pathak et al. [26] tried to answer is if delay asymmetry is constant across time for two end-hosts. For this the fluctuations of forward and reverse delays were logged, and the conclusions are that delay asymmetry changes when delay changes.

3.5 Attack source identification

In this section, we will begin by discussing the aspects of the Internet that makes *IP traceability* difficult and continue with solutions that address these aspects with and without routers support.

According to Peng et al. [28], the Internet was made for scalability and not for security, which led all the complexity to end-hosts while leaving the core networks simple. So routers do not know the complete paths that packets take and are not capable of performing authentication. This lack of functionality gives rise to a technique known as IP spoofing, where the IP address source is forged. This technique is widely used to perform *distributed denial of services (DDoS)* attacks. The following solutions try to provide IP traceability even in the presence of IP spoofing.

3.5.1 Router-based approaches The solution presented in [6] probabilistically mark packets by inserting edge information that will enable the victim to reconstruct the route to the origin of the packets. Each packet header has two fields reserved in the IP ID field. One is for the start and end of an edge and the other one is for the distance field that represents the number of routers that the packet has traversed since it was last marked.

The procedure for marking is the following: first by a random probability a router decides to mark a packet, by writing the hash of its own IP address in the edge field and setting the distance field to 0. If the distance field was already 0, then that means that the previous router already marked the packet and thus what the router decides to do is to XOR the hash of its IP address with the hash already present in the edge field, overwriting the value. Routers that do not mark the packet, increment the distance field.

The reconstruction procedure made by the victim supposes there is an upstream router map available. The idea is to divide all the edge fields based on the distance field. At distance equal to 0, the victim will perform the hash function to all addresses of routers one hop away, present in the map, and compare them with the edge fields. If there is a match then these addresses are going to be included in the set of the reconstructed path. Using the addresses found, the victim decodes the previous routers hop-by-hop. This approach has serious disadvantages like the difficulty in deployment and the need of a lot of packets to successfully reconstruct the path.

Pi [36] is a defense mechanism that like [6] relies on network support, but unlike [6] does not need to reconstruct the path taken by packets from the attacker

to the victim, which is hard due to the limitation of space of the IP header. The key idea is to build a fingerprint in the IP ID field, which is located in the IP header. The fingerprint will characterize a certain path. If a fingerprint matches an attacker identifier then all the subsequent packets, with this identifier, will be dropped, ultimately leading the victim into a proactive role. The algorithm for marking involves a router selecting n bits from the hash of its IP address and the previous hop IP address. The consideration of both these fields help in avoiding collisions. The n bits are written in the IP ID field, in a position which is calculated through TTL mod [16/n]. One limitation of this marking scheme is that the IP ID field is of limited size therefore markings done by routers further away may be overwritten by routers closest to the victim. The authors' of the paper try to minimize this by preventing any markings from routers who are in the same AS of the victim.

3.5.2 Victim-based approaches Solutions that are router-based fall short in comparison with victim-based approaches in terms of deployment since a potential victim has much more incentive in deploying security measures than a network service provider.

One example of these victim-based approaches is *Hop-Count Filtering (HCF)* [14]. This solution is supported by the notion that an IP addresses can be spoofed but the number hops made by packets cannot. The idea is to have a mapping table IP-to-hop-count and depending on the the information registered receiving traffic is dropped or not.

Since the hop-count is not directly in any field of an IP packet, it has to be calculated through the TTL field. This value is decremented hop by hop from the source to the destination, therefore the hop count will be the initial TTL minus the final TTL. The problem with this is that the initial TTL varies from machine to machine but accordingly to a study made by the Swiss education and research network [2], modern operating systems (OSs) use a small set of initial values therefore this value can be inferred.

To be able to build an efficient IP-to-hop-count table, the following objectives must be achieved,

- accurate IP to Hop count mapping
- up to date IP to Hop count mapping
- moderate storage requirement

In order to reduce the space requirements, the authors' executed a technique called IP Address Aggregation where hosts are grouped according to the first 24 bits and the groups formed are divided even more based on hop-counts.

