IJIRSET

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

Source Address Validation Implementation by Using BGP

R.Vishal, R.Angeline

M.Tech (CSE) Student, SRM University, Ramapuram Campus, Chennai,India

Assistant Professor, Department of Computer Science and Engineering, SRM University, Ramapuram Campus, Chennai, India

Abstract: The persistent evolution of the Internet continues to transform the way individuals, as well as businesses, educational institutions, and government organizations access, share, and communicate information. Convergence of digital voice, video, and data, is further consolidating the Internet as a critical infrastructure. One of the main routing protocols in the Internet and current de facto standard is the Border Gateway Protocol (BGP). Presently ubiquitous, BGP is a critical component of the exponentially growing network of routers that constitutes our contemporary Internet. Carrier networks, as well as most large enterprise organizations with multiple links to one or more service providers use BGP. 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 Source Address Validation Implementation (SAVI) 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. SAVIs 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 SAVI 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, SAVIs 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: Distributed Denial-of-Service (DDoS), Source Address Validation Implementation (SAVI), Border Gateway Protocol (BGP), Source Address Validation.

I. INTRODUCTION

The connectionless paradigm of the network layer of the Internet has naturally made possible source address spoofing, that is, the use of a source address by a node which is not allowed to use that address. Attackers can use spoofed source addresses to prevent tracing and to defeat source-based filtering when performing flood based denial-of-service or poisoning attacks, or when propagating worms or malware. The concern about the risks induced by source address spoofing has resulted in the recommendation of the deployment of ingress filtering. This technique consists of the filtering of any packet with a source address that does not belong to the set of prefixes assigned to the part of the topology from which the packet comes. Ingress filtering is usually performed close to the site at which packets are originated, in either the egress router of the site or the ingress router of the direct provider. This strategy of deploying filters close to the site implies not only more effective protection, but also that it is easier to determine the prefixes corresponding to the site, by either explicit configuration or inference from the routing tables. The effectiveness of this technique largely depends on its widespread deployment, since any host connected to a site where ingress filtering is not performed could generate packets with any forged address.

However, even if ingress filtering were universally deployed, we would still face residual vulnerabilities. Because ingress filtering operates at the prefix level, an attacker can still spoof any address from the prefix assigned to that part of the topology. This enables a number of serious attack vectors, allowing, for example, worms/malware to spoof a source address in order to hide the identity of the infected system. For a thorough description of this and other attacks enabled by source address spoofing that are not protected by ingress filtering.

In this paper, we present SAVI, a mechanism that complements ingress filtering and provides increased protection. SAVI is in the final stages of standardization in the Internet Engineering Task Force (IETF). In a nutshell, SAVI protects each individual address from spoofing by placing the source address filters closer to the nodes, preferably in the layer 2 switches that connect the nodes of a link. In this deployment scenario, a switch configures a SAVI binding between an IP address and a node-specific layer 2 binding anchor. The physical port of the switch to which the node is attached is the canonical example of a binding anchor. The bindings are automatically created by inspecting the protocol exchange used to configure the IP addresses of the nodes. This allows the switch to filter out packets that do not correspond to an existing SAVI binding. The contributions of this paper are the following. We provide an integrated perspective of the SAVI solution whose components are described in multiple IETF documents, we provide insights into the rationale for key design decisions, and we provide the background needed to understand the different requirements that lead to the final design; we also compare the solutions and discuss their applicability.

II. OBJECTIVE AND MOTIVATION

The objective of this work is to reduce datagram's with spoofed IP addresses from the Internet. This can be aided by Identifying and dropping datagram's whose source address binding is incompatible with the Internet topology and learned information. This can be done at sites where the relationship between the source address and topology and binding information can be checked. It uses Network Ingress Filtering. Ingress filtering primarily prevents a specific network from being used for attacking others. This technique consists of the filtering of any packet with a source address that does not belong to the set of prefixes assigned to the part of the topology from which the packet comes. Ingress filtering is usually performed close to the site at which packets are originated, in either the ingress router of the site or the ingress router of the direct provider. This strategy of deploying filters close to the site implies not only more effective protection, but also that it is easier to determine the prefixes corresponding to the site, by either explicit configuration or inference from the routing tables. The effectiveness of this technique largely depends on its widespread deployment, since any host connected to a site where ingress filtering is not performed could generate packets with any forged address.

