International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 1, Issue 2, June 2014, PP 1-7
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Defending IP Spoofing through Inter Domain Packet Filter on BGP Updates

Ms Yogita Deshmukh

Dept. of M-TECH CSE
ABHA Gaiyakwad Patil College of Engineering
Nagpur
yogita.4849@gmail.com

Abstract: *The Distributed Denial-of-Service (DDoS) attack is a serious threat to the legitimate use of the Internet. Prevention mechanisms are thwarted by the ability of attackers to forge or spoof the source addresses in IP packets. By employing IP spoofing, attackers can evade detection and put a substantial burden on the destination network for policing attack packets. In this paper, we propose an inter domain packet filter (IDPF) architecture that can mitigate the level of IP spoofing on the Internet. A key feature of our scheme is that it does not require global routing information. IDPFs are constructed from the information implicit in Border Gateway Protocol (BGP) route updates and are deployed in network border routers. We establish the conditions under which the IDPF framework correctly works in that it does not discard packets with valid source addresses. Based on extensive simulation studies, we show that, even with partial deployment on the Internet, IDPFs can proactively limit the spoofing capability of attackers. In addition, they can help localize the origin of an attack packet to a small number of candidate networks.*

Keywords: *IP spoofing, DDOS, BGP, network-level security and protection, routing protocols*

1. INTRODUCTION

DISTRIBUTED Denial-of-Service (DDoS) attacks pose an increasingly grave threat to the Internet, as evident in recent DDoS attacks mounted on both popular Internet sites and the Internet infrastructure [1]. Alarmingly, DDoS attacks are observed on a daily basis on most of the large backbone networks [2]. One of the factors that complicate the mechanisms for policing such attacks is IP spoofing, which is the act of forging the source addresses in IP packets. By masquerading as a different host, an attacker can hide its true identity and location, rendering source based packet filtering less effective. It has been shown that a large part of the Internet is vulnerable to IP spoofing. Although attackers can insert arbitrary source addresses into IP packets, they cannot control the actual paths that the packets take to the destination. Based on this observation, Park and Lee [12] proposed the route-based packet filters as a way of mitigating IP spoofing. The idea is that by assuming single-path routing, there is exactly one single path p(s, d) between the source node s and the destination node d. Hence, any packet with the source address s and the destination address d that appear in a router that is not in p(s, d) should be discarded. The challenge is that constructing such a route based packet filter requires the knowledge of global routing information, which is hard to reconcile in the current Internet routing infrastructure [13]. Inspired by the route-based packet filters [12], we propose an inter domain packet filter (IDPF) architecture, a route based packet filter system that can be constructed solely based on the locally exchanged BGP updates, assuming that all ASs employ a set of routing policies that are commonly used today The key contributions of this paper are given as follows. First we describe how we can practically construct IDPFs at an AS by only using the information in the locally exchanged BGP updates. Second, we establish the conditions under which the proposed IDPF framework works correctly in that it does not discard packets with valid source addresses. Third, to evaluate the effectiveness of the proposed architecture, we conduct extensive simulation studies based on AS topologies and AS paths extracted from real BGP data. The results show that, even with partial deployment, the architecture can proactively limit an attacker's ability to spoof packets. When a spoofed packet cannot be stopped, IDPFs can help localize the attacker to a small