For an attacker to successfully evade such a table, it is necessary to set an initial TTL value T' for each packet such that the difference between T' with

the number of hops *hz* from the flooding source to the victim, is equal to the difference between the initial TTL value T and the hop count *hs* from the spoofed IP address and the victim. The attacker can easily calculate *hz*, by performing traceroute, and infer T. Yet the calculation of hs from a randomly selected IP address requires the attacker to build an a priori hop count table, which is much more difficult than building the table from the victim since the attacker does not have access of the final TTLs of normal traffic. The need for a significant amount of updates to keep the system up-to-date is a drawback of the system.

3.6 Avoidance routing

Avoidance routing is a technique to steer traffic around a specific zone. Systems that respond to network failures by performing avoidance routing, need first to locate the area to avoid. These detection mechanisms may prove to be relevant for our work. Furthermore, by evading the ASes where it is known that the hijacking attack takes place it is possible to prevent these attacks. The following works present mechanisms for performing avoidance routing, whereas the last describes a way to prove that traffic did not traverse a certain forbidden region.

In the method explained by Kline and Reiher [19], users only need to set security properties that they wish to avoid in their avoidance request. Security properties are information regarding the geopolitical location, ownership and router type. Routers are configured to be aware of their own security properties. A router, by receiving an avoidance request, starts looking for a route to the destination that satisfies the security properties defined in the request. If a route is found then the request is forwarded to that route, if there is no path with the correct conditions then the router will start a depth first search sending request messages to all interfaces, by turn, which route to the destination. It will stop if it receives a success message, meaning that a path was successfully found, or if it receives failure messages from all of them which means that no route was found. The overhead associated with the verification of the security properties is non negligible.

LIFEGUARD [17] is a system which has the purpose of increasing Internet availability to networks where hosts reside, also known as edge networks. It uses a set of techniques that allow the location of failures, even in the presence of asymmetric routing, and the rerouting around outages. The rerouting is made by using the BGP loop-prevention mechanisms in which an edge network O, puts the number of the AS A into path advertisements marking A as already visited and consequently leading to the rejection of the announcement upon arrival to A, and the withdrawal of the path from its neighbors, forcing all these ASes to find routes to O that do not involve A. The failure detection system employed by LIFEGUARD involves four steps.

– The first one is to keep information on the round-trip times between the source and the destinations to distinguish what times are normal and what times are not, eventually generating candidates to failure locations.

– In the second step, LIFEGUARD isolates direction of failure using spoofed pings from vantage points. For example if these probes reach the vantage points and not the source, it means there is a problem in the reverse path.
– For the third step, the intention is to narrow down the point of failure in the not working direction. If the problem is in the forward direction, then LIFEGUARD uses traceroute to measure the portions of the working path. If it is in the reverse path, vantage points perform pings to all nodes in the forwarding path, and then execute reverse traceroute to all pingable nodes.
– Finally, in the fourth step, hops that were pingable from the failure direction are removed from the candidate set. The first unreachable hop does not have a functional path to the source and therefore rerouting around it may return connectivity to the source.

The next and final system of this section, guarantees that network traffic passed or not in some zones in the network.

The key idea behind Levin et al. [22] is to prove that traffic traversing a certain relay node, known in this context as alibi, on a path from a source to a destination and traffic passing through a forbidden zone are mutually exclusive. This is due to the fact that the latency for the traffic to reach a destination traversing any node in the forbidden zone and the alibi node is much bigger than going via alibi node alone. Thus an alibi must be a node that is sufficiently far from the forbidden zone, in order for the mutually exclusive event to be verifiable. When an alibi receives a packet, it signs it and sends the signature to the source meaning a proof of avoidance. All the source needs to do is to send a query to a peer in the overlay network, specifying a region to avoid, a destination and the identification of a peer that probably is not in the forbidden region. A peer q will forward the query to a peer n if the latency between q and n is bigger than the latency between q and the closest node in the forbidden region. There is no mention in the paper about how trusted peers are found and for this reason it implies that all participating nodes are trusted.

