Topology-Based Detection of Anomalous BGP Messages

Christopher Kruegel, Darren Mutz, William Robertson, and Fredrik Valeur

Reliable Software Group
University of California, Santa Barbara
{chris,dhm,wkr,fredrik}@cs.ucsb.edu

Abstract. The Border Gateway Protocol (BGP) is a fundamental component of the current Internet infrastructure. Due to the inherent trust relationship between peers, control of a BGP router could enable an attacker to redirect traffic allowing man-in-the-middle attacks or to launch a large-scale denial of service. It is known that BGP has weaknesses that are fundamental to the protocol design. Many solutions to these weaknesses have been proposed, but most require resource intensive cryptographic operations and modifications to the existing protocol and router software. For this reason, none of them have been widely adopted. However, the threat necessitates an effective, immediate solution.

We propose a system that is capable of detecting malicious inter-domain routing update messages through passive monitoring of BGP traffic. This approach requires no protocol modifications and utilizes existing monitoring infrastructure. The technique relies on a model of the autonomous system connectivity to verify that route advertisements are consistent with the network topology. By identifying anomalous update messages, we prevent routers from accepting invalid routes. Utilizing data provided by the Route Views project, we demonstrate the ability of our system to distinguish between legitimate and potentially malicious traffic.

Keywords: Routing Security, BGP, Network Security

1 Introduction

Research in network security is mainly focused on the security of end hosts. Little attention has been paid to the underlying devices and protocols of the network itself. This has changed with the emergence of successful attacks against the infrastructure of the global Internet that resulted in major service interruptions. The services to handle the translation between domain names and IP addresses (such as the Domain Name System) and protocols to facilitate the exchange of reachability information (such as routing protocols) have been recognized as essential to correct network operation.

The Internet can be described as an interconnected collection of autonomous domains or local networks, each of which is subject to the administrative and technical policy of a single organization. There exist two types of routing protocols: *intra-domain* and *inter-domain* routing protocols. The task of intra-domain

G. Vigna, E. Jonsson, and C. Kruegel (Eds.): RAID 2003, LNCS 2820, pp. 17–35, 2003.

routing protocols is to ensure that hosts inside a single domain or local network can exchange traffic. The goal of inter-domain routing protocols, on the other hand, is to exchange reachability information between such domains. This enables hosts to communicate with peers that are located in different networks.

There are several different intra-domain protocols used today (e.g., RIP [19], OSPF [22]), while the Border Gateway Protocol (BGP) is the de facto standard for inter-domain routing.

Version 4 of the Border Gateway Protocol was introduced in RFC 1771 [28]. It specifies an inter-autonomous system routing protocol for IP networks. The definition given for an *autonomous system* (AS) is "*a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASes*".

The basic function of BGP is to enable autonomous systems to exchange reachability information that allows so-called BGP speakers to build an internal model of AS connectivity. This model is used to forward IP packets that are destined for receivers located in other ASes. The protocol includes information with each reachability message that specifies the autonomous systems along each advertised path, allowing implementations to prune routing loops. In addition, BGP supports the aggregation of path information (or routes) and utilizes CIDR (classless inter-domain routing) to decrease the size of the routing tables.

The protocol operates by having BGP speakers, usually routers, in different ASes exchange routing information with their *BGP peers* in the neighboring ASes. In addition to announcing its own routes, a BGP speaker also relays routing information received from its peers. By doing this, routing information is propagated to all BGP speakers throughout the Internet. The two basic operations of the BGP protocol are the announcement and the withdrawal of a route. The routing data itself is exchanged in `UPDATE` messages. Although BGP defines three other message types, none of these are directly related to the routing process. A route consists of a set of IP prefixes (stored in the `NLRI` – *network layer reachability information* – field of an `UPDATE` message), together with a set of attributes. When a route is announced, the sending BGP speaker informs the receiver that the IP prefixes specified in the `NLRI` field are reachable through the sending AS. The withdrawal process revokes a previous announcement and declares certain IP prefixes as no longer reachable via the AS. The most important attribute of an announcement is the `AS_PATH`. It specifies the path (i.e., the sequence of autonomous systems) that the route announcement has previously traversed before reaching that AS. Other attributes give information about the origin of a route or indicate whether routes have been aggregated at a previous AS.

Recently, a security analysis of BGP [23] and related threat models [5] pointed out two major areas of vulnerabilities of the inter-domain routing process.

One area includes threats that emanate from outsiders. Outsiders can disrupt established BGP peer connections and thereby launch denial of service attacks. They do not have privileges to influence the routing infrastructure directly, but can attempt to gain access to (break into) a legitimate router or impersonate a trusted BGP peer. Threats at this level usually do not aim at the routing protocol design but at the implementation level, in which bugs or vulnerabilities in routing software can be exploited to crash a machine or to elevate one's privileges. It might also be possible to bypass the authentication scheme to impersonate a legitimate BGP peer.

When an outsider is successful in compromising a trusted machine or an attacker already is in legitimate control of such a router, the focus shifts to direct threats from BGP peers. This area includes problems that occur when routers that legitimately participate in the routing infrastructure maliciously (or by accident) insert incorrect routing information. This can be the announcement of false IP address origins or invalid AS paths. Attacks at this level primarily focus on vulnerabilities in the routing protocol design and exploit the fact that there exists a significant level of trust between BGP peering partners. Invalid updates can propagate despite message filtering performed by many ASes, because it is often impossible to evaluate the validity of an update message given only local information. This might lead to worst-case scenarios where a single malicious or misconfigured router influences the routing state of the whole Internet.

