Chapter 14

GLOBAL INTERNET ROUTING FORENSICS

Validation of BGP Paths using ICMP Traceback

Eunjong Kim, Dan Massey and Indrajit Ray

Abstract The Border Gateway Protocol (BGP), the Internet's global routing protocol, lacks basic authentication and monitoring functionality. Thus, false routing information can be introduced into the Internet, which can cause the total collapse of packet forwarding and lead to denial-of-service or misdirected traffic. While it may be impossible to prevent such an attack, we seek to provide the routing infrastructure with a mechanism for identifying false paths through efficient validation, proper recording and forensic analysis of routing data. Towards this end, we propose a novel BGP path verification technique using ICMP traceback messages that has been extended to include AS-PATH and link connectivity information. The approach can be easily deployed as it does not require modifications to BGP.

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 sectors and have a major impact on the economy. In response, a variety of end system security techniques, such as encrypted connections and VPNs have been proposed. However, almost all of these 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. This paper examines techniques for detecting invalid routes in the Internet infrastructure and

presents an effective approach for gathering and extracting routing data from the network that can be used for forensic analysis.

At the global infrastructure level, the Internet consists of thousands of Autonomous Systems (ASs), each identified by a unique number. An AS can be viewed as a group of links and routers that are under the same administrative control. The ASs are responsible for routing information over the Internet backbone. The Border Gateway Protocol (BGP) [5] is the de facto inter-AS routing protocol; it is used to exchange reachability information between ASs. BGP is designed to cope with events that alter the structure of the Internet, such as the addition of new links and new ASs, the failure (temporary or long lasting) of links, and changes in routing policies. However, BGP contains very limited security mechanisms and thus presents several interesting challenges for path validation and routing forensics.

BGP implicitly assumes that routers advertise valid information. For example, suppose that AS 12145 (Colorado State University) incorrectly (maliciously) reports that it has a 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 spoof the `www.largecompany.com` website. As a result, `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 very long time for the changes to propagate throughout the Internet. In addition, owing to the large number of BGP destinations and the large volume of BGP routing changes, a particular BGP path change is unlikely to trigger any alarms at remote sites. Nonetheless, such actions have the potential to significantly disrupt 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.

This paper presents an approach for monitoring, gathering and validating a route to a destination. The technique works as follows. Suppose AS_1 has incorrect path information for AS_2. This can be due to one of several reasons, e.g., malicious advertisement of wrong path information by a neighboring AS of AS_1 or misconfiguration at AS_1. Under 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 ASs have invalid path information about it. If AS_1 (and the

other ASs) are reachable from AS_2, then AS_2 can alert these ASs that incorrect path information has been introduced.

The proposed approach uses ICMP (Internet Control Message Protocol) traceback messages. As data packets flow through routers, occasional packets (one in twenty thousand) generate ICMP traceback messages. 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, our methodology focuses on paths that actively carry data traffic. There may be more than 18,000 ASs that have some path to `www.largecompany.com`, but relatively few of these sites may be actively sending data traffic. By using the ICMP traceback mechanism, monitoring and validation messages are only sent for paths that are actively in use. The ICMP traceback messages are enhanced with AS-PATH information and link connectivity information. Also, traceback messages are sent along multiple (ideally disjoint) paths to reduce the probability that packets are (maliciously or otherwise) dropped or corrupted. Thus, a router can dynamically keep track of paths used to reach the destination, monitor routing changes for the actively used paths to this destination, and produce logs that can be used to reconstruct routes in the event of a suspected attack. As a side-effect, this approach provides a more fault-tolerant, fault-resilient, reliable and secure BGP routing for the Internet infrastructure.

