GROUP SIGNATURE PROTOCOL FOR SECURITY TREATMENT FOR BLOCKING MISBEHAVING USERS

S Kishor Kumar 1*, M.Rajasekhar 2*

1. *Research scholar, Avr and Svr Engineering College, Kurnool*
2. *Asst Prof, Avr and Svr Engineering College, Kurnool.*

Abstract: *IP spoofing is almost always used in what is currently one of the most difficult attacks to defend againstdenial of service attacks, or DoS. Since crackers are concerned only with consuming bandwidth and resources, they need not worry about properly completing handshakes and transactions. Rather, they wish to flood the victim with as many packets as ossible in a short amount of time. In order to prolong the effectiveness of the attack, they spoof source IP addresses to make tracing and stopping the DoS as difficult as possible. When multiple compromised hosts are participating in the attack, all sending spoofed traffic; it is very challenging to quickly block traffic. While some of the attacks described above are a bit outdated, Such as session hijacking for host-based authentication services, IP spoofing is still prevalent in network scanning and probes, as well as denial of servicefloods. However, the technique does not allow for anonymous Internet access, which is a common misconception for those unfamiliar with the practice. Any sort of spoofing beyond simple floods is relatively advanced and used in very specific instances such as evasion and connection hijacking..*

1. INTRODUCTION

In recent years, more and more networks with sensitive or even business critical data on them are being interconnected. Simultaneously, hacker activity has grown tremendously because of freely available hacker tools. In order to protect networks, so-called firewalls are deployed that protect against hacker activities. One of the ways to implement a firewall is to make use of so-called packet filters. Packet filtering has proved to be a handy tool to put access controls to IP traffic. Packet filters can be used to block IP packets based on certain criteria such as the protocol used and various protocol characteristics. In early packet filters, filtering decisions were made based solely on the packet that is currently inspected. Data like the source and destination addresses and in UDP and TCP cases the source and destination ports could be used in the filtering decisions.

Even the well known 'established' keyword, was based on static information (it inspected the presence of the ACK and RST flags in TCP return traffic. Such filtering could be very well used to protect against spoofing attacks where the attacker would send packets that seem to originate from systems on the inside of the packet filter.

II. IP PACKET FILTERING

Whenever a packet is allowed by the filter code to pass through the filtering host, the filter code allows for the creation of a state entry. New packets arriving at the filtering host are first matched against the state entry table. In case of a match, the state entry is updated if necessary and the packet is allowed through. First, the source address, source port, destination address and destination ports are matched. If a match is found, a special piece of code is executed that inspects if the ack and seq fields are valid for the given state entry.

**** S Kishor Kumar***
Research scholar, Avr and Svr Engineering Colleg, Kurnool.

IP spoofing will remain popular for a number of reasons. First, IP spoofing makes isolating attack traffic from legitimate traffic harder: packets with spoofed source addresses may appear to be from all around the Internet Second it presents the attacker with an easy way to insert a level of indirection.

III. BGP

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 reach ability 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.

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. The Border Gateway Protocol (BGP) is the core routing protocol of the Internet. It maintains a table of IP networks or 'prefixes' which designate network reach ability among autonomous systems (AS).

It is described as a path vector protocol. 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 Hop counter: A hop count is the number of times that a URL request from a Web browser ventures from one router to another. Ingress filter work: Ingress filtering primarily prevents a specific network from being used for attacking others.

IV. INTERDOMAIN PACKET FILTER

Route-based packet filters-mitigating IP spoofing 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. Problem: filter requires the knowledge of global routing information, which is hard to reconcile in the current Internet routing infrastructure. Constructing route-based packet filters-acquire the complete knowledge of routing decisions made by all other AS. BGP is a policy-based routing protocol in that both the selection and the propagation of the best route to a destination at an AS are guided by some locally defined routing policies. Existing system uses Network Ingress Filtering. Ingress filtering primarily prevents a specific network from being used for attacking others.