We propose a technique that is capable of detecting malicious BGP updates utilizing geographical location data from the `whois` database and the topological information of an AS connectivity graph. By passively monitoring `UPDATE` messages, the connectivity graph is constructed by connecting two autonomous systems if traffic can be directly exchanged between them. Using this graph, we classify all autonomous system nodes as either *core* or *periphery* nodes. In general, core nodes represent the autonomous systems of the Internet backbone (such as large ISPs) while periphery nodes correspond to local providers, companies or universities. An important observation is that periphery AS nodes that are directly connected to each other are also close in terms of geographic distance. In most cases, peripheral autonomous systems have at most a few links to core nodes to obtain connectivity to distant networks and additionally peer only with partners in their geographic neighborhood. This observation leads to the determination that a valid `AS_PATH` contains at most a single sequence of core nodes, which must appear consecutively. That is, a path that has traversed core nodes and enters a periphery node never returns to the core of the graph. By checking the `AS_PATH` attribute of update messages, we can determine if the sequence of autonomous systems satisfies the constraints dictated by our observations and detect violations.

The structure of the paper is as follows. Section 2 presents related research in the area of routing security. Section 3 introduces the underlying threat model and discusses the attacks the system is designed to detect. Section 4 and Section 5 explain our proposed detection techniques. Section 6 provides experimental validation of important assumptions and reports on the results of our system. Section 7 outlines future work and Section 8 briefly concludes.

2 Related Work

Much research effort has focused on the security issues of intra-domain routing protocols [4, 24, 27, 30] and systems that perform intrusion detection for RIP [12, 21] as well as for OSPF [12, 26] have been proposed.

In contrast to intra-domain protocols, research on inter-domain protocols has concentrated on BGP, and its apparent weaknesses. Several authors have proposed extensions to the BGP protocol [14, 30] that attempt to authenticate routing information by means of cryptography. These modifications aim at countering threats from BGP peers that inject bogus information into the routing process, exploiting the fact that this information cannot be verified and, therefore, has to be trusted.

The most well-known approach is called the Secure Border Gateway Protocol (S-BGP) [13, 14] and operates as follows. During the propagation of an UPDATE message from AS to AS, each member on the path appends its information to the message and cryptographically signs the result before passing it along. This allows everyone in the chain to verify that the NLRI information is correct and that the update has actually traversed the autonomous systems that appear in the AS_PATH attribute. Unfortunately, this solution requires a public key infrastructure in place that assigns public keys to all participating autonomous systems. Because it cannot be expected that S-BGP will be adopted by all ASes simultaneously, it is necessary to be backward compatible with BGP. Hence, during the transition phase, an attacker might send information using the old protocol. In case of plain BGP updates, the level of trust in the included routing information is set by the site policy. The obvious risk is that such policies will often default to accepting the normal BGP update, especially in the beginning of the change-over.

A major drawback of S-BGP and related schemes is the requirement to apply changes to the existing protocol. Such changes not only imply a huge cost as hardware devices need to be upgraded, but there is also a reluctance to switch to designs that are not proven to work effectively on a large scale. Currently, it is not clear whether S-BGP will eventually take hold or how long the necessary transition phase will last. In [9], Goodell et al. highlight the fact that existing BGP security approaches have not been widely deployed. The authors consider the protocols' limited ability to be incrementally deployed, the high computational costs and the infeasibility of modifying the vast installed base of BGP software as the main contributors to the slow rate of adoption. Recognizing these limits, a protocol (ASRAP – autonomous system routing authority protocol) that can be incrementally deployed in parallel to the existing routing infrastructure is proposed. Similar to S-BGP, this protocol allows autonomous systems to verify routing updates. Unlike S-BGP, however, the UPDATE messages themselves are not modified. Instead, each participating AS has to provide an ASRAP service that can be queried by others to verify transmitted routing updates. The authors themselves realize that the success of their solution requires AS administrators to install such services and maintain an additional database, initially without receiving any obvious benefit. Even if such a solution is eventually realized, it would take a considerable amount of time until a majority of ASes support ASRAP. In the meantime, however, there is a need to provide a mechanism that can help routers to decide whether information received in update messages appears suspicious or not. This functionality is provided by our techniques to verify route updates.

3 Threats from BGP Peers

Threats from BGP peers have their origin in the trust a router has to place in the information it receives from its peers. The protocol standard does not include or suggest any mechanism to verify this information – that is, the routing data. Therefore, a malicious or misconfigured router can propagate invalid route advertisements or route withdrawals virtually without restrictions.

The most important information in a routing `UPDATE` message consists of the reachability information in the `NLRI` field and the `AS_PATH` attribute. The `NLRI` field specifies the IP address blocks that are either announced as reachable through a route or that are withdrawn as unreachable at this point in time. The `AS_PATH` attribute enumerates the autonomous systems that have to be traversed to reach the announced address blocks. This is needed to prevent routing loops but can also be used to make routing decisions based on policy or performance metrics. For example, when receiving a route to the same target IP address via multiple routes, the shorter one (as represented by less intermediate entries in the `AS_PATH` attribute) can be chosen.

As neither the reachability information nor the path attribute can be validated by a BGP peer receiving an `UPDATE` message, a malicious router is able to

1. specify an invalid AS path to an IP block so that the path includes the malicious AS itself (i.e., invalid AS path announcement).
2. announce that it controls an IP block that it has no authority over (i.e., IP Address ownership violation).

Such malicious injections can cause traffic to be routed to the malicious AS while legitimate sites become unreachable. This enables the attacker to perform man-in-the-middle attacks or to launch a large-scale denial of service.

Although many ISPs employ filters to discard invalid route updates, these mechanisms do not provide sufficient protection. This is confirmed by the continuous occurrences of incidents [7,15,16,18] where invalid BGP updates are accepted, leading to large scale loss of connectivity. The following two sections describe detection techniques that are capable of identifying updates that are suspicious in the two ways enumerated above.

