

GLOBAL INTERNET ROUTING FORENSICS: VALIDATION OF BGP PATHS USING ICMP TRACEBACK

Eunjong Kim, Dan Massey and Indrajit Ray

Department of Computer Science

Colorado State University

{kimeu, massey, indrajit}@cs.colostate.edu

Abstract Nearly all network applications rely on the global Internet routing infrastructure to compute routes and deliver packets. Unfortunately, false Internet routes can be maliciously introduced with relative ease into the routing infrastructure. This is because Border Gateway Protocol (BGP), the Internet's global routing protocol, lacks basic authentication and monitoring functionalities. If false routes are introduced, it can lead to total collapse of packet forwarding leading to denial of service or misdirected traffic. Currently, it is impossible to prevent such malicious injection of false traffic routes. We believe that an ability to identify false paths through efficient validation, proper recording and forensic analysis of routing data, will considerably help in the prosecution of the miscreant and will act as a strong deterrent. In this work we propose such a mechanism. For each path information, we use ICMP (Internet Control Message Protocol) traceback message with AS-PATH information and link connectivity information. Our path verification technique is proportional to the amount of traffic carried on a path, uses efficient off-line verification technique with which each router independently and dynamically keeps track of local database, and allows a destination to monitor its routes, detect false paths used by remote sites, and record routing data for later forensic analysis in the event of an attack. Last but not the least, our approach does not require modifications to the BGP protocol and hence can be easily deployed.

Keywords: Routing forensics, BGP, ICMP Traceback

1. INTRODUCTION

The Internet plays an increasingly important role in commerce, government, and personal communication. A large scale attack (or even an unintended operational error) can seriously disrupt service to critical services and have a major impact on the economy. In response, a variety of *end system* security techniques such as encrypted connections and VPNs have evolved. However,

almost all such systems rely on the unsecured Internet infrastructure to compute routes and deliver packets. If the Internet infrastructure fails to deliver data packets, there is very little the end systems can do to recover. In this paper, we examine techniques for detecting invalid routes in the Internet infrastructure and present an effective approach for gathering and extracting routing data from the network that can be used later on for forensic analysis.

At the global infrastructure level, the Internet consists of thousands of **Autonomous Systems (AS)**. An AS can be viewed as a group of links and routers that are under the same administrative control. Each AS is assigned a unique number. For example, Colorado State University is AS 12145 and AT&T is AS 7018. There are currently over 18,000 active Autonomous Systems on the Internet. The ASes are ultimately responsible for routing traffic through the Internet. The **Border Gateway Protocol (BGP)** [12] is the de facto inter-AS routing protocol; it is used to exchange reachability information between ASes. BGP is designed to cope with events that alter the structure of the Internet such as addition of new links and new ASes, the failure (temporary or long lasting) of links, and changes in routing policies. However, BGP presents several interesting challenges for path validation and routing forensics.

BGP contains very limited security mechanism and implicitly assume that routers advertise valid information. For example, suppose that AS 12145 (Colorado State University) incorrectly (maliciously) reports that it has direct connection to `www.largecompany.com`. Other BGP routers will believe this route and portions of the Internet will select this path as the best route to `www.largecompany.com`. When the traffic arrives at AS 12145, the traffic may simply be dropped or someone may attempt to imitate (spoof) the `www.largecompany.com` website. As a result of this false route, `www.largecompany.com` may notice a drop in traffic. If AS 12145 later withdraws its false route, BGP routers at some point will simply switch back to the valid path. However, it will take a considerably long time for the changes to propagate throughout the Internet. In addition, owing to the large number of BGP destinations and the large volume BGP routing changes, a particular BGP path change is unlikely to trigger any alarms at remote sites. Nonetheless such actions has the potential to cause significant damage to the affected site. Extracting enough routing information from the network so as to be able to identify the reason for this lost traffic (namely that it has been triggered by some AS announcing an invalid path information) is quite challenging with current techniques.

In this paper, we present an approach for monitoring, gathering and validating the route to a destination. The technique works briefly as follows. Suppose AS_1 has an incorrect path information for AS_2 . This can be owing to one of several causes like malicious advertisement of wrong path information by a neighbouring AS of AS_1 , or misconfiguration at AS_1 and so on. With our approach, AS_2 will eventually know that AS_1 has an incorrect path information

about AS_2 ¹. In addition, AS_2 has the potential to know what other ASes have invalid path information about itself. If AS_1 (and the other ASes) are reachable from AS_2 , then AS_2 can alert these ASes (maybe via some protocol which is, however, outside the scope of the current work).

Our approach is based on exploiting the ICMP (Internet Control Message Protocol) traceback message. As data packets flow through routers, occasional packets (recommended one in twenty thousand) also generate an ICMP traceback message. These traceback messages allow a destination to reconstruct the path used to reach the destination. Unlike other approaches that attempt to monitor or validate all paths, we focus on paths that are actively carrying data traffic. There are over 18,000 different Autonomous Systems that have some path to `www.largecompany.com`, but relatively few of these sites may be actively sending data traffic. By exploiting the ICMP traceback mechanism, we only send monitoring and validation messages for paths that are actively in use. We enhance the ICMP traceback message with AS-PATH information, and link connectivity information. We also send traceback messages along three (ideally disjoint) paths to reduce the probability that packets are (maliciously or otherwise) dropped or corrupted. The net result allows a router to dynamically keep track of paths used to reach the destination, monitor routing changes for the actively used paths to this destination, and provide a log that can be used to reconstruct routes in the event of a suspected attack. As a side-effect, our approach provides a more fault-tolerant, fault-resilient, reliable and secure BGP routing protocol for the entire Internet infrastructure.