number of candidates ASs, which can significantly improve the IP traceback situation [7]. In addition, IDPF-enabled Ass (and their customers) provide better protection against IP spoofing attacks than the ones that do not support IDPFs. This should give network administrators incentives to deploy IDPFs. The idea of IDPF is motivated by the work carried out by Park and Lee [12], who evaluated the relationship between network topology and the effectiveness of route-based packet filtering. They showed that packet filters constructed based on the global routing information can significantly limit IP spoofing when deployed in just a small number of ASs. In this work, we extend the idea and demonstrate that filters that are built based on local BGP updates can also be effective. Unicast reverse path forwarding (uRPF) [17] requires that a packet is forwarded only when the interface that the packet arrives on is exactly the same used by the router to reach the source IP of the packet. If the interface does not match, the packet is dropped. Although this is simple, the scheme is limited, given that Internet routing is inherently asymmetric, that is, the forward and reverse paths between a pair of hosts are often quite different. The uRPF loose mode [18] overcomes this limitation by removing the match requirement on the specific incoming interface for the source IP address. A packet is forwarded, as long as the source IP address is in the forwarding table. However, the loose mode is less effective in detecting spoofed packets. In Hop-Count Filtering (HCF) [19], each end system maintains a mapping between IP address aggregates and valid hop counts from the origin to the end system. Packets that arrive with a different hop count are suspicious and are therefore discarded or marked for further processing. In Path Identification [20], each packet along a path is marked by a unique Path Identifier (Pi) of the path. Victim nodes can filter packets based on the Pi carried in the packet header. Stack Pi [21] improved the incremental deployment property of Pi by proposing two new packet marking schemes. In [22], Li et al. described SAVE, which is a new protocol for networks to propagate valid network prefixes along the same paths that data packets will follow. Routers along the paths can thus construct the appropriate filters by using the prefix and path information. Bremler-Barrand Levy proposed a spoofing prevention method (SPM) [23], where packets that were exchanged between members of the SPM scheme carry an authentication key that is associated with the source and destination AS domains. Packets arriving at a destination domain with an invalid authentication key(with respect to the source domain) are spoofed packets and are discarded. In the Packet Passport System [24], a packet that originated in a participating domain carries a passport that is computed based on secret keys shared by the source domain and the transit domains from the source to the destination. Packets carrying an invalid passport are discarded by the transit domains.

2. BORDER GATEWAY PROTOCOL AND INTERCONNECTIONS

In this section, we briefly describe a few key aspects of BGP that are relevant to this paper for a comprehensive description). We model the AS graph of the Internet as an undirected graph G = (V, E). Each node v 2 V corresponds to an AS, and each edge e (u, v) 2 E represents a BGP session between two neighboring ASs u, v 2 V. To ease the exposition, we assume that there is at most one edge between a pair of neighboring ASs. Each node owns one or multiple network prefixes. Nodes exchange BGP route updates, which may be announcements or withdrawals, to learn of changes in reachability to destination network prefixes. A route announcement contains a list of route attributes associated with the destination network prefix. Of particular interest to us are the path vector attribute as path, which is the sequence of ASs that this route has been propagated over, and the local pref attribute that describes the degree of local preference associated with the route. We will use r.as path, r local pref, and r.prefix to denote the as path, the local pref, and the destination network prefix of r, respectively. Let r.as path = hvkvk+1 . . . v1v0i. The route was originated (first announced) by node v0, which owns the network prefix r.prefix. Before arriving at node v_k, the route was carried over nodes v1, v2vk+1 in that order. For i = k, k + 1, we say that edge e(vi,vi-1)is on the AS path, that is, e(vi,vi-1)2 r.as path. When there is no confusion, route r and its AS path r.as path are interchangeably used. For convenience, we also consider a specific destination AS d. All route announcements and withdrawals are specific to the network prefixes owned by d. For simplicity, notation d is also used to denote the network prefixes owned by the AS d. As a consequence, a route r that can be used to reach the destination d.

2.1. Policies and Route Selection

Each node only selects and propagates to neighbors a single best route to the destination, if any. Both the selection and the propagation of best routes are governed by locally defined routing policies. Two distinct sets of routing policies are typically employed by a node. Import policies and export policies. Neighbor-specific import policies are applied upon routes learned from neighbors, whereas neighbor-specific export policies are imposed on locally selected best routes before they are propagated to the neighbors. In general, import policies can affect the "desirability" of routes by modifying route attributes. Let r be a route (to destination d) received at v from node u. We denote by import (v u)[{r}+ the possibly modified route that has been transformed by the import policies. The transformed routes are stored in v's routing table (1) Here, N(v) is the set of v's neighbors. Among the set of candidate routes candidate R (v, d), node v selects a single best route to reach the destination based on a well-defined procedure. To aid in description, we shall denote the outcome of the selection procedure at node v, that is, the best route, as best R(v, d), which reads the best route to destination d at node v. Having selected best R (v, d) from candidate R(v, d), v then exports the route to its neighbors after applying neighbor-specific export policies. The export policies determine if a route should be forwarded to the neighbor, and if so, they modify the route attributes according to the policies (see Section 3.2). We denote by export(v -> u)[{r}+ the route sent to neighbor u by node v after node v applies the export policies on route r. BGP is an incremental protocol. updates are generated only in response to network events. In the absence of any event, no route updates are triggered or exchanged between neighbors, and we say that the routing system is in a stable state. Formally,

Definition 1 (stable routing state): A routing system is in a stable state if all the nodes have selected a best route to reach other nodes and no route updates are generated (or propagated).