4 Detection of Invalid AS Path Announcements

An invalid AS path is an `AS_PATH` entry in an `UPDATE` message that announces a potential route to a certain IP address range, although the route does not exist. The AS path specifies the sequence of autonomous systems that a route announcement has previously traversed and describes a potential route to the destination IP addresses in the NLRI field. When a malicious AS crafts an update message with an invalid AS path, it offers a route to the advertised IP destinations that does not exist. Such update messages mislead other ASes, causing them to send traffic to the malicious AS and enabling the aforementioned man-in-the-middle and denial of service attacks.

It is infeasible to determine the validity of an AS path that has not been observed before by solely analyzing single BGP update messages. Consider a malicious AS that advertises a direct route through itself to the address block that it intends to hijack. The update message is crafted such that it appears to originate from the victim AS and an AS that receives such a message cannot tell whether a new, legitimate connection has been established or whether the route is invalid.

4.1 AS Connectivity Graph

The required additional information that enables us to analyze AS_PATH entries is obtained from the topology of the AS connectivity graph. This graph is only based on autonomous systems and the links between them. We do not consider single routers. We observe that each AS, in addition to having authority over a set of IP address blocks, is connected to a set of neighboring autonomous systems. The idea is that these inter-AS connections can be extracted or, to be more precise, sufficiently well approximated from processing UPDATE messages.

The AS connectivity graph is a graph G that consists of a set of n vertices $\{v_1, v_2, \ldots, v_n\}$, each representing a different autonomous system. Each vertex is labeled with a unique identifier that represents the 16-bit autonomous system number that is assigned to organizations by their responsible Internet number registry (e.g., American Registry for Internet Numbers – ARIN [2]). Each node v_i can be connected to zero or more other vertices $\{v_j, \ldots, v_k\}$ by undirected edges $\{e_{ij}, \ldots, e_{ik}\}$. An edge (or link) e_{ij} represents a direct network connection between the autonomous systems represented by v_i and v_j such that routers located in those systems can exchange traffic without having to traverse another autonomous system. Connections between ASes manifest themselves as adjacent AS numbers in the AS_PATH attributes of UPDATE messages. More precisely, they can be retrieved from sequence segments of AS_PATH attributes.

In addition to sequence segments that show the exact trail of a route update, an AS_PATH attribute can also contain AS sets. AS set segments store autonomous systems in an arbitrary fashion and commonly appear in the announcement of aggregated routes. Aggregated routes are utilized to decrease the size of routing tables and are created by an AS that does not forward BGP update messages from its peers immediately. Instead, it collects these messages and merges their address sets into a single block. The AS then announces the resulting block, essentially claiming that it owns the aggregated address space. In most cases, however, there is no single AS path that leads to all the aggregated IP address destinations and the AS_PATH attributes have to be merged as well. This is done by combining all autonomous systems into the unordered collection of AS numbers called AS set. This AS set is then used as new AS_PATH attribute.

The AS set is needed to prevent routing loops. If the sets were omitted and the aggregating router announced a path originating at the local AS, the route might be propagated back to one of the autonomous systems that originally announced parts of the aggregated route. This AS would be unable to determine that it has previously announced parts of that aggregated route itself and might install or forward the update instead of discarding it. Although the omission of the AS sets when propagating aggregated routes is bad practice and might lead to routing loops, it is the default setting in Cisco's BGP implementation. When an AS set is encountered in the AS_PATH attribute, no connectivity information can be retrieved from it.

Several previous studies of the BGP topology [11, 20, 32] have utilized data extracted from BGP routing tables or BGP update messages. The resulting graphs have proven to be useful in determining correspondence between IP addresses, prefixes and ASes. A common classification in this research distinguishes between *core* and *periphery* nodes of the connectivity graph. According to [10], the core consists of international, national and North American regional providers while the periphery corresponds to metropolitan area providers, companies and universities. In [33], the core and the periphery nodes are called *transfer* and *stub* nodes, respectively. The authors state that the connectivity graph is hierarchical with transfer nodes being highly interconnected while stub nodes attach to at most a few other stub nodes and have one or two links to transfer nodes.

Both studies utilize the node degree (i.e., the number of neighbors or links to other nodes) as a distinguishing criteria to classify ASes as either core or periphery systems. Following this observation, we adapted a technique described in [6] to determine the core nodes of the AS graph. The algorithm operates by successively pruning nodes from the graph that have a degree of two or less (i.e., nodes that have at most two connections to the remaining nodes in the graph). The pruning is continued until no more nodes can be removed, and each removed node is classified as periphery. Note that each node of the graph can be evaluated multiple times as the pruning progresses. It is possible that the number of links of a node with a degree greater than two is reduced due to other nodes that are removed from the connectivity graph. When the algorithm terminates, all remaining nodes are classified as core nodes. This process labels between 10% an 15% of all autonomous systems as core nodes, a finding that is in agreement with the two studies mentioned above [10, 33] as well as results reported in [8]. For exact results obtained for a selection of data sets, see Section 6.

Other methods [8, 31] to classify autonomous systems are possible and might improve our detection results. In our work, however, we have chosen the straightforward approach presented in [6] and leave the assessment of alternative classification algorithms for future work.

When the core and periphery nodes of the connectivity graph have been determined, the complete AS connectivity graph can be decomposed into *clusters* of periphery nodes. This decomposition is achieved by removing all core nodes from the graph. The resulting graph is no longer connected – instead it consists of many small groups of interconnected periphery nodes. These groups no longer have paths between them. This can be expected, as the core nodes represent the backbone of the Internet that provides the links between collections of smaller networks. A set (or group) of periphery nodes where each node is reachable from every other node in the set via at least one path is called a cluster. Note that the path between nodes in a cluster may contain intermediate nodes. It is not necessary that each node in a cluster has a direct link to every other node. One exemplary cluster of six ASes located around Toronto in Ontario, Canada is shown in Figure 1. The distances between individual autonomous systems range from 0 kilometers, when two ASes are in the same town (here Toronto), to 238 kilometers. The figure also shows four uplinks that connect ASes to core nodes (such as the link from AS 2767 – AT&T Canada to AS 7018 – AT&T WorldNet).