The rest of the paper is organized as follows. In the next section, we briefly overview the BGP threat model. Section 3 provides an overview of current solutions proposed by other groups. Section 4 describes our approach, and in Section 5 we summarize and conclude our suggested approach. Unsolved problems and future works are discussed in Section 6.

2. BGP AND ITS VULNERABILITIES

BGP is a *path vector routing algorithm* where each router advertises its best route to a destination. The route to a destination includes the full path of Autonomous Systems used to reach the destination. As a result, a router does not learn the full Internet topology and only a best path to each destination. Given this partial topology information, it is hard to confirm the validity of a particular path and BGP routers tend to accept *any* path that is advertised. This leads to the following possible exploits:

¹Currently the same information can be obtained by a BGP administrator going over BGP log records which can be in the millions. However, no mechanism exists that will alert the BGP administrator to go over the log records.

2.1. MISCONFIGURATION

Mahajan et al studied globally visible BGP misconfigurations in [10]. They define two types of BGP misconfiguration. One is origin misconfiguration which means the accidental injection of routes into global BGP tables such as injecting one or many more-specific prefixes (deaggregation), announcing part of someone else's address space (hijacks) and propagating private network prefixes. The other is export misconfiguration which is the accidental export of routes in violation of an ISP's policy. Misconfigurations increase routing load by generating unnecessary BGP updates. The incorrect announcement can disrupt connectivity either partially or globally.

2.2. DELIBERATE ATTACKS

Two main attacks in the network is falsification and denial-of-service (DoS). A falsification is defined as a bogus BGP protocol message that differs from a message that a correctly configured router would send [18]. Attackers can falsify withdrawn routes, path attributes and network layer reachability information (NLRI) which are components of the UPDATE message. The *blackhole attack* is a general attack that relies on falsification. In a blackhole attack, a malicious AS injects wrong routing information to attract traffic that would otherwise not flow through it, thus gaining control of a path. The blackhole attack can occur not only by falsification but also by misconfiguration. This threat can be prevented by using IPsec and proper certificates. Denial-of-service (DoS) is an AS may accidentally filter out routes it otherwise announces. To remote observer, this denial-of-service is indistinguishable from a failure. DoS attack is a resource exhaustion attack. An attacker may be able to make a router perform resource-intensive operations, such as public-key certification verification or signature generations. Those operations make the router slow down.

3. RELATED WORK

Perlman examined *Byzantine behavior* within routing protocols in her dissertation [11] and was among the first to consider routing security. She defined two different type of failures such as, a *simple failure* which consists of a node or link becoming inoperative, and ceasing to function at all and *Byzantine failures* which are caused by nodes or links which continue to operate, but incorrectly. By her definition, a node with Byzantine failure may corrupt messages, forge messages, delay messages, or send conflicting messages to different nodes. In [4], Pei et al. summarize the overall effort for resilient Internet routing and proposed defenses. They review the various approaches to improving the resiliency of the Internet routing protocols. We categorize the previous works

into two classes; Cryptographic based approaches and Non cryptographic based approaches.

3.1. CRYPTOGRAPHIC APPROACHES

Correct operation of the BGP routing infrastructure depends upon the integrity, authenticity, and timeliness of the routing information that BGP distributes via UPDATES. Using cryptographic keying material provides strong authorization and authentication capabilities to BGP control traffic. Most of cryptographic based protocols make use of digital signatures and public key certification. Validation is significantly expensive because of the significant amount of data and large number of signers.

Secure BGP: Secure BGP (S-BGP) [14] is a comprehensive security solution for BGP and it addresses most of BGP's architectural security problems. Public key infrastructure (PKI) is used to provide proper protection to the BGP routing information and validation. S-BGP protects the entire BGP UPDATE message by adding (i) Public Key Infrastructure (PKI) to authorize prefix ownership and validate routes, (ii) a new attribute which ensure the authorization of routing UPDATES, and prevents route modification of intermediate BGP speakers, and using (iii) IPSec to provide routing message confidentiality. S-BGP uses certification hierarchies: an address space PKI, and an AS ownership and router PKI. It protects the AS_PATH from modification and truncation, and unauthorized advertisements of an IP prefix. S-BGP requires several digital signatures in each UPDATE, and as a result has a high CPU overhead for verifying UPDATE messages.

Secure Origin BGP (soBGP): Ng proposed secure origin BGP (soBGP) which verifies the origin of route advertisements and prevents the advertisement of unauthorized prefixes [8]. All information pertaining to security is handled by the new SECURITY message type which is used to distribute three type of certificates. The Entity Certificate binds a node or router in the network to a public key. Authorization Certificates are used to verify an AS is authorized to advertise an address block. Policy Certificates provide details on policy or performance.

Secure Path Vector (SPV): Hu et al. proposed Secure Path Vector (SPV) [18] which improves on S-BGP through the use of more efficient symmetric cryptography for securing against the truncation and modification attacks. SPV is configurable to allow tradeoffs between security and CPU usage. With S-BGP, the performance overhead to verify and sign each UPDATE is unacceptable and the space overhead to store the public key is also significant. Since not only security but also performance is very important concern for the Internet, SPV uses high speed signature algorithm and verification tasks based on cryptographic hash function in order to provide very good performance.