2.2. AS Relationships and Routing Policies

The specific routing policies that an AS internally employs is largely determined by .connections between ASs follow a few commercial relations. A pair of ASs can enter into one of the following arrangements.

1. Provider to customer. In this arrangement, a customer AS pays the provider AS to carry its traffic. It is most common when the provider is much larger in size than the customer.

2. Peer to peer. In a mutual peering agreement, the Ass decide to carry traffic from each other (and their customers). Mutual peers do not carry transit traffic for each other.

3. Sibling to sibling. In this arrangement, two ASs provide mutual transit service to each other. Each sibling AS can be regarded as the provider of the other AS. An AS's relationship with a neighbor largely determines the neighbor-specific import and export routing policies. In this paper, we assume that each AS sets its import routing policies and export routing policies according to the rules specified in Tables 1 [15] and 2 [14], [16], respectively. These rules are commonly used by ASs on the current Internet. In Table 1, r1 and r2 denote the routes (to destination d) received by node v from neighbors u1 and u2, respectively. Customer (v), peer (v) provider (v), and sibling(v) denote the set of customers, peers, providers, and siblings of node v, respectively. The import routing policies in Table 1 state that an AS will prefer the routes learned from customers or siblings over the routes learned from peers or providers.

In Table 2, the columns marked with r1-r4 specify the export policies employed by an AS to announce routes to providers, customers, peers, and siblings, respectively. For instance, export rule r1 instructs that an AS will announce routes to its own networks, and routes learned from customers and siblings to a provider, but it will not announce routes learned from other providers and peers to the provider. The net effect of these rules is that they limit the possible paths between each pair of ASs. Combined together, the import and export policies also ensure the propagation of valid routes on the Internet. For example, combining the import and export policies, we can guarantee that a provider will propagate a route to a customer to other ASs (customers, providers, peers, and siblings). If an AS does not follow the import policies, for example, it may prefer an indirect route via a peer instead of a direct route to a customer. In this case, based on export rule

r3, the AS will not propagate the route (via a peer) to a customer to a peer, since the best route (to the customer) is learned from a peer. This property is critical to the construction and correctness of IDPFs (see Sections 4.2 and 4.3). The routing policies in Tables 1 and 2 are incomplete. In some cases, ASs may apply less restrictive policies. For the moment, we assume that all ASs follow the import and export routing policies specified in Tables 1 and 2 and that each AS accepts legitimate routes exported by neighbors. More general cases will be discussed at the end of the next section. If AS b is a provider of AS a and AS c is a provider of AS b, then we call c an indirect provider of a, and indirect customer of c.

Table1. *Import Routing Policies at an AS*

if $((u_1 \in customer(v) \cup sibling(v))$ and $(u_2 \in peer(v) \cup provider(v)))$ then $r_1.\text{local\_pref} > r_2.\text{local\_pref}$

Export rules		r1	r2	r3	r4
Export routes to		provider	customer	peer	sibling
Learned from	provider	no	yes	no	yes
	customer	yes	yes	yes	yes
	peer	no	yes	no	yes
	sibling	yes	yes	yes	yes
Own routes		yes	yes	yes	yes

Indirect siblings are defined in a similar fashion. The import and export routing policies in Tables 1 and 2 imply that an AS will distribute the routes to direct or indirect customers/siblings to its peers and providers. If e(u, v)€best R(s, d).as path, we say that u is the best upstream neighbor of node v for traffic from node s to destination d, and we denote u as u = best U(s, d, v). For ease of exposition, we augment the AS graph with the relationships between neighboring ASs. We refer to an edge from a provider to a customer AS as a provider-to-customer edge, an edge from a customer to provider as a customer-to-provider edge, and an edge connecting sibling (peering) ASs as sibling to- sibling (peer-to-peer) edge .A downhill path is a sequence of edges that are either provider-to-customer or sibling-to-sibling edges, and an uphill path is a sequence of edges that are either customer-to-provider or sibling-to-sibling edges. Gao [14] established the following about the candidate routes in a BGP routing table.

Theorem 1: If all ASs set their export policies according to r1-r4, a candidate route in a BGP routing table

Can be any of the following.

- An uphill path,
- A downhill path,
- An uphill path followed by a downhill path,
- An uphill path followed by a peer-to-peer edge,
- A peer-to-peer edge followed by a downhill path, or
- An uphill path followed by a peer-to-peer edge, which is followed by a downhill path.