Even if ingress filtering were universally deployed, we would still face residual vulnerabilities. In particular, because ingress filtering operates at the prefix level, an attacker can still spoof any address from the prefix assigned to that part of the topology. This enables a number of serious attack vectors, allowing, for example, worms/malware to spoof a source address in order to hide the identity of the infected system. For a thorough description of this and other attacks enabled by source address spoofing that are not protected by ingress filtering. In this paper we propose and study Source Address Validation Implementation (SAVI) as an effective counter measure to the IP spoofing-based DDoS attacks. SAVIs rely on BGP update messages exchanged on the Internet to infer the validity of source address of a packet forwarded by a neighbour. It correctly works without discarding any valid packets. Two distinct sets of routing policies are typically employed by a node:

1. Import policies -- Neighbor-secific import policies are applied upon routes learned from neighbors
2. 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
1. A downhill path is a sequence of edges that are either provider-to-customer or sibling-to sibling edges
2. An uphill path is a sequence of edges that are either customer-to-provider or sibling-to-sibling edges

III. LITERATURE REVIEW

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.
“The Spoofer Project: Inferring the Extent of Internet Source Address Filtering on the Internet**”,** Robert Beverly *MIT CSAIL* , Steven Bauer-Forging, or ”spoofing,” the source addresses of IP packets provides malicious parties with anonymity and novel attack vectors. Spoofing-based attacks complicate network operator’s defence techniques; tracing spoofing remains a difficult and largely manual process. More sophisticated next generation distributed denial of service (DDoS) attacks may test filtering policies and adaptively attempt to forge source addresses. To understand the current state of network filtering, this paper presents an Internet-wide active measurement spoofing project. Clients in our study attempt to send carefully crafted UDP packets designed to infer filtering policies. When filtering of valid packets is in place we determine the filtering granularity by performing adjacent net block scanning. Our results are the first to quantify the extent and nature of filtering and the ability to spoof on the Internet. We find that approximately one-quarter of the observed addresses, net blocks and autonomous systems (AS) permit full or partial spoofing. Projecting this number to the entire Internet, an approximation we show is reasonable, yields over 360 million addresses and 4,600 A Ses from which spoofing is possible. Our findings suggest that a large portion of the Internet is vulnerable to spoofing and concerted attacks employing spoofing remain a serious concern.

“Botz4Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds” Srikanth Kandula, Dina Katabi,Matthias Jacob ,Arthur Berger-Recent denial of service attacks are mounted by professionals using Botnets of tens of thousands of compromised machines. To circumvent detection, attackers are increasingly moving away from bandwidth floods to attacks that mimic theWeb browsing behavior of a large number of clients, and target expensive higher-layer resources such as CPU, database and disk bandwidth. The resulting attacks are hard to defend against using standard techniques, as the malicious requests differ from the legitimate ones in intent but not in content. We present the design and implementation of Kill-Bots, a kernel extension to protect Web servers against DDoS attacks that masquerade as flash crowds. Kill-Bots provides authentication using graphical tests but is different from other systems that use graphical tests. First, Kill-Bots uses an intermediate stage to identify the IP addresses that ignore the test, and persistently bombard the server with requests despite repeated failures at solving the tests. These machines are bots because their intent is to congest the server. Once these machines are identified, Kill-Bots blocks their requests, turns the graphical tests off, and allows access to legitimate users who are unable or unwilling to solve graphical tests. Second, Kill-Bots sends a test and checks the client’s answer without allowing unauthenticated clients access to sockets, TCBs, and worker processes. Thus, it protects the authentication mechanism from being DDoSed. Third, Kill- Bots combines authentication with admission control. As a result, it improves performance, regardless of whether the server overload is caused by DDoS or a true Flash Crowd.

“An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks**”** Vern Paxson-Attackers can render distributed denial-of service attacks more difficult to defend against by bouncing their flooding traffic off of reflectors; that is, by spoofing requests from the victim to a large set of Internet servers that will in turn send their combined replies to the victim. The resulting dilution of locality in the flooding stream complicates the victim’s abilities both to isolate the attack traffic in order to block it, and to use traceback techniques for locating the source of streams of packets with spoofed source addresses, such as ITRACE, probabilistic packet marking and SPIE. We discuss a number of possible defenses against reflector attacks, finding that most prove impractical, and then assess the degree to which different forms of reflector traffic will have characteristic signatures that the victim can use to identify and filter out the attack traffic. Our analysis indicates that three types of reflectors pose particularly significant threats: DNS and Gnutella servers and TCP-based servers (particularly Web servers) running on TCP implementations that suffer from predictable initial sequence numbers. We argue in conclusion in support of “reverse ITRACE”and for the utility of packet traceback techniques that work even for low volume flows, such as SPIE.

“Practical Network Support for IP Traceback**”**, Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson-This paper describes a technique for tracing anonymous packet flooding attacks in the Internet back towards their source.