2. Enhanced BGP iTrace

In the original ICMP traceback proposal [1], 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 address. When an IP packet passes through a router, iTrace is generated with a low probability of about 1/20,000 and sent to the destination. Lee, *et al.* [2] propose using cumulative IP information to verify the true IP packet origin. When a router receives a IP packet and forwards it, it generates an iTrace message and appends its own IP address; this iTrace message is sent to the next hop instead of to the destination. When a router receives an iTrace message, it appends its own IP address to the iTrace message. Mankin, *et al.* [4] have proposed an "intension-driven" version of iTrace. However, at best, their messages 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. To provide reliable and fault-tolerant BGP routing protocol, it is necessary to add appropriate mechanisms for monitoring and authenticating paths. 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 a policy when exporting a route. Generally, ASs 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 [3]. Our approach uses advanced filtering and ICMP traceback to provide both path and origin validation. However, adding more functionality into routers is not recommended as routers already handle many complicated functions. Therefore, our approach requires a separate server or a process that provide security mechanisms.

2.1 Modified ICMP Traceback Messages

Our approach uses an extended form of the ICMP traceback (iTrace) message. Instead of authenticating BGP announcement messages and updating messages, it uses the actual data traffic to collect proper connectivity information for AS-PATH and prefix origin validation. As data packets traverse a route, each router on the path generates iTrace messages. These iTrace messages contain information about the traced packet source and destination address, previous link, and the AS-PATH which each router finds in its routing table to reach the destination.

Table 1 presents the list of tags for message elements. We add the last three tags, *0x10* for *Traced Packet Source Address*, *0x11* for *Traced Packet Destination Address*, and *0x12* for *AS-PATH* information. The other elements in Table 1 are defined in [1]. In the following, we briefly discuss the three new 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 4 octets for an IPv4 address and 6 octets for an IPv6 address; hence, the LENGTH field is either 0x0004 or 0x0006. The element format is presented in Figure 1.

AS-PATH Information (TAG = 0x12): This element contains AS-PATH information, which is found in a BGP routing table. The length of the element is variable since the number of ASs on the path is not fixed. The element format is almost the same as in Figure 1 except for the LENGTH(variable) and VALUE(variable length) fields. The Back Link element is used for link connectivity information from the perspective of

Table 1. ICMP traceback tags [1].

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*

the iTrace message generator. In the VALUE field, an AS number pair is added for one of the sub elements.

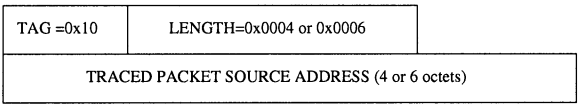

Figure 1. Traced packet source address element format.

2.2 AS-PATH Validation

Figure 2 shows how the approach works for path validation. In the example, the CSU web server (129.82.100.64) is connected to AS1. The AS-PATH from UCLA (131.179.96.130) to the CSU web server is [AS8 AS7 AS6 AS1]. When the UCLA client sends data to the CSU web server, the data traffic traverses this path (solid line with arrows). When a data packet is sent by a client from a UCLA machine, all the routers along the path (AS8, AS7, AS6, AS1) generate iTrace messages with a probability of 1/20,000. When the data packet traverses the AS7 router, it generates iTrace messages with the data packet's source address (131.179.96.130) and the data packet's destination address (129.82.100.64), its previous

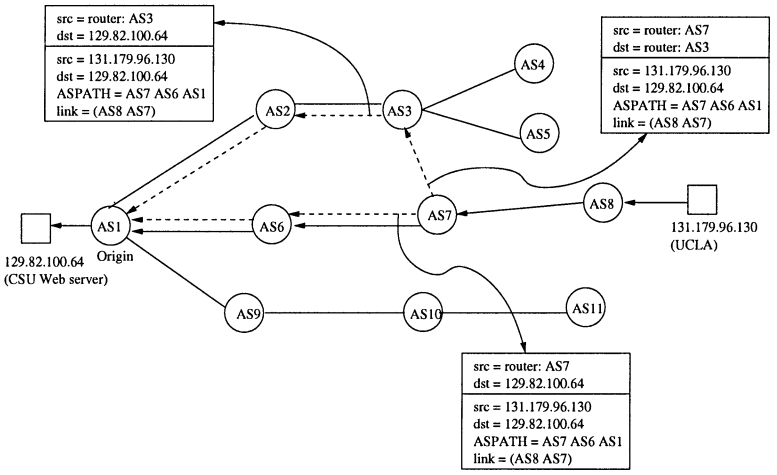

Figure 2. Valid path verification with ICMP traceback messages.