3.2. NON CRYPTOGRAPHIC APPROACHES

Although cryptographic approaches provide a theoretical security guarantee, problems with public key infrastructures and concerns over overhead, correct implementation, and so forth have prevented these techniques for being adopted in today's network. None of the cryptographic techniques is currently available in the Internet and there are no definitive plans to add them. Non-cryptographic based solutions may provide fewer guarantees, but can be deployed more readily in today's network.

Multiple origin AS (MOAS): Multiple origin AS (MOAS) conflicts occur when multiple ASes announce themselves as the origin of a particular prefix. It occurs for various reasons including misconfiguration, multihoming, or malicious attacks [16]. MOAS conflicts can occur for valid reasons, such as multihoming and exchange points or invalid routes. Zhao et al. proposed to use the BGP community attribute for a list of the valid originating ASes to detect false route announcement and prefix hijacking [17]. Any AS which is not in the MOAS list is regarded invalid and alarm will be triggered. Then further verification process is required.

Route Filtering: Route filtering is a common way to control BGP messages. With proper filters, bad routes will be stopped from propagating near or at its source thus reduce impact to a small scope of the Internet. Therefore Attackers has less chance to introduce routes on the fly to the whole Internet and launch attacks. It localizes the impact of the bad routes and reduces the weakness which potentially will be explored by attackers. Wang et al. proposed path-filtering approach for protecting BGP routes to the top level DNS servers [7]. Their approach can be deployed for the top-level DNS servers (or for the well-known destinations) since their approach truly depends on the high degree of redundancy and stability in network connectivity to the server locations.

Interdomain Routing Validation (IRV): Interdomain Routing Validation [5] is receiver-driven protocol which means recipients of BGP UPDATE messages to corroborate the received information. IRV is independent of the routing protocol and is used in conjunction with BGP.

Routing Path Verification: Pei et al. developed a simple and effective approach to detecting invalid routing announcement in Routing Information Protocol (RIP) [3]. RIP is a distance vector protocol which is a shortest path routing protocol. On the other hand, BGP is a path vector protocol and a policy based routing protocol. Their solution uses probing message for invalid route verification. We believe this verification technique can be deployable for the BGP. If we verify the UPDATE message in a proper manner, false paths can be detected before it propagates and poisons other nodes.

Source Tracing: Not only validating update messages and verifying their authenticity, accuracy, and consistency, but also ensuring the robustness of

packet forwarding improve the security of the routing protocol [15]. The key idea behind secure traceroute is to securely trace the path of existing traffic, rather than that of special traceroute packets, to prevent adversaries from misleading the tracer by treating traceroute and normal traffic differently. Secure traceroute responses are also authenticated, to verify their origin and prevent spoofing or tampering.

ICMP Traceback (iTrace): In the original ICMP Traceback proposal [2], ICMP Traceback (iTrace) is defined to carry information on routes that an IP packet has taken. This mechanism is used to deal with denial-of-service attacks by verifying the source IP. When an IP packet passes through a router, iTrace is generated with a low probability of about 1/20000 and sent to the destination. Lee et al propose to use cumulative IP information to verify the true IP packet origin [6]. When a router receives a IP packet and forward it, it generates an iTrace message and appends its own IP address and this iTrace message is sent to the next hop instead to the destination. When a router receives an iTrace message, it appends its own IP address to the iTrace message. To improve the efficiency and usefulness of original iTrace, Mankin et al. propose the enhanced “Intension-Driven” iTrace with the indication for useful and valuable iTrace messages [1]. However, at best these message simply record the path of links and routers that packets may have taken. They provide no information on why a router selected a particular next hop.

4. ENHANCED BGP ITRACE

To provide reliable and fault-tolerant BGP routing protocol, we need to add appropriate mechanisms for monitoring and authenticating paths. As we reviewed in previous section, cryptography-based protection mechanisms, such as DNSSEC, S-BGP, SPV etc, require significant computation and space overhead in the router. It is too heavy to be deployed in the current Internet but is regarded more secure against deliberate attacks. On the other hand, non cryptography-based protection techniques, such as path filtering, ICMP Traceback etc, which utilize certain properties from the network infrastructure to guard against faults and attacks, are generally regarded ineffective but more deployable than cryptography-based techniques. As we studied in Section 3, there is no single complete solution to provide security and resilience to routing infrastructure.

Path vector protocols are similar to the distance vector protocol, except each routing update includes a list of routers on the route instead of using the metric. Path information provides routers with partial information about topological connectivity. By default, a path vector protocol will choose a route with the shortest path but policies may specify specific routers to prefer or to avoid. Therefore, a node may want to verify each hop that the routing update has

traversed as recorded in the path. A general way to perform verification is each node adds an authentication information in the packet and the recipient individually verify each authentication information. This approach requires the network overhead to carry a message authentication code (MAC) and to verify it for each node in the path. We want to avoid using this approach with Internet topology based verification technique.

BGP is a policy-based routing protocol and each AS chooses the path among the multiple routes it receives from its neighbors for the same prefix according to its own criteria. An AS also can apply policy when exporting a route. Generally, ASes filter incoming or outgoing announcements to implement policies such as peering and transit. Filtering can be implemented using prefix filters, access lists, and route maps. Using these basic primitives and a few others, an AS can control the flow of announcements between its routers and their BGP peers [10]. We use the advanced filtering technique with well-defined filtering rules and ICMP traceback to provide both path and origin validation. However, adding more functionality into routers is not a recommendable approach since routers already handle many complicated functions. Therefore, our approach suggests to use a separate server or a process to provide security mechanism.