3.7 Summary

The previous sections describe various techniques that allow route monitoring. Section 3.3 demonstrates how different traceroute tools work using ICMP, UDP and TCP. It also showed problems associated to classical traceroute, which involved the load balancers and asymmetry of the network, and solutions that minimize those problems using Paris traceroute and reverse traceroute. Therefore, the combination of Paris traceroute and reverse traceroute enables more reliable results. In Section 3.4 different measuring tools were discussed. Table 1 presents a comparison between the tools. Among these tools, owping is apparently the best because it separates the forward latency from the reverse latency providing more useful information to applications like streaming. Tokyo-ping is also interesting due to the adaptation of Paris traceroute it provides more reliable information.

Table 1: Comparison of different latency estimation tools

Tool	Approach	Measurement	Advantage	Disadvantage
Ping	Host based	RTT	Integrated in every operating systems	Latencies highly varied
Tokyo-Ping	Host based	RTT	Consistent latencies measurements	Not operational in linux systems
NetDyn	Host based	RTT	Low Overhead	Clock synchronization
King	DNS based	RTT	No infrastructure needed	Hosts need to be close to DNS servers
Owping	Host based	OWD	Separation of forward and reverse delay	Clock synchronization

Section 3.5 illustrated ways to perform IP traceback even if attackers perform IP spoofing. Three approaches were presented and a comparison between them is shown in Table 2. HCF has a significant advantage in relation to the other two approaches because, since it is victim based, it is much more easy to deploy.

Table 2: Comparison of different IP traceback methods

Solution	Approach	Method	Advantage	Disadvantage
IP marking scheme	Router-based	Path reconstruction	Low overhead to routers	Need a lot of packets to reconstruct path
PI	Router-based	Path identification	Does not need full path	Path differentiable markings may be overwritten
HCF	Victim-based	Hop count	Easy to deploy	Constant need of updates

Finally in Section 3.6, systems that increase availability by performing avoidance routing were presented. In these systems LIFEGUARD [17] stands out due to its outage detection system which can be successfully performed even in the presence of asymmetric routing.

4 Architecture

In this section, we will propose a system that detects BGP route hijacking based on the works presented in the previous sections. We will begin by presenting the intuition behind the solution with an example of an ongoing route hijacking, and conclude by specifying the components and the tools necessary for the proposed solution. One of our requisite is not to change router configurations, therefore the proposed system will be implemented in the application layer.

4.1 System behavior and intuition

Three important metrics for our proposed system are the latency, the hop count, and the path between a source and a destination. The time that packets take from a source to a destination is hard for an attacker to evade. However due to the congestion in the network, caused by legitimate reasons, it is also difficult to detect a route hijacking. For this reason, we propose adding the number of hops which is a technique that can be avoided by attackers. A sophisticated attacker, since it has hijacked the traffic, can bypass this monitoring technique by adjusting the TTL field of the packets in order to reproduce the number of hops of a normal path between the source and the destination. Therefore our proposal will keep state of the number of hops and the time that packets take from end-to-end.

An anomaly caused by the two states, OWD and hop count, gives us enough reason to start investigating the path that packets are taking. For this, traceroute is issued but as we have seen in Section 3.3.1, routers may be configured to block certain protocols like ICMP, UDP or TCP. Thus traceroutes with different protocols are executed and the results are merged generating the most complete path possible.

One counter-measure that attackers can perform to disrupt techniques that rely on the data plane, like traceroute, is treating differently traceroute traffic and data traffic which may lead to an incorrect interpretation of the results obtained. For example, a malicious router could drop only traceroute probes while still forwarding data packets, indicating an existence of a faulty link when there is none. To thwart this we will adapt a technique explained in Padmanabhan and Simon [25] that reliably identifies a faulty link by marking traceroute packets and obtaining a confirmation of the reception of these packets from each forwarding node.

Finally hops of the authenticated path are mapped to autonomous systems. This mapping increases accuracy because we only need one router from a autonomous system to correctly obtain a path that packets are taking. The resulting path is compared with previous paths which enables to draw conclusions about a route hijacking attack.

Figure 4 shows an example of the detection system on an ongoing route hijacking attack, where the monitoring functionality is explained step-by-step. In the figure, S and D represent the machines that will run our mechanism, being clients or servers. Each of them stores a table that has the OWD, the hop count, and the path that packets take. In this particular example S possesses the OWD and the hop count of the traffic from D to S, but the path that it has is from S to D. The ASes represent different autonomous systems, in which the AS2 executes a malicious announcement to AS3 misleading the traffic from S to pass also in AS2.