This work is motivated by the increased frequency and sophistication of denial-of-service attacks and by the difficulty in tracing packets with incorrect, or “spoofed”, source addresses. In this paper we describe a general purpose traceback mechanism based on probabilistic packet marking in the network. Our approach allows a victim to identify the network path(s) traversed by attack traffic without requiring interactive operational support from Internet Service Providers (ISPs). Moreover, this trackback can be performed “post-mortem” – after an attack has completed. We present an implementation of this technology that is incrementally deployable, (mostly) backwards compatible and can be efficiently implemented using conventional technology.

“Inferring Internet Denial-of-Service Activity**”** David Moore-In this paper, we seek to answer a simple question: “How prevalent are denial-of-service attacks in the Internet today?”. Our motivation is to understand quantitatively the nature of the current threat as well as to enable longer term analyses of trends and recurring patterns of attacks. We present a new technique, called “backscatter analysis”, that provides an estimate of worldwide denial-of service activity. We use this approach on three week-long datasets to assess the number, duration and focus of attacks, and to characterize their behavior. During this period, we observe more than 12,000 attacks against more than 5,000 distinct targets, ranging from well known ecommerce companies such as Amazon and Hotmail to small foreign ISPs and dial-up connections. We believe that our work is the only publically available data quantifying denial-of-service activity in the Internet.

IV. SAVI ARCHITECTURE

SAVI prevents IP source address spoofing by filtering out packets for which a SAVI binding does not exist. A SAVI binding is an association between an IP address and a binding anchor, a property of the host’s network attachment, such as the physical port of the SAVI device to which the host connects. The binding anchor must be verifiable and hard to spoof. The current SAVI specifications consider the host’s attachment port as the binding anchor. In this case, packets with a given IP source address are forwarded only when arriving from a particular physical port. However, SAVI specifications are flexible enough to support other binding anchors, such as IEEE 802.1X security associations. We can identify two main architectural components of SAVI, the binding creation mechanism and the filtering mechanism. SAVI bindings are created dynamically as a result of the traffic inspection process, according to the address configuration mechanism used in the link. Each SAVI solution defines its own rules for the creation and refreshing of bindings. The filtering mechanism inspects data packets and verifies their source address and anchors against the existing list of bindings.

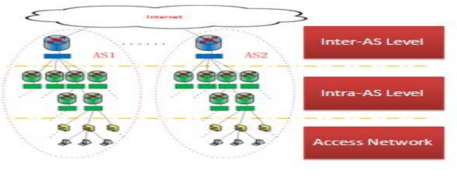


Figure1. SAVI System Architecture

The address configuration messages used to create the bindings are processed according to special filtering rules, as we describe in detail for each particular SAVI solution. Note that the filtering and binding creation processes are essentially orthogonal. In this section we describe the filtering mechanisms that are common for the different means to create bindings. BASE router marks packets with a unique MAC and use the MAC as the incoming direction.The MAC is distributed via BGP. In this process, the community property of BGP update message is used.

V. BGP AND MESSAGES

SAVIs rely on BGP update messages exchanged on the Internet to infer the validity of source address of a packet forwarded by a neighbor.BGP is considered a "Path Vector" routing protocol. BGP was not built to route within an Autonomous System (AS), but rather to route between AS's. BGP maintains a separate routing table based on shortest AS Path and various other attributes, as opposed to IGP metrics like distance or cost. For BGP to function, BGP routers (called speakers) must form neighbor relationships (called peers).

BGP forms its peer relationships through a series of messages. First, an OPEN message is sent between peers to initiate the session. The OPEN message contains several parameters: KEEPALIVE messages are sent periodically (every 60 seconds by default) to ensure that the remote peer is still available. If a router does not receive a KEEPALIVE from a peer for a Hold-time period (by default, 180 seconds), the router declares that peer dead. UPDATE messages are used to exchange routes between peers. BGP systems send update messages to exchange network reach ability information. BGP systems use this information to construct a graph that describes the relationships among all known ASs.

Update messages consist of the BGP header plus the following optional fields:

1. Unfeasible routes length—Length of the withdrawn routes field.
2. Withdrawn routes—IP address prefixes for the routes being withdrawn from service because they are no longer deemed reachable.
3. Total path attribute length—Length of the path attributes field; it lists the path attributes for a feasible route to a destination.
4. Path attributes—Properties of the routes, including the path origin, the multiple exit discriminators (MED), the originating system's preference for the route, and information about aggregation, communities, confederations, and route reflection.
5. Network layer reachability information (NLRI)—IP address prefixes of feasible routes being advertised in the update message.