4.1. AS-PATH VERIFICATION WITH ICMP TRACEBACK MESSAGES

Routers need a mechanism to authenticate each routing information that is received from its peers not only for dynamic filter information update, but also for both origin and path validation. Ideally, each router has to authenticate any route announcement and update message before accepting it to detect false routing information. However, authentication of both route announcements and update messages is not properly applied current BGP protocol.

Our approach uses ICMP Traceback (iTrace) message with a small modification for this purpose. Instead of authenticating BGP announcement messages and update messages, we use actual data traffic to collect proper connectivity information and AS-PATH information for AS-PATH and prefix origin validation. When data packets traverse along the route, each router on the path generates iTrace messages. Those iTrace messages have the information of traced packet source and destination address, previous link, and the AS-PATH which each router finds in its own routing table to reach the destination. This solution fully relies on the Internet topology and is efficiently verify the AS-PATH and prefix origin without using any cryptography. ICMP Traceback message is carried in an ICMP packet. Figure 1 presents the current traceback message format and Figure 2 shows the individual element structure. ICMP Traceback message body consists of a series of individual elements. More details are found in [2].

Type (1byte)	Code = 0 (1byte)	Checksum (2bytes)
Message body		

Figure 1. ICMP Traceback Message Format

TAG (1byte)	LENGTH (2byte)	VALUE (1byte)
-------------	----------------	---------------

Figure 2. Message Element Structure

Table 1. ICMP Traceback Tags [2]

Tag	Element Name
0x01	Back Link
0x02	Forward Link
0x03	Interface Name
0x04	IPv4 Address Pair
0x05	IPv6 Address Pair
0x06	MAC Address Pair
0x07	Operator-Defined Link Identifier
0x08	Timestamp
0x09	Traced Packet Content
0x0A	Probability
0x0B	RouterId
0x0C	HMAC Authentication Data
0x0D	Key Discloser List
0x0E	Key Discloser
0x0F	Public-Key Information
0x10	Traced Packet Source Address
0x11	Traced Packet Destination Address
0x12	AS-PATH Information

Table 1 is the list of tags for message elements. We add last three tags, **0x10** for **Traced Packet Source Address**, **0x11** for **Traced Packet Destination Address**, and **0x12** for **AS-PATH** information. Our modified iTrace message has important BGP information which consists of router's previous link and AS-PATH from iTrace generator to the destination which is found in AS router's routing table. Most of elements in table 1 are defined in [2]. Here we explain newly suggested tags.

Traced Packet Source Address (TAG = 0x10) / Traced Packet Destination Address (TAG = 0x11): This element contains the traced packet source address / destination address which is a 4-octet for IPV4 address or 6-octet for IPV6

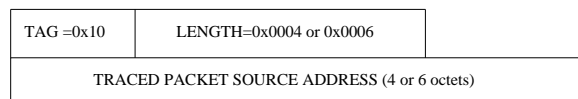


Figure 3. Traced Packet Source Address element format

address; hence the LENGTH field is 4-octet (0x0004) or 6-octet (0x0006). The element format is presented in Figure 3.

AS-PATH Information (TAG = 0x12): This element contains AS-PATH information which is found in BGP routing table. The length of the element is variable since the number of ASes on the path is not fixed. The element format is almost same as Figure 3 except LENGTH(variable) and VALUE(variable length).

We use **Back Link** element for link connectivity information from the perspective of the iTrace message generator. In the value field, we add AS number pair for one of the subelements.

Figure 4 shows how our solution works for the path validation. In this example, the CSU web server (129.82.100.64) is connected to AS1. AS-PATH from UCLA (131.179.96.130) to CSU web server is [AS8 AS7 AS6 AS1]. When UCLA client sends data to CSU web server, the data traffic traverses this path which is presented with solid line with arrows. When a data packet is sent by a client from a UCLA machine, all routers along the path (AS8, AS7, AS6, AS1) have chances to generate iTrace messages with probability of 1/20000. When the data packet traverses the AS7 router, it generates iTrace messages with the data packet's source address (131.179.96.130) and data packet's destination address (129.82.100.64), its previous link as (AS8 AS7) and the AS-PATH from itself to the destination [AS7 AS6 AS1]. This AS-PATH is found from AS7's BGP routing table. When AS7 router forwards the data packet, it generates two identical iTrace messages. One iTrace message is attached to ICMP header which has a source as AS7 itself and a destination as the same as the data packet's destination (129.82.100.64). The other iTrace message is attached to ICMP header which has a source as AS7 itself but a destination as an arbitrary node (in this example, it is AS3) which hopefully has different path to reach the destination. When AS3 router receives iTrace message, it simply changes ICMP header to send this iTrace message to the data packet's destination. The new ICMP header has a source as AS3 and a destination as 129.82.100.64. How to pick an arbitrary node to send the iTrace message is not discussed here. This problem is out of the scope of the present paper. Here, we just assume a random node is selected by the router with a proper manner. The only restriction of picking a node is this node should know how to handle iTrace messages. Other intermediate AS routers do exactly same thing as AS7 does when they propagate the data packet to the destination. However, the iTrace