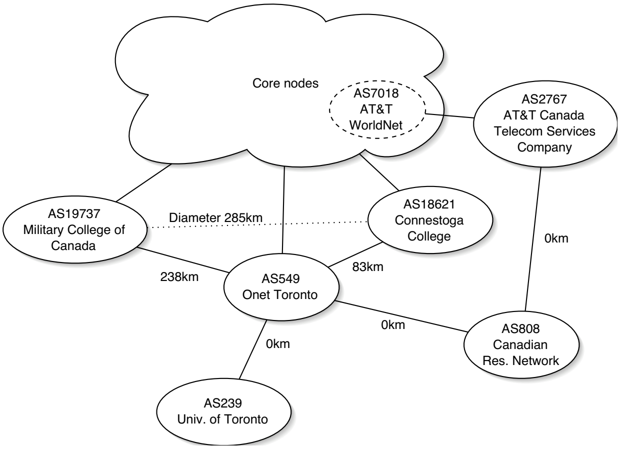


Fig. 1. AS Cluster around Toronto, Canada

We claim that the *geographical distances* between autonomous systems that are represented by the nodes of a single cluster are small. To capture the geographical distances of a cluster more precisely, we define the *cluster diameter* as the maximum geographical distance between any two of its ASes. In Figure 1, the maximum distance is 285 kilometers between AS 18621 and AS 19737. For the calculation of the cluster diameter, it is not necessary that a direct link exists between the most distant ASes. The idea is to confine a geographic area or region in which all ASes are located. The validity of our claim can be intuitively understood by the fact that periphery nodes represent small networks that are connected to large providers to obtain Internet connectivity (represented by core nodes) and to other small networks that are located in close vicinity. It is neither economically nor technically reasonable for a periphery network to install a direct link to another periphery network that is far away (from a geographical point of view). Our hypothesis is verified by deriving the cluster diameters for AS connectivity graphs (containing several thousand ASes) constructed from update messages collected at different points in time over the last two years. For a description of the test methodology and the exact results, consult Section 6.

4.2 Detection Techniques and Limitations

Based on the partition of the AS connectivity graph into core and periphery nodes and the observation that the cluster diameter is small, we define the following two constraints that a valid AS_PATH must satisfy.

1. The sequence of autonomous systems in an AS_PATH may only contain a single subsequence of core ASes. That is, a path that has traversed and left the core may never enter the core again.
2. All consecutive pairs of periphery ASes in an AS_PATH must either be part of the same cluster or, when they establish a link between two previously unconnected clusters, must be in close geographical vicinity.

The first constraint ensures that valid routes between two periphery ASes only traverse the core once and do not detour through a different periphery system before reaching the destination. As the core represents the backbone of the Internet, it is capable of delivering packets between any two periphery clusters directly. This constraint is also discussed in [33] and [8]. Both authors observe that valid paths traverse the core only once and do not have any intermediate periphery nodes.

The second constraint refers to direct connections between periphery systems. As shown in Section 6, periphery ASes that are directly connected are in a close geographical vicinity. When two periphery ASes are consecutively listed in an AS path, a direct link between these two is indicated. When a link between the two ASes already exists or when both belong to the same cluster, the connection is considered legitimate. When the link connects two previously unconnected clusters, the geographical distance between the two ASes has to be calculated. To be a valid update message, this distance has to be below a certain, adjustable threshold that can also depend on the diameter of the two clusters involved. For our experiments, this threshold is set to the maximum between the sum of the two cluster diameters and 300 kilometers[1].

The two aforementioned constraints allow us to validate certain properties of the AS paths in BGP update messages. For example, a malicious periphery AS that attempts to craft an invalid path to a victim usually cannot simply announce a direct route to the victim's AS. This is because such a direct link would violate the second constraint (assuming that the malicious AS and the victim are far enough away). In case the malicious nodes attempts to evade detection and

[1] 300 kilometers was chosen as a reasonable low value to capture the notion of close proximity. The threshold was selected prior to the evaluation and was not tuned to improve the experimental results afterwards.

inserts a core AS between itself and the victim's AS, the advertisement of this new route to any core AS would result in a sequence of core nodes that is interrupted by the offending AS. Such an update message would then violate the first constraint.

An obvious restriction of the topology-based approach is that only connections between periphery ASes can be validated using the geographical distance measurement. When a core node installs a new, direct route to another node (which may be either periphery or core), there is no reason why this announcement should be distrusted. It is perfectly reasonable to conclude that simply another direct link has been established. This limitation, however, only affects updates sent by large providers. Since these organizations usually employ network monitoring and implement high security standards, the threat that emanates from them is small compared to local providers or companies. Also note that this limitation does not affect updates that providers receive from their peers. They can still be checked and potential problems detected.

Another limitation prevents the detection of invalid updates when an AS claims that is has a direct connection to another autonomous system that is in a close geographical vicinity. In this case, the distance between the nodes representing the attacker and the victim AS is short and the model assumes that a valid, direct route has been installed. This allows a malicious AS to affect routing to other ASes that are located nearby. However, only a limited number of periphery ASes are located close to any specific autonomous system. This puts a limit on the number of potential targets and the freedom that the attacker has in choosing the victim. When an attacker attempts to forge a route to a distant AS, our system is capable of detecting the invalid path update.