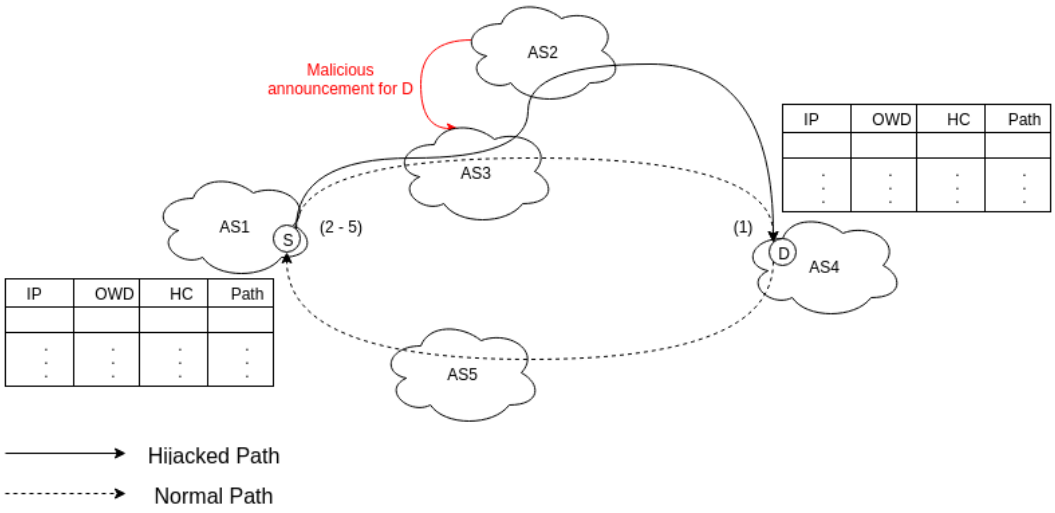

Figure 4: Example of route monitoring in the network

(1) - D reacts to the anomalies it observed in time and hop count, by sending a traceroute request to S.

(2) - S executes Paris traceroutes with different protocols (ICMP, UDP, TCP) and joins the different results, leading to a most complete path possible.

(3) - To confirm the path generated in (2), a secure traceroute procedure is conducted. Since this procedure is heavyweight, due to the sharing of secretes, the forwarding node that performs it must be as close as possible to the destination. Figure 5 exemplifies this step, where S is performing the procedure by giving a signature that all traceroute packets will have to the investigated node. This node sends a response to S if it has received the traceroute packets, with the signature.

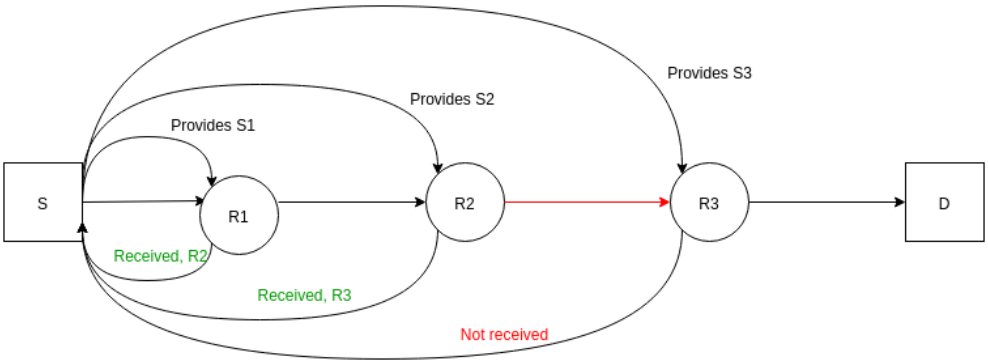

Figure 5: Example of authentication of a path

(4) - If the path is authenticated with the procedure shown in (3), the IP addresses are mapped to autonomous systems with the help of public Internet registries. Note that we only need one router from each AS to correctly obtain the path that the packets are taking.

(5) - Finally, S compares the path obtained with the path stored in its table and draws conclusions about the existence of a route hijacking attack.

4.2 System components and details

This section illustrates the relation between the components, as seen in Figure 5, and the tools that are going to be used.