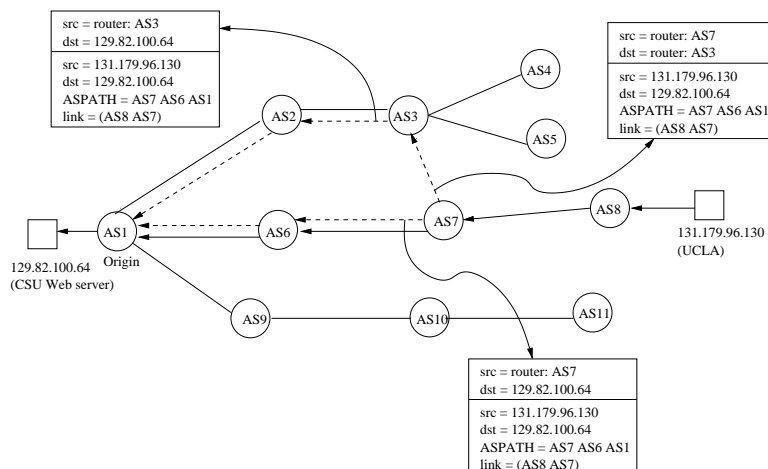


Figure 4. Valid Path Verification with ICMP Traceback messages

messages generated by each router has slightly different information. The one iTrace message, which is received by AS3, traverse along the path, [AS3 AS2 AS1], to reach the destination. The other iTrace message, which is directly sent to the destination follows the path, [AS7 AS6 AS1]. When the data packet arrives in AS6, the router does same thing as AS7 did to generate and send iTrace messages. All other routers (AS4, AS5, AS9, AS10, AS11) never see either data packets and iTrace messages.

In the destination, the router first checks each iTrace message's source field. It finds three different iTrace messages with the same source, 131.179.96.130. One is generated by AS6, another is generated by AS7 and the third is generated by AS8. The router constructs the path from source to destination based on link informations, which are Link() (this means client is directly connected), Link(AS8 AS7) and Link(AS7 AS6), and path informations which are [AS8 AS7 AS6 AS1], [AS7 AS6 AS1] and [AS6 AS1]. If there are no AS-PATH conflict, the router regards AS-PATH, [AS8 AS7 AS6 AS1], as a valid path from UCLA (131.179.96.130) to CSU web server (129.82.100.64).

The destination router may construct either a path tree or a path set for all source and destination pairs. If the destination uses a path tree, the router build a path tree from the information which is collected by all iTrace messages it receives. The path tree has a root as itself and leaves are the sources of data packets. Each path on the tree from the root to a leaf corresponds to each AS-PATH. If the destination uses a path set, it make a collection of paths from all sources. The decision between constructing all paths from sources to this node and building one path tree is an implementation issue. It depends on efficiency and overhead of space and performance to implement.

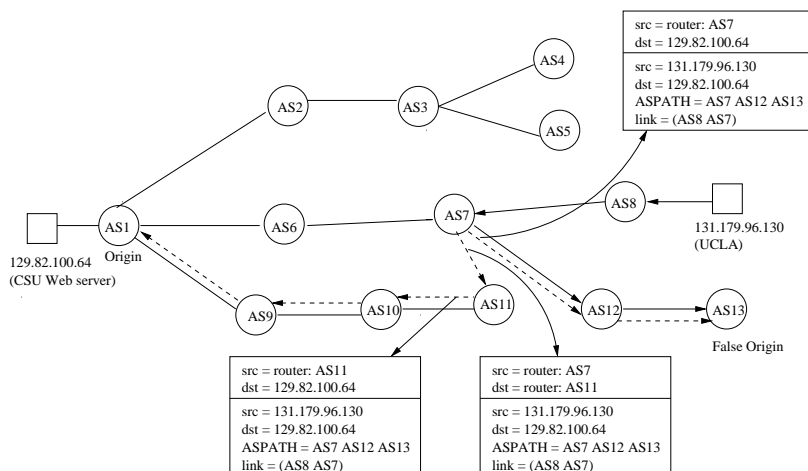


Figure 5. Invalid Path with False Origin Example

Whenever the destination node receives iTrace messages, it compares new information with previous information. Any inconsistency triggers an alarm. There are three different situations and the reaction of the destination should be different based on inconsistent types and reasons. The first one is AS-PATH is not directly connected to the destination. For example, the destination node, AS3, gets the iTrace message with AS-PATH, [AS1 AS2 AS3] and AS2 is not its next hop neighbor. This is very serious situation since it is an obvious sign of attack. In this case, the router immediately sets a flag and sends an emergency message to a system operator. The second one is AS-PATH is not consistent which means it does not match any previous AS-PATH information. This case can be interpreted as two possible way. One is an attack case in which a false origin or malicious router sends a wrong reachability information to its neighbors. Misconfiguration is also possible in this case. However, we do not distinguish misconfiguration from an attack since the effects are same. The other is routing convergence occurs this inconsistency when a link or a node fails. For both cases, the destination produces a warning message and more verification processes are required to decide the reason. The third one is one router on the path announces a wrong AS-PATH information to make the AS-PATH is longer than the real one. This scenario occurs when a router misconfigures the path to reach the destination or intentionally injects a wrong reachability information. In this case, our approach detects that false AS-PATH based on missing path derivation. Because real data traffic does not traverse routers which are not on the path, the destination never gets the iTrace messages from them. Next two examples show what are the possible scenarios for last two cases and how AS-PATH validation with iTrace works to detect them.