link as (AS8 AS7) and the AS-PATH from itself to the destination [AS7 AS6 AS1]. This AS-PATH is found in AS7's BGP routing table. When router AS7 forwards the data packet, it generates two identical iTrace messages. One iTrace message is attached to the ICMP header, which has AS7 as its source and the same destination as the data packet's destination (129.82.100.64). The other iTrace message is attached to the ICMP header which has AS7 as its source but a destination as an arbitrary node (AS3 in example), which hopefully has different path to reach the destination. When AS3 receives an iTrace message, it simply changes the ICMP header to send the iTrace message to the data packet's destination. The new ICMP header has AS3 as its source and 129.82.100.64 as its destination. We do not discuss how a node is picked to send the iTrace message as it is outside the scope of this paper. Instead, we simply assume that a random node is selected by the router; the only restriction is that the node should know how to handle iTrace messages. Other intermediate AS routers operate similarly to AS7 when they propagate data packets to their destinations. However, the iTrace messages generated by each router have slightly different information. One iTrace message, which is received by AS3, traverses 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 follows the same procedure as AS7 to generate and send iTrace messages. All the other routers (AS4, AS5, AS9, AS10, AS11) do not see the data packets and iTrace messages.

At 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 the source to the destination based on link information: Link() (this means client is directly connected), Link(AS8 AS7) and Link(AS7 AS6), and path information: [AS8 AS7 AS6 AS1], [AS7 AS6 AS1] and [AS6 AS1]. If there are no AS-PATH conflicts, the router regards AS-PATH, [AS8 AS7 AS6 AS1], as a valid path from UCLA (131.179.96.130) to the CSU web server (129.82.100.64).

The destination router constructs a path tree or a path set for all source and destination pairs. If the destination uses a path tree, the router builds a path tree from the information, which is collected by all the iTrace messages it receives. The path tree has itself as the root node; its leaves correspond to the source addresses of data packets. Each path on the tree from the root to a leaf corresponds to an AS-PATH. If the destination uses a path set, a collection of paths is created from all sources. The decision between constructing all paths from sources to this node and building one path tree is an implementation issue that depends on efficiency, space overhead and performance.

When a destination node receives an iTrace message, it compares the new information with previous information. Any inconsistency triggers an alarm. Three different situations can exist, and the reaction of the destination to each is different. The first is when AS-PATH is not directly connected to the destination, e.g., destination node, AS3, gets an iTrace message with AS-PATH: [AS1 AS2 AS3] and AS2 is not its next hop neighbor. This is an obvious sign of attack; therefore, the router immediately sets a flag and sends an emergency message to the system operator. The second situation is when AS-PATH is not consistent, i.e., it does not match any previous AS-PATH information. This can be interpreted in two possible ways: one is an attack in which a false origin or malicious router sends wrong reachability information to its neighbors, and the other is misconfiguration. However, we do not distinguish misconfiguration from an attack since the effects are same. The third situation occurs when one router on the path announces wrong AS-PATH information to make the AS-PATH longer than the real one. This occurs when a router misconfigures the path to reach the destination or intentionally injects wrong reachability information. In this case, our approach detects the 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 receives iTrace messages from them.