The problem of dynamically updating the network model is left for future work. In our current design, it is necessary to rebuild the network model when the underlying topology changes in an extent that causes a significant raise in the number of false alarms. Note, however, that the network topology model can be built very fast. The model creation process required, for our experiments, update messages collected in a period of less than a day before it converged. Convergences was reached when new update messages did not result in any new information inserted into the graph for more than six hours. The detection process also exhibits a certain robustness against invalid updates during the model creation phase. Although invalid information is entered into the topology graph, the defect remains confined locally.

5 Detection of IP Address Ownership Violation

An IP address ownership violation occurs when an AS announces an IP block that it is not entitled to. This announcement is done by setting the `NLRI` field of the update message to the illegitimate IP range and transmitting it using an `AS_PATH` that starts with the malicious AS. An AS that receives such a message considers the originating malicious AS authoritative for the announced IP block and will forward corresponding packets there (given that it has not received a more preferable path to that IP block from the legitimate owner).

This problem, also called Multiple Origin AS (MOAS) conflict, was extensively studied by Zhao et al. [34]. The authors point out that MOAS conflicts occur relatively frequently, and also provide several non-malicious explanations for this phenomenon.

One possibility to distinguish between malicious and legitimate MOAS conflicts are BGP protocol enhancements, either using cryptographic solutions such as S-BGP [14] or protocol extensions such as MOAS lists [35]. A MOAS list contains a list of all ASes that are entitled to announce a certain IP block, and is attached to BGP route announcements. Although individual MOAS lists can be altered or forged, the solution relies on the rich connectivity of the Internet. It is assumed that a router will, in addition to a malicious MOAS list sent by an attacker, also receive a valid MOAS list from a legitimate source, thus being able to detect the inconsistency and raise an alarm.

In contrast to that, we pursue a more naïve strategy and attempt to prevent as many MOAS conflicts as possible that originate from probably legitimate configurations. This is done by ignoring BGP updates with aggregated `NLRI` fields or set `COMMUNITIES` attributes, as described in more detail below. Also, updates that announce large IP ranges (in our case, network masks with 8 or less bits) are excluded from our model. This approach aims to reduce the number of false positives, with the downside of an increased false negative rate. Future work will investigate improvements of this technique.

In our current approach, we build a model that stores a mapping between IP address blocks and their respective, authoritative ASes to detect address ownership violations. This mapping is constructed from BGP update messages during the model building phase. In the simplest case, the IP address block that a particular AS owns can be extracted directly from update messages. An IP range is announced by its owner by creating a suitable BGP `UPDATE` message and sending it to the peering partners of the particular autonomous system. As each AS forwards such updates, it is required to prepend its own number to the already existing `AS_PATH` attribute. Thus the originating AS appears as the last entry in the path list. Whenever our system observes a BGP message announcing an address block, the mapping between the IP range and its owner is inserted into our model.

It is not a requirement that an AS actually owns an IP block to be entitled to announce it. In fact, it is possible (and quite common) that an autonomous system would be granted the right to announce a block for a related AS. All IP packets that are forwarded to that AS are then correctly relayed to the actual target. In such a case, however, the actual owner is not supposed to announce the address block itself. For an external observer, it appears as if the address block is owned by the AS that announces it.

Unfortunately, there are situations when the owner of an IP block cannot be identified easily. The most common reason is the aggregation of IP address ranges. As previously stated, when an AS performs aggregation it claims that it is the origin of the aggregated address space, effectively masking the true owners of the aggregated IP subranges. An autonomous system that performs this step is required to tag this announcement with a special flag. This enables an external observer to identify aggregated update messages as such. A mechanism similar to aggregation is used with communities. The `COMMUNITIES` attribute was introduced in RFC 1997 [3] and is used to describe a group of autonomous systems with common properties. It is used to control the propagation of routing information based on communities instead of IP prefixes and AS paths alone in an attempt to simplify routing policies. When routes from different ASes that belong to the same community are aggregated, the aggregation tag is not set. Nevertheless, the original source of an update can no longer be determined with certainty.

The straightforward solution to the problem of aggregated routes[2] is to simply exclude them from the analysis. Unfortunately, a malicious AS could then evade detection by marking a route update as aggregated. Therefore, update messages that announce aggregated routes cannot be discarded immediately. Instead, we only discard these updates when the originating AS is a core node. In this case, it is very probable that the `NLRI` field contains IP ranges of many different destinations and the information cannot be reliably utilized. In the case of a periphery node, however, a mapping between the aggregated IP block and the corresponding AS is installed. When a periphery AS aggregates routes, we assume that the aggregated IP blocks are unlikely to be announced independently by the actual owner (that is a periphery AS as well). This assumption is confirmed by the low false alert rate that our system produces (as shown in Section 6).

The knowledge of IP address ownership helps to detect attacks or misconfigurations where an AS announces an address block that is not under its authority. `UPDATE` messages that contain addresses in their `NLRI` field that are already owned by someone else are classified as malicious. For similar reasons as outlined above, we discard all aggregated routes that originate at core nodes.

In general, the ownership of an address is relatively stable. Although flapping connections or broken links may cause a specific route to certain target addresses to be withdrawn, we cannot delete the address binding from our database as the ownership of the respective block has not changed. The problem of changes in the ownership of IP blocks can be solved in two ways. One approach involves a human operator that notices the increase of alleged attacks caused by clashing IP blocks and removes the old binding after making sure that the alerts are incorrect. Then, the new owner of the now vacant address can be entered into the model and normal operation continues. A more sophisticated automated mechanism determines whether the previous owner has recently announced the disputed IP blocks. When a sufficient amount of time has elapsed since the last announcement, the new owner is considered to be legitimate and ownership is transferred.

[2] In the following discussion, the term aggregated routes applies to update messages with community attributes as well.