VI. SOURCE ADDRESS VALIDATION

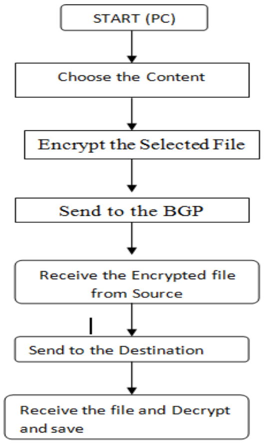
BGP (Border Gateway Protocol) is a protocol that communicates across the network and also monitoring the client present in the network. BGP is a protocol for facilitating communications between routers in different autonomous systems. An autonomous system (AS) is a network or group of networks under a shared technical administration and with common routing policies. BGP conveys information about AS-Path topologies and achieves inter-AS routing without constraining the underlying network topology. An intra-AS routing protocol that is, Interior Gateway Protocol (IGP), examples of which are Routing Information Protocol (RIP), Open Shortest Path First (OSPF), etc.—provides the routing within an autonomous system. In some circumstances, BGP is used to exchange routes within an AS. In those cases, it is called Internal BGP (I-BGP), as opposed to External BGP (E-BGP) when used between ASs.
There are four possible message types used with BGP, all consisting of a standard header plus specific packet-type contents:

OPEN- First message to open a BGP session, transmitted when a link to a BGP neighbor comes up. It contains AS number (ASN) and IP address of the router who has sent the message.
UPDATE- Message embracing routing information, including path attributes. It contains Network Layer Reachability Information (NLRI), listing IP addresses of new usable routes as well as routes that are no longer active or viable and including both the lengths and attributes of the corresponding paths.

NOTIFICATION- Final message transmitted on a link to a BGP neighbor before disconnecting. It usually describes atypical conditions prior to terminating the TCP connection, and provides a mechanism to gracefully close a connection between BGP peers

KEEP-ALIVE-Periodic message between BGP peers to inform neighbor that the connection is still viable by guaranteeing that the transmitter is still alive. It is an application type of message that is independent of the TCP keep-alive option.

It has all client details as a table. The connection is established with the client and the Router. The Encrypted data is transmitted to the Router which can send or redirect to the correct destination address. The Router checks whether the sender and receiver are proper to the network. So the validation module loaded with client details from BGP update messages. Whenever the packet comes through the router, it will check the source address of that packet with loaded client details.



In case the sender (hacker) is not a proper member in the network then that node is said to the attacker node, then the message will not sent to the destination. Otherwise the message will send to the destination address. The Border Gateway Protocol, used to manage routing policy between large networks.

VII. CONCLUSION

We have proposed and studied SAVI as an effective counter measure to the IP spoofing-based DDoS attacks. SAVIs rely on BGP update messages exchanged on the Internet to infer the validity of source address of a packet forwarded by a neighbor. SAVIs are constructed from the information implicit in BGP route updates and are deployed in network border routers. We studied the conditions under which the SAVI framework can correctly work without discarding any valid packets. Our simulation results showed that, even with partial deployment on the Internet, SAVIs 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. BGP is a core component of the Internet, connecting virtually all autonomous systems across the globe. It has prevailed due to its continuous adaptation to varying requirements, and will continue to be the standard protocol of inter-domain routing. Equipment vendors, carriers and service providers, as well as enterprise customers depend on the interoperability, scalability, and performance of their network equipment to perform multiple services, critical to their communications and core infrastructures.

REFERENCES

[1] D. McPherson, F. Baker, and J. Halpern, “SAVI Threat Scope,” draft-ietf-savi-threat-scope-05, Apr. 2011.
[2] P. Ferguson and D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing,” IETF RFC 2827, May 2000.
[3] J. Bi et al., “SAVI Solution for DHCP,” draft-ietf-savi- dhcp-15, Sept. 2012.
[4] E. Nordmark, M. Bagnulo, and E. Levy-Abegnoli, “FCFS SAVI: First-Come First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses,” IETF RFC 6620, May 2012.
[5]. S. Thomson, T. Narten, and T. Jinmei, “IPv6 Stateless Address Auto configuration,” IETF RFC 4862, Sept. 2007.
[6] M. Bagnulo and A. Garcia-Martinez, “SEND-Based Source-Address Validation Implementation,” draft-ietf- savi-send-08. Sept. 2012.
[7] J. Arkko et al., “Secure Neighbor Discovery (SEND),” IETF RFC 3971, Mar. 2005.
[8] IEEE, “IEEE 802.1X-2010 Port-Based Network Access Control,” Feb. 2010.
[9] J. Wu et al., “Source Address Validation Improvement Framework,” draft-ietf-savi-framework-06, Dec. 2011.
[10] R. Droms. “Dynamic Host Configuration Protocol,” IETF RFC 2131, Mar. 1997.
[11] R. Droms et al., “Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” IETF RFC 3315, July 2003.
[12] T. Narten et al., “Neighbor Discovery for IP Version 6 (IPv6),” IETF RFC 4861, Sept. 2007.