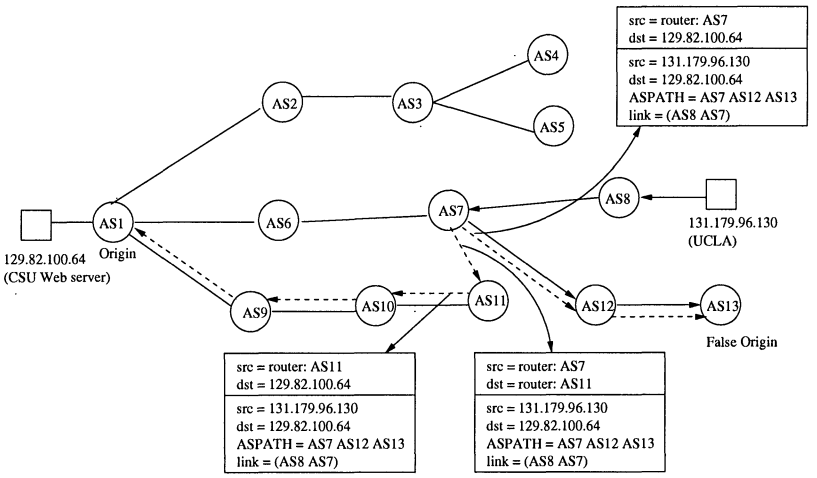

Figure 3. Invalid path with false origin.

In the following, we present examples of the scenarios and demonstrate how they can be detected via AS-PATH validation with iTrace.

2.3 BGP iTrace Under Attacks

Figure 3 presents a possible attack scenario. AS13 is a false origin which impersonates as the owner of the CSU web server. In this case, the 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 though the correct path in this example is [AS1, AS6, AS7, AS8], the intermediate routers on the false path simply propagate all the data packets sent from UCLA to the wrong destination. This is because these intermediate routers cannot see the entire network topology. All these routers generate iTrace messages with the wrong path information. AS7, in particular, generates two iTrace messages with the wrong AS-PATH – [AS7 AS12 AS13]. One of these iTrace messages is sent to the false destination AS13, and the other to the neighboring node AS11. AS11 forwards this iTrace message to the correct destination, which then detects a path inconsistency. It is quite possible that AS11 sends an iTrace message to the false destination. However, because of the rich connectivity of the Internet, there is high probability that an iTrace message is sent to a node that has the path to the correct destination. Indeed, if an iTrace message is sent as far as possible from the iTrace generator, the message has a good chance of reaching the correct destination.

When the iTrace message reaches the correct destination, the router notes that the AS-PATH is [AS7 AS12 AS13], which is generated by

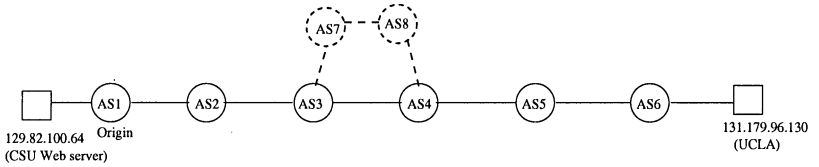

Figure 4. Invalid path with false reachability information.

Table 2. AS-PATH information collected by the the destination.

iTrace Originator	AS-PATH
AS2	[AS2 AS1]
AS3	[AS3 AS2 AS1]
AS4	[AS4 AS8 AS7 AS2 AS1]
AS5	[AS5 AS4 AS8 AS7 AS2 AS1]

AS7. The router recognizes that this information is incorrect as AS13 does not match the anticipated value (AS8), and AS12 is not its NEXT-HOP. This is an obvious attack; a flag is set and a report is sent to the system operator. Thus, with this iTrace message, the destination node is not only able to verify the incorrect AS-PATH, but also detect and locate the false origin.