Proposed System in our project: we propose and study 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. It correctly works without discarding any valid packets two distinct sets of routing policies are typically employed by a node: import policies Neighbor-specific import policies are applied upon routes learned from neighbors export policies. -- Whereas neighbor-specific export policies are imposed on locally selected best routes before they are propagated to the neighbors. BGP is an incremental protocol a downhill path is a sequence of edges that are either provider-to-customer or sibling-to sibling edges an uphill path is a sequence of edges that are either customer-to-provider or sibling-to-sibling edges in our project we propose and study 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. Minimize the denial of service attacks. For finding possible path we don't need globule routing information. Reducing the IP spoofing through BGB updates, this will overcome the drawback of finding BEST route.

We are constructing the IDPF to control IP Forging. Before controlling IP Forging We have to know what the attacks are present in the internet and why we are controlling IP Forging ,is 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 steal the source addresses in IP packets.

By employing IP stealing or forging, attackers can evade detection and put a substantial burden on the destination network for policing attack packets. In this, we propose an inter-domain packet filter (IDPF) architecture that can mitigate the level of IP forging 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 forging capability of attackers. In addition, they can help localize the origin of an attack packet to a small number of candidate networks.

In this chapter, the intuition behind the IDPF architecture is discussed, it is shown how IDPFs are constructed using BGP route updates, and establishes the correctness of IDPFs.After that, the case where ASs have routing policies that are less restrictive are been discussed. It is assumed that the routing system is in the stable routing state in this chapter.IDPFs fare with network routing dynamics is also discussed.

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.

Definition 2 (route-based packet filtering): Node v accepts packet M(s, d) that is forwarded from node u if and only if e (u, v) Î 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, the correctness of a packet filter is defined as follows:

Definition 3 (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 bestR(s, d). Computing route-based packet filters requires the knowledge of bestR(s, d) on every node, which is impossible in BGP. IDPF overcomes this problem.

IDPF Overview

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. Formally, let

feasibleR(s, d) denote the set of feasible routes from s to d. Then, feasibleR(s, d) can recursively be defined as follows:

feasibleR(s,d) = { < s ⊕ ∪ feasibleR(u,d) > },
u: import (s κ u) [{r}] { }
r.prefix = d,u İ N(s)

Where ⊕ is the concatenation operation, for example, { s ⊕ { <ab> , <uv> }} = { <sab> , <suv> }.

It is noticed that feasibleR(s,d) contains all the routes between the pair that does not violate the import and export routing policies. Obviously, bestR(s,d) İ candidateR(s,d) ⊆ feasibleR(s,d). Each of the feasible routes can potentially be a candidate route in a BGP routing table. Theorem 1 also applies to feasible routes.

Definition 4 (feasible upstream neighbor): Consider a feasible route r İ feasibleR(s,d). If an edge e(u,v) is on the feasible route, that is, e(u,v) İ 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 feasibleU(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 bestR(s,d) İ candidateR(s, d) feasibleR(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 feasibleR(s,d).

Consider the example in Fig. 4.1. Figs. 4.1 a and 4.2 b present the topological routes implied by the network connectivity and feasible routes constrained by routing policies between source s and destination d, respectively.

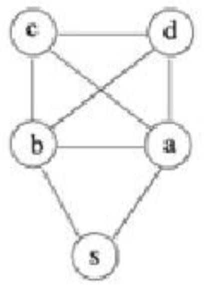

Fig 4.1 An Example network topology

In Fig. 4.2b, 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.

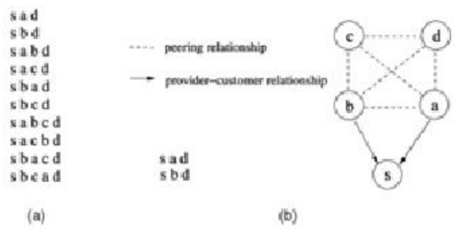

Fig 4.2 Routes between source s and destination d (a) Topological routes implied by connectivity. (b) Feasible routes constraints by routing policies

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. As an example, Let us consider the situation at node d. Given that only nodes a and b (but not c) are on the feasible routes from s to d, node d can infer that all packets forwarded by node c and allegedly from source s are spoofed and should be discarded. It is clear that IDPF is less powerful than routebased packet filters [12], since the IDPFs are computed based on feasibleR(s,d) instead of bestR(s,d). However, feasibleU(s,d,v) can be inferred from local BGP updates, whereas bestU(s,d,v) cannot be inferred.