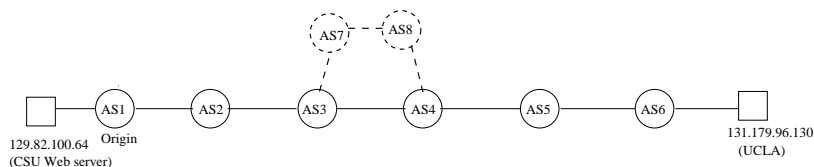


Figure 6. Invalid Path with False Reachability Information Example

Figure 5 presents a possible attack scenario for the first case. AS13 is a false origin which impersonates itself as the owner of CSU web server. In this case, AS-PATH to reach the destination 129.82.100.64 is [AS8 AS7 AS12 AS13]. The data traffic from UCLA (131.179.96.130) uses this false path. Even if the right path to reach from the source to the destination in this example is [AS1, AS6, AS7, AS8], since the intermediate routers on the false path can not see the whole network topology, they simply propagates all data packets, which are sent from UCLA, to the wrong destination. All of them generate iTrace messages and they have wrong path information. In Figure 5, AS7 generates two iTrace messages which have wrong AS-PATH information [AS7 AS12 AS13]. One is sent to the false destination and the other is sent to an arbitrary node AS11. AS11 sends this iTrace message to the right destination and now the destination detects path inconsistency with this message. It is possible for AS11 to send iTrace message to the false destination. However, because of the rich connectivity of the Internet, there is high probability that iTrace message is sent to the node which has the path to reach the correct destination. If iTrace message are sent as far as possible from the iTrace generator, it has good chances to reach the correct destination.

In the real destination, the router examines iTrace messages and finds AS-PATH is [AS7 AS12 AS13] which is generated from AS7. The router detects this information is not correct since AS13 is not itself and AS12 is not its NEXT-HOP. This is an obvious attack. It sets a flag and sends a report to a system operator. With this iTrace message, the destination node not only verifies wrong AS-PATH but also detect and locate a false origin.

Figure 6 shows a third case example. In this example, AS-PATH from UCLA to CSU is [AS6 AS5 AS4 AS3 AS2 AS1]. Somehow AS4 has AS-PATH to reach CSU web server is [AS4 AS8 AS7 AS3 AS2 AS1]. Based on this reachability information, AS5 has AS-PATH to reach the same destination as [AS5 AS4 AS8 AS7 AS3 AS2 AS1]. When data packets are sent from UCLA, all routers along the path generate iTrace messages. When AS1 collects iTrace messages and examines each iTrace message, it gets AS-PATH information as in Table 2. Nothing is inconsistent but the destination never gets iTrace message originated from AS7 or AS8. During reasonably long time if the destination does not receive any direct AS-PATH information from both AS7

Table 2. AS-PATHs which the destination collects

<i>iTrace Originator</i>	AS-PATH
AS2	[AS2 AS1]
AS3	[AS3 AS2 AS1]
AS4	[AS4 AS8 AS7 AS2 AS1]
AS5	[AS5 AS4 AS8 AS7 AS2 AS1]

and AS8, the destination suspects it is possible both AS7 and AS8 are not on the path which data packets traverse. The reason may be AS4 gets wrong reachability information from its neighbors or injects it by itself. Only based on this information, the reason can't be determined but it knows AS-PATHs from both AS4 and AS5 may not be correct. In this case, the destination triggers an alarm and notifies this observation to the operator. Further investigation is required to figure out this problem.

4.1.1 iTrace Generation. iTrace messages are generated by the routers on the path along which the real data traffic traverse. The generation procedures and implementation requirement of message generation are described in [2].

4.1.2 AS-PATH and origin Validation. Our AS-PATH validation is different from other AS-PATH validation techniques which try to authenticate AS-PATH information in BGP routing announcement or update messages. They need another mechanism to validate the prefix origin since AS-PATH validation does not guarantee the authentication of prefix origin. With our approach, the destination router independently derives AS-PATH from iTrace messages based on real data traffic. Each AS-PATH information from iTrace messages provides complete or partial view of path from source to destination. Since a prefix origin correspond to the last router of AS-PATH, suggested approach does not require any separate validation process.

Algorithm 1 AS-PATH Validation Algorithm

Input: *iTrace messages*

Output: *report message*

Procedure *ASPathValidation*

begin

 /* *longest(s, d)* is longest AS-PATH from source
 (*s*) to destination (*d*),

longestSet is a collection of *longest(s, d)* */

longest(s, d) = null; longestSet = {}; tracedAS = {}

timer = 5min

```

while forever do
  switch (event)
    event an iTrace message has arrived do
      begin
        remove the ICMP header
        get (s, d) source and destination of iTrace message
        get ASPATH from iTrace message
        get sendAS from iTrace message
        get longest(s, d) from longestSet
        /* Check if AS-PATH is directly connected with itself */
        if the last link of ASPATH  $\neq$  NEXTHOP
          /* this is an attack */
          set a flag and send an emergency message to the operator
        else
          if ASPATH is subpath of longest(s, d)
            tracedAS = tracedAS  $\cup$  sendAS
            /* current longest path is shorter than ASPATH */
          else if longest(s, d) is subpath of ASPATH
            longestSet = longestSet - longest(s, d)
            longest(s, d) = ASPATH
            tracedAS = tracedAS  $\cup$  sendAS
            longestSet = longestSet  $\cup$  longest(s, d)
          else
            /* AS-PATH is inconsistent */
            send inconsistent path warning message to operator
          endif
        endif
      end
    event timer is expired do
      begin
        /* there are some subpaths which are never received */
        if  $\exists AS \in longest(s, d)$  and  $AS \notin tracedAS$ 
          send unreceived subpath warning message to operator
          longest(s, d) = null; longestSet = {}
          tracedAS = {}
          set timer with 5 minutes
        endif
      end
    endwhile
  end

```

4.2. FILTERING

Effective filtering techniques help the BGP security since BGP is policy based protocol and uses routing filtering to enforce various routing policies. Applying filters to incoming BGP route advertisement is very common technique for routing configuration with using optional BGP attributes such as weight, local preference, and so on. These filters prevent the router from accepting inappropriate route announcements which violate the policies. However, getting complete routing filters is very difficult since it requires to know all routing policies and relationship of ASes, and to view a global AS topology.