Figure 4 presents another example. Here, AS-PATH from UCLA to CSU is [AS6 AS5 AS4 AS3 AS2 AS1]. Somehow, AS4 reflects that the AS-PATH to reach CSU web server is [AS4 AS8 AS7 AS3 AS2 AS1]. Based on this reachability information, AS5 has the AS-PATH to reach the same destination as [AS5 AS4 AS8 AS7 AS3 AS2 AS1]. When data packets are sent from UCLA, all the routers along the path generate iTrace messages. AS1 collects and examines each of these iTrace messages. The resulting accumulated AS-PATH information at AS1 is shown in Table 2. No inconsistencies are noted, but the destination never gets iTrace messages originating from AS7 or AS8. After a sufficiently long time, if the destination does not receive any direct AS-PATH information from both AS7 and AS8, the destination will suspect that neither AS7 nor AS8 are on the path that data packets traverse. The plausible causes at this stage are either that AS4 obtains incorrect reachability information from its neighbors or that AS4 injects this information itself. Based solely on the AS-PATH information, the cause cannot be precisely determined. In this case, the destination triggers an alarm and notifies the operator of this observation. Further analysis is required at this stage to diagnose the problem.

AS-PATH and AS Origin Validation Algorithm: Our AS-PATH validation approach differs from techniques that authenticate AS-PATH information in BGP routing announcement or update messages. These techniques need an additional mechanism to validate the prefix origin. This is because AS-PATH validation, by itself, does not guarantee the authentication of prefix origin. In our approach, the destination router independently derives AS-PATH from iTrace messages based on real traffic. Indeed, AS-PATH information from iTrace messages provides partial or complete views of a path from source to destination. Since a prefix origin corresponds to the last router of AS-PATH, our approach does not require a 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 ≠ 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 ∪ 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 ∪ sendAS
                longestSet = longestSet ∪ 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 ∃ AS ∈ longest(s, d) and AS ∉ tracedAS
              send unreceived subpath warning message to operator
              longest(s, d) = null; longestSet = {}
              tracedAS = {}
              set timer with 5 minutes
          endif
        end
    endwhile
end
```

3. Conclusions

This paper describes a technique for fortifying the Internet routing infrastructure with a mechanism to identify false path information. The approach, based on efficient validation, proper recording and forensic analysis of routing data, integrates several partial solutions that have been proposed elsewhere. The ICMP traceback (iTrace) is adapted to provide efficient path validation mechanisms. In particular, the iTrace message is modified to include important BGP information such as Source AS, link connectivity information and AS-PATH information. The iTrace message facilitates checking the validity of paths. A unique feature is that real traffic is used to validate paths. Furthermore, filtering, local database management, path and origin verification work in a fully distributed manner and guarantee good availability and scalability.

It is important to note that the proposed approach does not use cryptographic techniques. This is because public key schemes require an established PKI that involves significant overhead to generate and verify signatures; this affects scalability and deployability using the existing infrastructure. In contrast, our approach depends on the distributed nature of the Internet to spread the correct information and corroborate paths, and it uses the Internet topology to detect impersonated routes and invalid paths.

Recent studies have shown that implementation and/or misconfiguration errors are responsible for a significant portion of traffic [3]. However, in this work, we do not take any extra steps to differentiate between these errors and malicious attacks because both cause the same reachability and convergence problems.

The proposed approach provides security mechanisms without any operational degradation of BGP. Also, it facilitates incremental deployability and scalability that adapt well to the real world.

Notes

1. Currently, 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.

References

[1] S. Bellovin, ICMP traceback messages, Internet Draft, March 2001.

[2] H. Lee, V. Thing, Y. Xu and M. Ma, ICMP traceback with cumulative path: An efficient solution for IP traceback, *Proceedings of the Fifth International Conference on Information and Communications Security*, pp. 124-135, 2003.

[3] R. Mahajan, D. Wetherall and T. Anderson, Understanding BGP misconfiguration, *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pp. 3-16, 2002.

[4] A. Mankin, D. Massey, C. Wu and L. Zhang, On design and evaluation of intention-driven ICMP traceback, *Proceedings of the Tenth IEEE International Conference on Computer Communications and Networks*, pp. 159-165, 2001.

[5] Y. Rekhter and T. Li, Border Gateway Protocol 4, RFC 1771, July 1995.

[6] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu and L. Zhang, Protecting BGP routes to top level DNS server, *IEEE Transactions on Parallel and Distributed Systems*, vol. 14(9), pp. 851-860, 2003.