6 Experiments

We have developed several criteria that help to assess the validity of routing data using an underlying model of the global routing infrastructure. Our model, consisting of the mapping of IP prefixes to ASes and the AS connectivity graph, is built by processing routing updates collected at *Looking Glass* sites such as the one run by the University of Oregon [32]. Looking Glass sites are passive BGP peers that maintain connections to a number of major BGP routers throughout the Internet and obtain the routers' forwarding tables as well as any `UPDATE` messages the routers receive. This allows one to get BGP data from multiple vantage points in different locations. The data is archived and made publicly available.

The techniques described in the previous sections have been implemented to detect potentially invalid route messages sent by BGP peers. Note that the detection system does not have to be installed at the actual BGP routers. Instead, in a setup that is similar to the one used by Looking Glass sites, `UPDATE` messages can be forwarded by routers to a regular desktop machine where the analysis can be performed.

The empirical evaluation of our approach uses BGP data collected during four different weeks over the last two years. The first data set contains BGP update messages collected during the week starting from April 5th, 2001, the second set starting from January 10th, 2002, the third set starting from September 15th, 2002, and the fourth starting from March 3rd, 2003. The first day of each week was used to build the IP address to AS mapping and the AS connectivity graph. The subsequent six days were then used as an input for the detection process. We assume that the day utilized for the model creation phase is free of any major incidents. However, minor misconfigurations are likely to occur. This results in a slightly imprecise topology graph, and thereby, might result in incorrect detections. We claim that the effect of these misconfigurations is small; a claim that is supported by the evaluation of the quality of the model and the detection process in the following two sections.

6.1 Model Validation

Our detection mechanisms depend upon both reliable classification of *core* and *periphery* ASes, as well as the validity of the assumption that ASes making up each cluster in the periphery are geographically close. Prior to investigating the detection performance of the system, this section explores these requirements in more detail.

Table 1 provides statistical data for each AS connectivity graph constructed from the BGP update messages of the first day of the respective four data sets. The iterative algorithm for partitioning the AS connectivity graph into core and periphery nodes (described in Section 4.1) performs well. Upon removing the core, the remaining nodes in the graph fall into disjoint clusters. The total number of core AS nodes represent, on average, 12.6% of the total number of

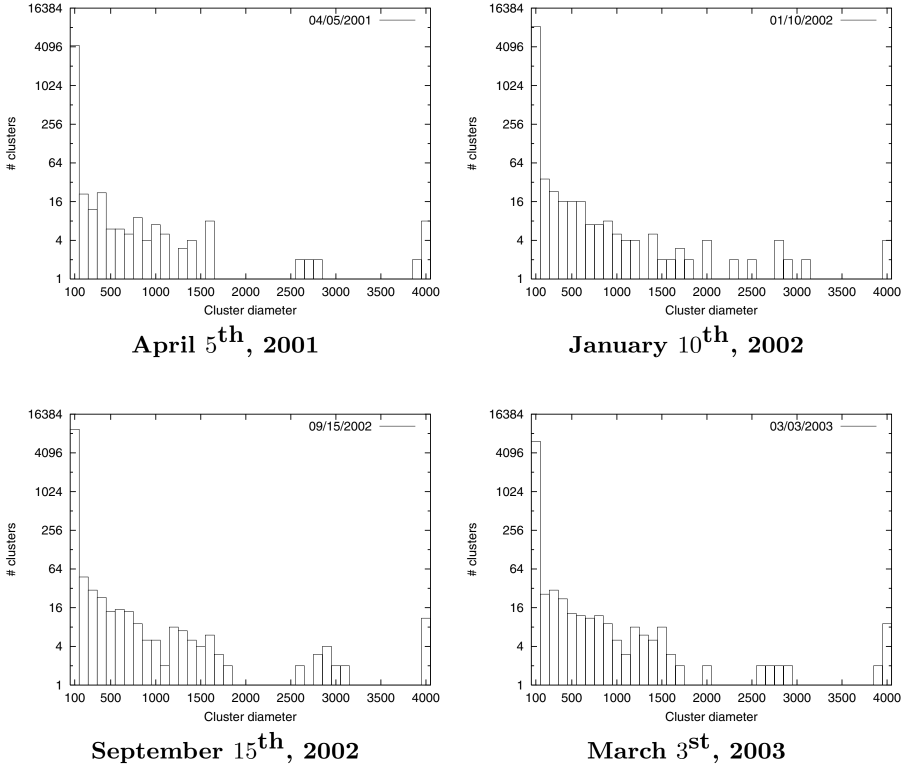
Table 1. AS Connectivity Graph Statistics

Date	Periphery	Core	Clusters	Max. Size	Avg. Size
Apr. 5th, 2001	5831 (89.5%)	686 (10.5%)	4437	64	1.31
Jan. 10th, 2002	10592 (85.1%)	1860 (14.9%)	8692	72	1.20
Sep. 15th, 2002	12006 (87.1%)	1773 (12.9%)	9762	63	1.23
Mar. 3rd, 2003	8422 (87.9%)	1162 (12.1%)	6418	68	1.31

nodes in the graph. This is in close agreement with [10] and [33], which find about 10% of ASes that constitute the core of the Internet. The number of nodes in each cluster is small, usually one, but there are also large clusters with a few tens of nodes. Table 1 shows, for each data set, the number of clusters (*Clusters*) as well as the maximum (*Max. Size*) and average number of nodes per cluster (*Avg. Size*).