3. INTER DOMAIN PACKET FILTERS

In this section, we discuss the intuition behind the IDPF architecture, describe how IDPFs are constructed using BGP route updates, and establish the correctness of IDPFs. After that, we discuss the case where ASs have routing policies that are less restrictive than the ones in Tables 1 and 2. We shall assume that the routing system is in the stable routing state in this section. We will discuss how IDPFs fare with network routing dynamics in the next section.

Let M(s, d) denote a packet whose source address is s (or more generally, the address belongs to AS s) and whose destination address is d. A packet filtering scheme decides whether a packet should be forwarded or dropped based on certain criteria. One example is the route-based packet filtering [12].

Definition2 (route-based packet filtering): Node v accepts packet M(s, d) that is forwarded from node u if and only if e(u, v) € best R(s, d). Otherwise, the source address of the packet is spoofed, and the packet is discarded by v. In the context of preventing IP spoofing, an ideal packet filter should discard spoofed packets while allowing legitimate packets to reach the destinations. Since, even with the perfect routing information, the route-based packet filters cannot identify all spoofed packets [12], a valid packet filter should focus on not dropping any legitimate packets while providing the ability to limit spoofed packets. Accordingly, we define the correctness of a packet filter as follows.

Definition3 (correctness of packet filtering): A packet filter is correct if it does not discard packets with valid source addresses when the routing system is stable .Clearly, the route-based packet filtering is correct, because valid packets from source s to destination d will only traverse the edges on best R(s, d). Computing route-based packet filters requires the knowledge of best R(s, d) on every node, which is impossible in BGP. IDPF overcomes this problem.

4. IDPF OVERVIEW

The following concepts will be used in this section. A topological route between nodes s and d is a loop-free path between the two nodes. Topological routes are implied by the network connectivity. A topological route is a feasible route under BGP if and only if the construction of the route does not violate the routing policies imposed by the commercial relationship between ASs (Tables 1 and 2). Formally, let feasible R(s, d) denote the set of feasible routes from s to d.

Definition4 (feasible upstream neighbor): Consider a feasible route r2 feasible R(s, d). If an edge e(u, v) is on the feasible route, that is, e(u, v) € r.as path, we say that node u is a feasible upstream neighbor of node v for packet M(s, d). The set of all such feasible upstream neighbors of v (for M(s, d)) is denoted as feasible U(s, d, v). The intuition behind the IDPF framework is the following. First, it is possible for a node v to infer its feasible upstream neighbors by using BGP route updates. The technique for inferring feasible upstream neighbors is described in the next section. Since best R(s, d) 2 candidate R(s, d) + feasible R(s, d), a node can only allow M(s, d) from its feasible upstream neighbors to pass and discard all other packets. Such a filtering will not discard packets with valid source addresses. Second, although network connectivity (topology) may imply a large number of topological routes between a source and a destination, the commercial relationship between ASs and routing policies employed by ASs act to restrict the size of feasible R(s, d). Consider the example in Fig. 1. Figs. 2a and 2b present the topological routes implied by the network connectivity and feasible routes constrained by routing policies between source s and destination d, respectively. In Fig. 2b, we assume that nodes a, b, c, and d have mutual peering relationship, and that a and b are providers to s. We see that although there are 10 topological routes between source s and destination d, we only have two feasible routes that are supported by routing policies. Of more importance to IDPF is that although the network topology may imply that all neighbors can forward a packet allegedly from a source to a node, feasible routes constrained by routing policies help limit the set of such neighbors. It is clear that IDPF is less powerful than route-based packet filters [12], since the IDPFs are computed based on feasible R(s, d) instead of best R(s, d). However, feasible U (s, d, v) can be inferred from local BGP updates, whereas best U(s, d, v) cannot.

5. CONSTRUCTING IDPFS

The following lemma summarizes the technique for identifying the feasible upstream neighbors of node v for packet M(s, d).

Lemma1: Consider a feasible route r between source s and destination d. Let v 2 r.as path and let u be the feasible upstream neighbor of node v along r. When the routing system is stable, export(u -> v)[f best R(u, s)g+ 6= f assuming that all ASs follow the import and export routing policies in Tables 1 and 2 and that each AS accepts legitimate routes exported by neighbors. Lemma 1 states that if node u is a feasible upstream neighbor of node v for packet M(s, d), node u must have exported to node v its best route to reach the source s.