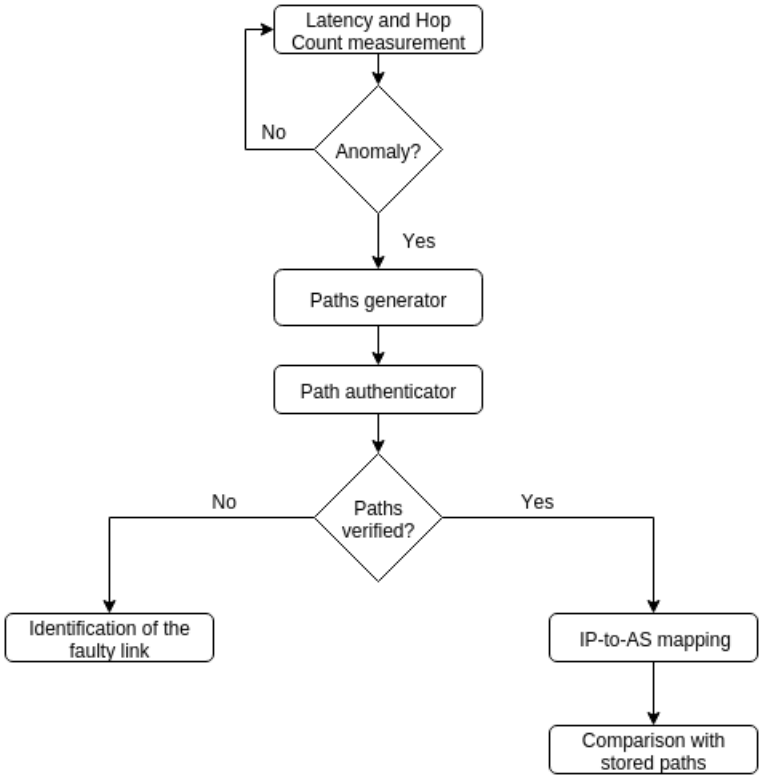

Figure 6: Fluxogram of the proposed solution

– **Latency and hop count measurement** - A component that continuously compares times, using owping [26], and the number of hops for each source and destination, using the technique in [14].

- **Paths generator** - Paris based traceroutes, as seen in [3], with different protocols (UDP, ICMP and TCP) are executed, and the results are combined to obtain the most complete path possible.
- **Path authenticator** - The path obtained by the paths generator module is authenticated using the procedure, explained in [25], that reliably confirms the full path or the faulty link. It is important to note that this method works only in open networks like overlays, which are networks built on top of networks and nodes from these networks are connected via virtual links.
- **IP-to-AS mapping** - The IPs from the authenticated path are mapped to autonomous systems with the help of Internet registries. This mapping increases the accuracy.
- **Comparison with stored paths** - The final AS path is compared with previous measurements which will enable conclusions to be drawn.

5 Evaluation

We will evaluate our proposed solution experimentally by building a prototype. The experimentation will be conducted in a platform like PlanetLab [1], which allows testing the scalability of the prototype and provides control over the nodes. We will emulate some route hijacking but also analyze carefully the data collected to detect the existence of real route hijacks, in case they happen during the experimentation time.

To evaluate the overall performance of our proposed solution, we will measure the accuracy and the latency of the detection. For accuracy we will consider two important metrics, the false positive ratio and the false negative ratio. False positive ratio indicates the percentage of false route hijacks, while false negative ratio indicates the percentage of routes hijacks that were not reported. These metrics will be calculated considering individual techniques and the union of all techniques, which will enable a better understanding of the impact caused by each technique to the detection system. For latency, we will consider how many measurements are necessary to successfully detect a route hijacking with different thresholds, these thresholds define how much must the measurements deviate in order to declare route hijacking. Our focus is not in the time that it takes to detect a route hijacking, because this time is directly correlated to the frequency that measurements are performed corresponding to a trade-off of how fast routes hijacking are discovered, and overhead associated to the measurement traffic. In detail we want to measure the following:

- False positive and negative ratios considering first, the hop count and time of packets and then considering also the paths of the packets.
- How many measurements are needed to detect a route hijacking, under different thresholds.
- Test the impact on the accuracy, of the proposed solution, by varying the amount of nodes being monitored.