For each cluster, we calculate the cluster diameter as defined in Section 4.1. This requires determining the maximum geographical distance between any two of its ASes. To obtain the distance between two autonomous systems, it is necessary to determine the locations for both ASes and to calculate the great circle geographic distance between them. The location for an AS is extracted from the `whois` database entry of the appropriate local registry (ARIN [2] for the US and Canada, RIPE for Europe, LACNIC [17] for Latin America and APNIC [1] for Asia and the Pacific). The `whois` entries in the ARIN database list the city, state and country for the autonomous system location in explicitly marked fields. This makes it straightforward to extract the required data. The other three databases, however, do not follow a standardized method of specifying locations. Therefore, we have developed a parser that retrieves the provided organizational description and contact information for each AS and attempts to determine a probable geographical position. Manual inspection of a few hundred locations indicate that the extraction of geographical data is successful. Additionally, our results show that connected periphery ASes are in close proximity (see Figure 2 below). Note, however, that the location information is only useful for periphery nodes. Although core nodes have a specific geographic location as well, their corresponding networks usually span a large geographical area and, thus, this information has less value. Only for peripheral ASes, the location information is meaningful.

Figure 2 is a log-scale histogram plot that shows the distribution of cluster diameters for the four datasets considered in this evaluation. In all cases, the fraction of clusters whose diameter is greater than 300 kilometers is less than 2.4%. There is a relatively small number of high-diameter (i.e., $\geq$ 4000 km) outliers in each plot. These are due to obviously incorrect or stale entries in the `whois` database or are caused by special purpose links operated by ASes that are not classified as core nodes. For example, NASA operates a branch office in Moscow with its own AS number and this AS has a direct connection to a location in the US. However, the special links are expected to be stable and the



April 5th, 2001

January 10th, 2002

September 15th, 2002

March 3st, 2003

Fig. 2. Cluster Diameter Distribution (logarithmic scale)

installation of such a connections is a relatively infrequent event. Therefore, we do not expect a noticeable influence on the number of false alerts; an assessment that was confirmed by our measurements shown in Section 6.2. Note that we did not manually change any of these 'anomalies' for the evaluation of the detection process, but we expect them to contribute to the observed false alarms.

6.2 Detection Evaluation

The detection approach was evaluated on BGP data collected during four different weeks over the last two years. We used the first day of each week to build our models and the subsequent six days as an input to the detection algorithms.

The first two data sets are important to assess the detection capability of our system as both hold traces of significant misconfiguration problems. The first data set (that starts on April 5th, 2001) contains an incident where Cable and Wireless (AS3561) propagated more than 5000 improper route announcements from one of its downstream customers [7, 18] that illegitimately claimed large IP ranges. This lead to global connectivity problems for about 90 minutes. Clearly, the corresponding messages should be identified as IP address ownership

Table 2. Alert Overview

Week starting at	Update Messages	Address Violation		Invalid AS Path	
		Correct	Incorrect	Correct	Incorrect
Apr. 5th, 2001	1,507,673	2148	18	0	0
Jan. 10th, 2002	5,918,085	0	23	76	0
Sep. 15th, 2002	7,065,598	0	23	0	0
Mar. 3rd, 2003	5,499,401	0	14	0	0

violations. The second data set (that begins on January 10th, 2002) contains evidence of a misconfiguration where a customer propagated routes received from its upstream provider to a peering partner [18]. This peer then continued to forward the routes to its own upstream provider, thereby advertising routes to parts of the Internet through the customer AS itself. This misconfiguration is similar to an attack where a periphery AS announces a route to other periphery ASes through itself and the involved updates should be classified by our system as invalid AS paths.

The third and fourth data set are, after reviewing the mailing list of the North American Network Operators' Group [25] for the periods in question, free of major attacks or misconfigurations. These two weeks serve as more recent data to verify that our assumptions about the AS connectivity graph are valid and to provide an estimate for the false positive rate for the Internet at present.

Table 2 provides the results of our detection system. For each data set (collected over six days), the table shows the total number of processed update messages, the number of IP address ownership violations and the number of invalid AS paths that were reported. All alerts are classified as either correct or incorrect. An alert is classified as correct when it is obviously related to one of the two incidents in the first two data sets as described above. All other alerts are classified as incorrect. Closer examination of the incorrect alerts reveals that a large percentage is due to the misclassification of periphery nodes that are in fact part of the core. Such misclassifications occur mainly for autonomous systems located in Asia or Eastern Europe. The update messages collected from Route View mostly contain traffic sent between ASes in Europe and the US, resulting in an AS connectivity graph that is sparse for other regions. As the node classification relies on the degree of connectivity, core nodes in those regions may not have enough neighbors in our graph and are thus labeled as periphery. To obtain more precise data from these regions, we would require BGP data from routers located in there. Other incorrect alerts might have been the result of actual misconfigurations, but no supporting evidence was found for the relevant dates on the network operator mailing list [25] and the alerts had to be classified as incorrect. Another possible cause are invalid data utilized to create the network topology model.

Note that the numbers in Table 2 reflect unique violations. That is, when multiple invalid update messages with identical routing information are observed, only one alert is produced. This shows the potential tremendous impact of a single misconfiguration on the global infrastructure. For example, the 2148 different address ownership violations detected during one day of the first test week were the result of a single incident.

It is interesting to observe that the properties of the network graph as well as the behavior of the system do not change noticeably for the test sets that reflect samples from a period of over two years. This indicates that our assumptions are stable and that detection can be performed reliably.

7 Future Work

This section gives an overview of our plans to extend the security model as well as our presented technique.

A useful extension of the proposed approach to BGP security is the inclusion of BGP policies. BGP speakers usually define policies that restrict the information exchange with their respective neighbors and influence routing decisions. This allows us to determine whether the current network graph conforms to the specified policies and raise an alarm in case a deviation is detected. Such deviations could either result from misconfigurations or malicious behavior.

The presented design does not automatically take into account changes of IP address ownership and the removal of connections between autonomous systems. It would be desirable to determine when IP address blocks have been transferred between ASes without the intervention of an operator that has to remove the binding from the database. This could be done by including information from the Route Arbiter Project [29] or from miscellaneous network information centers. Also, the information in withdrawal messages is not utilized. This is because it is not straightforward to extract topology information from such updates.