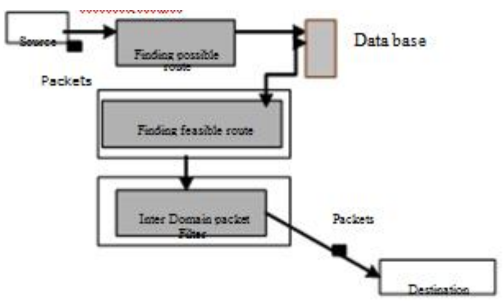

Fig 4.3 System Architecture

Constructing IDPFs

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

Lemma 1: Consider a feasible route r between source s and destination d. Let v Î r.as_path and let u be the feasible upstream neighbor of node v along r. When the routing system is stable, export(uÍv) [{ bestR(u,s) }] { },assuming that all ASs follow the import and export routing policies 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. 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:

Case 1: 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 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.

Case 2: e(u,v)is the peer-to-peer edge. This case can similarly be proved as case 1 (based on the import routing policies and the export routing policy r3).

Case 3: 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:

Definition 5 (IDPF): Node v will accept packet M(s,d) that is forwarded by a neighbor node u if and only if export(uÍv) [{ bestR(u,s) }] { }. Otherwise, the source address of the packet must have been spoofed, and the packet should be discarded by node Correctness of IDPF

Theorem 2: An IDPF, as defined in Definition 5, is correct.

Proof: Without loss of generality, consider source s, destination d, and a node v Î bestR(s,d).as_path such that v deploys an IDPF. To prove, it is needed to establish that v will not discard packetM(s,d) forwarded by the best upstream neighbor u along bestR(s,d).

Since bestR(s,d) Î candidateR(s,d) _ feasibleR (s,d), u is also a feasible upstream neighbor of node v for packet M(s,d). From Lemma 1, u must have exported to node v its best route to source s. That is, export(uÍv) [{ bestR(u,s) }]

{ }. From Definition 5, packet M(s,d), which is forwarded by node u, will not be discarded by v. The destination address d in a packet M(s,d) does not play a role in an IDPF node's filtering decision is noticed.

(Definition 5): By constructing filtering tables based on the source address alone (rather than both source and destination addresses), the per-neighbor space complexity for an

IDPF node is reduced from O(N2) to O(N), where N = |V | is the number of nodes in the graph (the route-based scheme can achieve the same complexity bound [12]).

It is worth noting that IDPFs filter packets based on whether the reachability information of a network prefix is propagated by a neighbor and not on how the BGP updates are propagated. As long as ASs propagates network reachability information according to the rules, IDPFs work correctly.

Moreover, the effectiveness of IDPFs is determined largely by the size of feasibleR(s,d), which is a function of the (relatively static) AS relationships. Hence, how the BGP updates are propagated does not affect both the correctness and the performance of IDPFs.

REFERENCES

[1]R. Beverly and S. Bauer, “The Spoofer Project: Inferring the Extent of Internet Source Address Filtering on the Internet,” Proc. First Usenix Steps to Reducing Unwanted Traffic on the Internet Workshop, July 2005.
[2]S. Kandula, D. Katabi, M. Jacob, and A. Berger, “Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds,” Proc. Second Symp. Networked Systems Design and Implementation, 2005.

[3]D. Moore, C. Shannon, D. Brown, G. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," ACM Trans. Computer Systems, vol. 24, no. 2, May 2006.
[4]R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of Internet Background Radiation," Proc. ACM Internet Measurement Conf., Oct. 2004.
[5]S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," Proc. ACM SIGCOMM Computer Comm. Rev., vol. 30, no. 4, Oct. 2000.
[6]P. Watson, "Slipping in the Window: TCP Reset Attacks," Proc. Fifth CanSec West/core04 Conf., 2004.
[7]J. Stewart, "DNS Cache Poisoning—The Next Generation,"technical report, LURHQ, Jan. 2003.
[8]V. Paxson, "An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks," ACM Computer Comm. Rev., vol. 31, no. 3, July 2001.
[9]"CERT Advisory ca-1996-21 TCP SYN Flooding and IP Spoofing Attacks,"CERT, http://www.cert.org/advisories/CA-1996-21.html, 1996.