When a router receives UPDATE messages, ingress filters are used to check the validity of the information and when UPDATE messages are sent to peers, egress filters are used to control which routes are announced [9]. Bellovin et al. applied filters to control the growth of BGP routing tables [13]. Their filtering rules are related with prefix allocation rules which are decided by the Regional Internet Registries (RIRs) or operators. To know the owner of each address block, Internet Routing Registries (IRRs) are commonly used. However, one of the problem to use IRR is its databases are not well-maintained and updated because it is difficult to keep track of the lists for the large number of routers. Therefore the records in IRR are often inaccurate.

In this paper, we use the route filtering technique, which was originally proposed in [7], with both general policies and advance filtering mechanism for routers to help verify the validity of the route announcement for the security purpose. Wang et al. proposed path-filtering approach for protecting BGP routes to the top-level DNS servers [7]. They used path-filtering to block invalid routes by restricting the routes to top-level DNS servers to change within a set of established routes, based on statistical analysis over history. Their approach can be deployed only for the top-level DNS servers (or for the well-known destinations) since their approach truly depends on the high degree of redundancy in the top-level DNS system and the stability in network connectivity to the server locations. This scheme effectively filters out potentially invalid paths based on the route history. The path filtering restricts the changes of the routes to the top-level DNS servers within a set of verified persistent routes. It provides resilient network infrastructure since it does not rely on the cryptographical techniques and IRRs. However their verification mechanism, which is used to identify new valid paths, only works for DNS system since it utilizes the redundancy of the DNS systems.

We cannot directly use this approach for all BGP routers because BGP routers may not have high redundancy and network connection stability. For the general BGP routing protection, our design improves this limitation with more practical verification technique with ICMP Traceback. Suggested route filters verify not only AS-PATH but also the origin AS of a route to which the prefix really

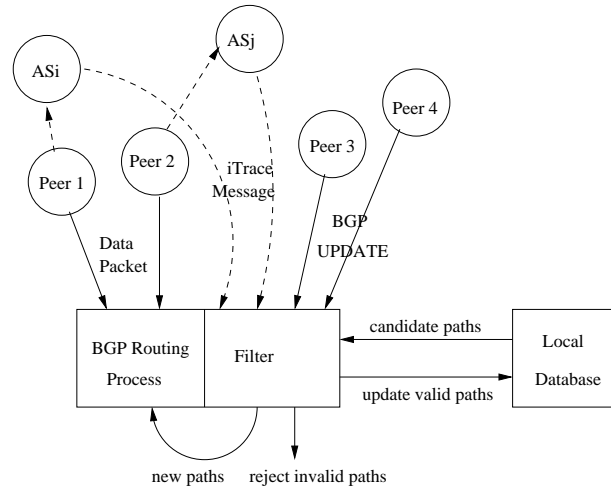


Figure 7. Advanced Path Filter Design

belongs. The destination independently construct AS-PATH with its perspective based on iTrace messages and it does not depend on BGP announcement messages and UPDATE messages. We use almost same filtering processes which are proposed in [7]. To make the filters more complete, we modify the verification process with a practical and efficient technique (See Section 4.1).

Figure 7 shows overall our approach. BGP router has a separate filter and a local database. The filter aggressively verifies all BGP UPDATE messages based on general policies, path history, MOAS conflict information, and ICMP traceback messages. Only verified new path will be added into the BGP routing table and will be updated in the local database.

General Policies: BGP allows many routing policies that control the type of routes announced and accepted by a BGP peers. Operators filter routes which violate global guidelines that are published by Regional Internet Registries (RIRs).

Statistical analysis over history: Wang et al. [7] used heuristics derived from routing operations to adjust the potential routes over time. Their approach keeps a set of back-up paths, which are already verified, for later use. Keeping a set of pre-verified back-up paths is good to reduce the replacement time in case of path withdraw announcement. We use same technique. All back-up path informations are kept in the local routing database and periodically updated and verified.

MOAS conflict based verification: Multiple origin AS (MOAS) conflicts occur when multiple ASes announce themselves as the origin of a particular prefix. It occurs for various reasons including misconfiguration, multihoming,

or exchange points. Multihoming and exchange points cause valid MOAS cases and operational errors, incorrect announcements, or malicious attacks result in invalid MOAS cases. Zhoa et al proposed to use a list of MOAS to detect MOAS conflict [17]. Their solution is to create a list of multiple ASes who originate a particular IP address prefix and attach this list to the route announcements. Our approach detects MOAS conflict without adding any additional information to the route announcements. Local routing database keeps track of the list of the valid MOASes. When a router receives the route announcements from the multiple origins, filters check each path and origin validity. Route validation process also checks the MOAS list which the local routing database keeps. If the path or origin of the announcement is not valid, or if the origin is not in the MOAS list, filters reject the announcement.

AS-PATH Validation with ICMP Traceback messages: This process is explained in Section 4.1.