8 Conclusion

The Border Gateway Protocol is the de facto standard for inter-domain routing in today's Internet. Although protocol design weaknesses and implementation flaws in many devices running BGP are well-known, it is difficult to overcome them. The huge base of installed equipment and the fact that, despite several successful attacks, global routing seems to work satisfactorily create an enormous reluctance to the adoption of newer protocols. Although approaches such as S-BGP seem appealing at first glance, they have not been widely deployed. In the meantime, the concept of "security by obscurity" is the only protection against potentially devastating attacks.

We have developed a technique to validate routing data in BGP `UPDATE` messages to protect routers from installing falsified routes. The mechanism is based on topology information of the autonomous systems connectivity graph and geographical data from `whois` databases. It is capable of identifying updates where a malicious or misconfigured router announces illegitimate IP address blocks or invalid routes that do not exist. Our system can be applied immediately and does not interfere with the existing infrastructure.

Acknowledgments

This research was supported by the Army Research Office, under agreement DAAD19-01-1-0484. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Army Research Office, or the U.S. Government.

References

1. Asia Pacific Network Information Centre. `http://http://www.apnic.net`.
2. American Registry for Internet Numbers. `http://http://www.arin.net`.
3. R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. IETF-RFC 1997, Aug 1996.
4. S. Cheung. An Efficient Message Authentication Scheme for Link State Routing. In *13th Annual Computer Security Applications Conference*, December 1997.
5. S. Convey, D. Cook, and M. Franz. An Attack Tree for the Border Gateway Protocol. IETF Internet Draft, Oct 2002.
6. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proceedings of ACM SIGCOMM*, 1999.
7. J. Farrar. Cable and Wireless Routing Instability. `http://www.merit.edu/mail.archives/nanog/2001-04/msg00209.html`.
8. L. Gao. On Inferring Autonomous System Relationships in the Internet. In *Proceedings of IEEE Global Internet*, November 2000.
9. Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick McDaniel, and Aviel Rubin. Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Network and Distributed Systems Security*, 2003.
10. R. Govindan and A. Reddy. An Analysis of Internet Inter-Domain Topology and Route Stability. In *IEEE InfoCom*, 1997.
11. B. Huffaker, A. Broido, k. claffy, M. Fomenkov, K. Keys, E. Lagache, and D. Moore. Skitter AS Internet Graph. CAIDA, Oct 2000.
12. Y.F. Jou, F. Gong, C. Sargor, X. Wu, F. Wu, H.C. Chang, and F. Wang. Design and Implementation of a Scalable Intrusion Detection System for the Protection of Network Infrastructure. In *DARPA Information Survivability Conference and Exposition*, January 2000.
13. S. Kent, C. Lynn, J. Mikkelson, and K. Seo. Secure Border Gateway Protocol (Secure-BGP) - Real World Performance and Deployment Issues. In *Proceedings of the Symposium on Network and Distributed System Security*, February 2000.
14. S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, April 2000.
15. C. Labovitz, A. Ahuja, and F. Jahanian. Experimental Study of Internet Stability and Wide-Area Network Failures. In *Fault-Tolerant Computing Symposium*, June 1999.
16. C. Labovitz, G. R. Malan, and F. Jahanian. Origins of Internet Routing Instability. In *IEEE INFOCOM*, March 1998.

17. The Latin American and Caribbean Internet Addresses Registry. `http://http://www.lacnic.net`.
18. R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfiguration. In *Proceedings of ACM SIGCOMM*, August 2002.
19. G. Malkin. RIP Version 2. IETF-RFC 2453, Nov 1998.
20. Sean McCreary and Bill Woodcook. PCH RouteViews archive. `http://www.pch.net/resources/data/routing-tables`.
21. V. Mittal and G. Vigna. Sensor-Based Intrusion Detection for Intra-Domain Distance-Vector Routing. In *Proceedings of the ACM Conference on Computer and Communication Security (CCS'02)*, Washington, DC, November 2002. ACM Press.
22. J. Moy. OSPF Version 2. IETF-RFC 2328, Apr 1998.
23. Sandra Murphy. Border Gateway Protocol Security Analysis. IETF Internet Draft, Nov 2001.
24. S.L. Murphy and M.R. Badger. Digital Signature Protection of the OSPF Routing Protocol. In *Proceedings of the Symposium on Network and Distributed System Security*, February 1996.
25. The North American Network Operators' Group. `http://www.nanog.org`.
26. D. Qu, B.M. Vetter, F. Wang, R. Narayan, F. Wu, F. Jou, F. Gong, and C. Sargor. Statistical Anomaly Detection for Link-State Routing Protocols. In *In Proceedings of the 1998 International Conference on Network Protocols*, October 1998.
27. A. Przygienda R. Hauser and G. Tsudik. Reducing the cost of security in link state routing. In *ISOC Symposium on Network and Distributed System Security*, February 1997.
28. Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). IETF-RFC 1654, Mar 1995.
29. Routing Arbiter Project. `http://www.ra.net`.
30. B.R. Smith, S. Murthy, and J.J. Garcia-Luna-Aceves. Securing Distance-Vector Routing Protocols. In *Proceedings of the Symposium on Network and Distributed System Security*, February 1997.
31. L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy From Multiple Vantage Points. In *IEEE INFOCOM*, 2002.
32. University of Oregon - Looking Glass. `http://antc.uoregon.edu/route-views`.
33. E. Zegura, K. Calvert, and M. Donahoo. A quantitative comparison of graph-based models for internetworks. *IEEE/ACM Transactions on Networking*, 5(6):770–783, December 1997.
34. X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflict. In *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, USA, November 2001.
35. X. Zhao, D. Pei, L. Wang, L. Zhang, D. Massey, A. Mankin, and S. F. Wu. Detection of Invalid Route Announcement in the Internet. In *International Conference on Dependable Systems and Networks*, 2002.