6 Schedule of Future Work

Future work is scheduled as follows:

- January 12 - March 1, 2016: Implementation of the proposed solution.
- March 2 - April 15, 2016: Experimental evaluation of the results.
- April 16 - May 30, 2016: Write the dissertation and a paper about the work done.
- May 31 - June 30, 2016: Finish the writing of the dissertation.
- July 1, 2016: Deliver the MSc dissertation.

7 Conclusion

This work intends to present a route hijacking detection system that does not need to access priveleged information like BGP messages, that is redundant enough to not be avoided by attackers, and accurate enough to detect the existence of an attack and where it occurs.

We discussed route hijacking detection systems that use the control/data-plane and the forms of prefix hijacking that are not completely covered by them. We analyzed different monitoring approaches that can contribute to our objective. These works consisted in measuring latency, under the form of RTT and OWD, attack source identification, with and without network support, traceroute, using different protocols and performing it in a reliable way, and finally avoidance routing mechanisms.

The inspiration provided by these works, helped us in constructing our monitoring solution proposal selecting the appropriate approaches that met our requirements.

Acknowledgments We thank Professor Miguel Pupo Correia and Professor Miguel Filipe Leitão Pardal, for their continued help and availability during the concretization of this report. We would also want to thank the rest of the Safe Cloud team for their support.

References

1. Planetlab : global research networks that supports the development of new network services https://www.planet-lab.org/
2. The swiss education and research network default ttl values in tcp/ip, 2002. http://www.map.meteoswiss.ch/map-doc/ftp-probleme.htm
3. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding traceroute anomalies with Paris traceroute. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement pp. 153–158 (2006)
4. Bolot, J.C.: End-to-end packet delay and loss behavior in the internet. ACM SIGCOMM Computer Communication Review 23(4), 289–298 (1993)

5. Butler, K., Farley, T.R., McDaniel, P., Rexford, J.: A survey of BGP security issues and solutions. Proceedings of the IEEE 98(1), 100–122 (2010)
6. Dawn Xiaodong Song, Perrig, a.: Advanced and authenticated marking schemes for IP traceback. INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE 2, 878–886 (2001)
7. E. Rosen, A. Viswanathan, R.C.: Multiprotocol label switching architecture (January 2001)
8. Flach, T., Katz-Bassett, E., Govindan, R.: Quantifying violations of destination-based forwarding on the internet. Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12 p. 265 (2012)
9. Gill, P., Schapira, M., Goldberg, S.: A survey of interdomain routing policies. ACM SIGCOMM Computer Communication Review 44(1), 28–34 (2013)
10. Goodin, D.: Hacking Team orchestrated brazen BGP hack to hijack IPs it did not own (Jul 12) (2015), http://arstechnica.com/security/2015/07/hacking-team-orchestrated-brazen-bgp-hack-to-hijack-ips-it-didnt-own/
11. Gummadi, K.P., Saroiu, S., Gribble, S.D.: King. Proceedings of the second ACM SIGCOMM Workshop on Internet measurment - IMW '02 p. 5 (2002)
12. Hu, X., Mao, Z.M.: Accurate Real-time Identification of IP Prefix Hijacking. 2007 IEEE Symposium on Security and Privacy - SP '07 (2), 3–17 (2007)
13. Jacobson, V.: traceroute ftp://ftp.ee.lbl.gov/traceroute.tar.gz
14. Jin, C., Wang, H., Shin, K.G.: Hop-count filtering. Proceedings of the 10th ACM conference on Computer and communication security - CCS '03 p. 30 (2003)
15. Jobst, M.: Traceroute Anomalies. Innovative Internet Technologies and Mobile Communications - IITM, and Aerospace Networks - AN (August), 47–55 (2012)
16. Karlin, J., Forrest, S., Rexford, J.: Pretty Good BGP: Improving BGP by Cautiously Adopting Routes. Proceedings of the 2006 IEEE International Conference on Network Protocols pp. 290–299 (2006)
17. Katz-Bassett, E., Scott, C., Choffnes, D.R., Cunha, I., Valancius, V., Feamster, N., Madhyastha, H.V., Anderson, T., Krishnamurthy, A.: Lifeguard. Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication - SIGCOMM '12 42(4), 395 (2012)
18. Katz-bassett, E., Scott, C., Sherry, J., Wesep, P.V., Anderson, T.: Reverse Traceroute. NSDI 10, 219–234 (2010)
19. Kline, E., Reiher, P.: Securing data through avoidance routing. Proceedings of the 2009 workshop on New security paradigms workshop - NSPW '09 p. 115 (2009)
20. Kruegel, C., Mutz, D., Robertson, W., Valeur, F.: Topology-based detection of anomalous bgp messages. Recent Advances in Intrusion Detection pp. 17–35 (2003)
21. Lad, M., Massey, D., Pei, D., Wu, Y.: PHAS: A prefix hijack alert system. Usenix Security pp. 153–166 (2006)
22. Levin, D., Lee, Y., Valenta, L., Li, Z., Lai, V., Lumezanu, C., Spring, N., Bhattacharjee, B.: Alibi Routing. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication pp. 611–624 (2015)
23. Luckie, M., Hyun, Y., Huffaker, B.: Traceroute probe method and forward IP path inference. Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC '08 p. 311 (2008)
24. Mao, Z.M., Rexford, J., Wang, J., Katz, R.H.: Towards an accurate AS-level traceroute tool. Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03 p. 365 (2003)
25. Padmanabhan, V.N., Simon, D.R.: Secure traceroute to detect faulty or malicious routing. ACM SIGCOMM Computer Communication Review 33(1), 77–82 (2003)