Proof: Since Theorem 1 applies to feasible routes, a feasible route can be one of the six types of paths in Theorem 1. In the following, we assume that the feasible route r is of type 6, that is, an uphill path followed by a peer-to-peer edge, which is followed by a downhill path. Cases where r is of types 1-5 can similarly be proved. To prove the lemma, we consider the possible positions of nodes u and v in the feasible route.

Case1. Nodes u and v belong to the uphill path. Then, node s must be an (indirect) customer or sibling of node u. From the import routing policies in Table 1 and the export routing policy r1 and the definition of indirect customers/siblings, we know that u will propagate to (provider) node v the reachability information of s.

Case2. The e(u, v) is the peer-to-peer edge. This case can similarly be proved as case 1 (based on the import routing policies in Table 1 and the export routing policy r3).

Case3. Nodes u and v belong to the downhill path. Let e(x, y) be the peer-to-peer edge along the feasible route r and note that u is an (indirect) customer of y. From the proof of case 2, we know that node y learns the reachability information of s from x. From the export routing policy r2 and the definition of indirect customers, node y will propagate the reachability information of s to node u, which will further export the reachability information of s to (customer) node v. Based on Lemma 1, a node can identify the feasible Upstream neighbors for packet M(s, d) and conduct IDPF as follows.

Definition5 (IDPF): Node v will accept packet M(s, d) that is forwarded by a neighbor node u if and only if export (u ->v[f best R(u, s)g 6= f Otherwise, the source address of the packet must have been spoofed, and the packet should be discarded by node v.

6. CONCLUSION

In this paper, we have proposed and studied an IDPF architecture as an effective countermeasure to the IP spoofing based DDoS attacks. IDPFs rely on BGP update messages exchanged on the Internet to infer the validity of source address of a packet forwarded by a neighbor. We showed that IDPFs can easily be deployed on the current BGP-based Internet routing architecture. We studied the conditions under which the IDPF framework can correctly work without discarding any valid packets. Our simulation results showed that, even with partial deployment on the Internet, IDPFs can significantly limit the spoofing capability of attackers. Moreover, they also help pinpoint the true origin of an attack packet to be within a small number of candidate networks, thus simplifying the reactive IP trace back process.

ACKNOWLEDGMENT

I would like to grate fully thank to Ms Pragati ptil for the inspiration ang Mr Yogesh Bhute for guiding and encouraging and explaining the topics.

REFERENCES

[1] "Unicast Reverse Path Forwarding Loose Mode "Cisco Systems, http.//www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newf%t/122t/122t13/ft+urpf.pdf, 2007.

[2] C. Jin, H. Wang, and K. Shin, "Hop-Count Filtering. An Effective Defense against Spoofed DDoS Traffic," Proc. 10th ACM Conf. Computer and Comm. Security, Oct. 2003.
[3] A. Yaar, A. Perrig, and D. Song, "Pi. A Path Identification Mechanism to Defend against DDoS Attacks," Proc. IEEE Symp. Security and Privacy, May 2003.
[4] A. Yaar, A. Perrig, and D. Song, "StackPi. New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense," IEEE J. Selected Areas in Comm., vol. 24, no. 10, Oct. 2006.
[5] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "Save. Source Address Validity Enforcement Protocol," Proc. IEEE INFOCOM, June 2002.
[6] A. Bremler-Barr and H. Levy, "Spoofing Prevention Method," Proc. IEEE INFOCOM, Mar. 2005.
[7] X. Liu, X. Yang, D. Wetherall, and T. Anderson, "Efficient and Secure Source Authentication with Packet Passport," Proc. Second Usenix Workshop Steps to Reducing Unwanted Traffic on the Internet (SRUTI '06), July 2006.
[8] P. Ferguson and D. Senie, Network Ingress Filtering. Defeating Denialof Service Attacks Which Employ IP Source Address Spoofing, RFC2267, Jan. 1998.
[9] The Team Cymru Bogon Route Server Project,"Team Cymru,

AUTHOR'S BIOGRAPHY

Miss Yogita V Deshmukh has completed her BE in information Technology from SHRI GAJANAN MAHARAJ COLLEGE OF ENGINEERING SHEGAON from Amravati University. Now she is pursuing MTECH in Computer Science and Engineering from Nagpur University. She is having about three years of teaching experience.