Local Routing Database: The basic idea of path-filtering mechanism is using a set of pre-verified back-up routes to accept as a valid route in case the current route is withdrawn. To keep track of possible candidate paths information, we use a separate local database which is dynamically updated and periodically verified by filtering process. Each router has its own database and it makes our solution work in fully distributed manner.

Aggressive filtering may make some parts of the Internet address space unreachable then result in a denial of service to legitimate, non-hostile routers. We need to check this by quantifying the potential/possible space reachability to prevent reachability problem. The measurement and evaluation of this problem is not included in this paper. One alternative way to avoid reachability problem is when the router receives the UPDATE message, it immediately adds and updates the new path before the verification process. Then the filtering process verifies the validity of the path. If the filter finds the path is invalid, it simply deletes the updated path from both the routing table and the local database. This way concerns performance and availability more than security. However, we assume the way a filter verifies the path before update is more desirable as long as verification time is reasonable. This decision is really related with the trade-off between security concerns and performance concerns.

5. SUMMARY AND CONCLUSION

The suggested approach is an integration of several existing partial solutions which have been proposed in previous work. While none of these provides a perfect solution for secure, reliable, robust, and fault-tolerant BGP protocol, the combination of them may provide a more effective, efficient and concrete solution.

We suggest the use of a separate process or server with routers to make BGP protocol more secure and reliable. The proposed approach verifies the UPDATE messages with well-defined filtering technique and filtering rules, which is similar to the way suggested in [7]. However, filtering itself does not guarantee the complete correctness of each path information which is received by neighbors. To provide efficient path validation mechanisms, we use ICMP traceback (iTrace) message with small modification. Our suggested iTrace message has important BGP information, such as Source AS, link connectivity information, and AS-PATH information to probe the validity of the paths. The important difference between our approach and other path validation approaches is it uses *real data traffic* to validate the correct path.

With our solution, filtering process, local database management, path and origin verification work in a fully distributed manner and that guarantees good availability and scalability.

We try to avoid using cryptographic technique since public key scheme requires significant space and time overhead to generate and to verify the signature. It also requires Public key infrastructure (PKI). Our solution truly depends on the distributed nature of the Internet to spread the correct information and to corroborate them. It utilizes the Internet topology to detect impersonated routes and invalid paths.

Recent studies show that implementation bugs and configuration mistakes are still responsible for a significant fraction of the traffic [10]. However, to distinguish malicious attacks from routing misconfiguration is not necessary, even if it is possible, since both of them cause almost same reachability and BGP convergence problems. In this paper, we do not include the way to distinguish between malicious attacks and misconfiguration.

We view our approach as a proper security mechanism without operational degradation of BGP protocol. We hope our work provides some improvement for BGP routing protocol with incremental deployability and scalability to adapt well to the real world.

References

- [1] A. Mankin, D. Massey, C.L. Wu, L. Zhang. On Design and Evaluation of Intention-Driven ICMP Traceback. In *IEEE International Conference on Computer Communications and Networks (ICCCN)*, October 2001.
- [2] Steven M. Bellovin. ICMP Traceback Messages. Internet Draft, March 2001.
- [3] D. Pei, D. Massey, and L. Zhang. Detection of Invalid Routing Announcements in the RIP Protocol. In *IEEE Global Communication conference (Globecom)*, December 2003.
- [4] D. Pei, D. Massey and L.Zhang. A Framework for Resilient Internet Routing Protocol. *IEEE Network*, 18(2):5–12, April 2004.
- [5] 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 *ISOC NDSS '03*, February 2003.
- [6] Henry C.J.Lee, Vrizlynn L.L.Thing, Yi Xu, and Miao Ma. ICMP Traceback with Cumulative Path, An Efficient Solution for IP Traceback. In *5th International Conference on Information and Communications Security*, October 2003.
 - [7] L. Wang, X.Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S.F. Wu and L. Zhang. Protecting BGP Routes to Top Level DNS Server. *IEEE Transactions on Paralle and Distributed Systems*, 14(9):851–860, September 2003.
 - [8] J. Ng. Extension to BGP to Support Secure Origin BGP, October 2002.
 - [9] O. Nordstrom and C. Dovrolis. Beware of BGP attacks. *Computer Communications*, pages 1–8, April 2004.
 - [10] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *ACM SIGCOMM 2002*, August 2002.
 - [11] R. Perlman. *Network layer protocols with Byzantine roubustness*. PhD thesis, MIT Lab. for Computer Science, 1988.
 - [12] Y. Rekhter and T. Li. Border Gateway Protocol 4. RFC 1771, July 1995.
 - [13] S. Bellovin, R. Bush, T. G. Griffin, and J. Rexford. Slowing Routing Table Growth by Filtering Based on Address Allocation Policies, June 2001.
 - [14] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (S-BGP). *IEEE JSAC Special Issue on Network Security*, 2000.
 - [15] V. N. Padmanabhan and D. R. Simon. Secure Tracerout to Detect Faulty or Malicious Routing. In *ACM SIGCOMM Computer Communication Review*, pages 77–82, 2003.
 - [16] X. Zhao, D. Pei, L. Wag, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *SIGCOMM Internet Measurement Workshop*, November 2001.
 - [17] X. Zhao, D. Pei, L. Wag, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Detection of Invalid Routing Annouements in the Internet. In *International Conference on Network and Distributed Systems (DSN)*, June 2002.
 - [18] Y.C. Hu, A. Perrig and M. Sirbu. SPV: Secure Path Vector Routing for Secure BGP. In *ACM SIGCOMM Computer Communication Review*, pages 179–192, October 2004.