26. Pathak, A., Pucha, H., Zhang, Y., Hu, Y.C., Mao, Z.M.: A Measurement Study of Internet Delay Asymmetry. Passive and Active Network Measurement pp. 182–191 (2008)
27. Pelsser, C., Cittadini, L., Vissicchio, S., Bush, R.: From Paris to Tokyo. Proceedings of the 2013 conference on Internet measurement conference - IMC '13 pp. 427–432 (2013)
28. Peng, T., Leckie, C., Ramamohanarao, K.: Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys 39(1), 3–es (2007)
29. Postel, J.: Internet control message protocol (September 1981)
30. Schlamp, J., Carle, G., Biersack, E.W.: A forensic case study on as hijacking. ACM SIGCOMM Computer Communication Review 43(1), 5 (2013)
31. Schlamp, J., Holz, R., Gasser, O., Korsten, A., Jacquemart, Q., Carle, G., Biersack, E.W.: Investigating the nature of routing anomalies: closing on subprefix hijacking attacks. Traffic Monitoring and Analysis pp. 173–187 (2015)
32. Shi, X., Xiang, Y., Wang, Z., Yin, X., Wu, J.: Detecting prefix hijackings in the internet with argus. Proceedings of the 2012 ACM conference on Internet measurement conference - IMC '12 p. 15 (2012)
33. Toonk, A.: Massive route leak causes Internet slowdown (May 31) (2015), http://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/
34. Vervier, P.a., Thonnard, O., Dacier, M.: Mind Your Blocks : On the Stealthiness of Malicious BGP Hijacks. Ndss '15 (February), 8–11 (2015)
35. Y. Rekhter, T. Li, S.H.: A border gateway protocol 4 (rfc4271) (January 2006)
36. Yaar, a., Perrig, a., Song, D.: Pi: a path identification mechanism to defend against DDoS attacks. Proceedings 19th International Conference on Data Engineering 03pages, 93–107 (2003)
37. Zhang, Z., Zhang, Y., Hu, Y.C., Mao, Z.M., Bush, R.: iSPY: Detecting IP Prefix Hijacking on My Own. Proceedings of the ACM SIGCOMM 2008 conference on Data communication - SIGCOMM '08 p. 327 (2008)
38. Zhao, X., Pei, D., Wang, L., Massey, D., Mankin, A., Wu, S.F., Zhang, L.: An analysis of BGP multiple origin AS (MOAS) conflicts. Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement Workshop - IMW '01 pp. 31–35 (2001)
39. Zheng, C., Ji, L., Pei, D., Wang, J., Francis, P.: A Light-weight Distributed Scheme for Detecting Ip Prefix Hijacks in Real-time. Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications pp. 277–288 (2007)