Submitted to ICN 2002

Organic Techniques for Protecting Virtual Private Network (VPN) Services from Access Link Flooding Attacks[1]

Ranga S. Ramanujan
Maher Kaddoura
John Wu
Kevin Millikin

Architecture Technology Corporation
9971 Valley View Road
Eden Prairie, Minnesota 55344, USA

Doug Harper
David Baca

Odyssey Research Associates
33 Thornwood Drive
Ithaca, New York 14850, USA

Abstract

Distributed Denial-of-Service (DDoS) attacks represent a serious threat to enterprises operating over the Internet. A notable form of DDoS attack is the access link flooding attack that directs spurious packet traffic over the access link connecting an enterprise's network (i.e., an edge network) to the public Internet. Such overloading of the network access link by the attack traffic may result in partial or total denial of service to the subscribers of the edge network.

This paper presents several design techniques for protecting edge networks against access link flooding attacks. The approach for survivability employed by these techniques is predicated upon making the failover mechanisms that are invoked by the system upon detection of an attack appear to be an unpredictable process from the perspective of the attacker. A prototype implementation of an operational survivable virtual private network (VPN) service built using these techniques is also described.

Contact author:

Dr. Ranga S. Ramanujan
Architecture Technology Corporation
9971 Valley View Road
Eden Prairie, Minnesota 55344, USA

Phone: (+1) 952 829 5864 extn. 120
Fax: (+1) 952 829 5871
Email: ranga@atcorp.com

[1] This material is based upon work supported by DARPA under contract number DAAH01-96-R048. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2002		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Organic Techniques for Protecting Virtual Private Network (VPN) Services from AccessLink Flooding Attacks	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Advanced Research Projects Agency,3701 N Fairfax Dr,Arlington,VA,22203**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

Distributed Denial-of-Service (DDoS) attacks represent a serious threat to enterprises operating over the Internet. A notable form of DDoS attack is the access link flooding attack that directs spurious packet traffic over the access link connecting an enterprise's network (i.e., an edge network) to the public Internet. Such overloading of the network access link by the attack traffic may result in partial or total denial of service to the subscribers of the edge network. This paper presents several design techniques for protecting edge networks against access link flooding attacks. The approach for survivability employed by these techniques is predicated upon making the failover mechanisms that are invoked by the system upon detection of an attack appear to be an unpredictable process from the perspective of the attacker. A prototype implementation of an operational survivable virtual private network (VPN) service built using these techniques is also described.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**15**	

1. Introduction

A large number of organizations and enterprises have geographically dispersed operations with a local area network (LAN) supporting the information processing needs at each of these locations. Traditionally, interconnection of the dispersed LAN sites has been accomplished using dedicated communication lines leased from a service provider. In addition, Internet access at each site is typically accomplished using another leased line (such as a T1 or T3 line) that connects the site to a local Internet service provider. With the advent of virtual private network (VPN) technology, organizations can now accomplish inter-site network connectivity over the Internet. By obviating the need for dedicated lines between the sites, this solution yields substantial cost savings.

A VPN operates by transporting traffic between the sites using tunnels established over the Internet between these sites. Currently, there are three tunneling protocols that are used in a majority of commercially available VPN products, i.e., IP Security (IPSec) [1], Point-to-Point Tunneling Protocol (PPTP) [2], and Layer 2 Tunneling Protocol (L2TP) [3]. The tunnels established and maintained by these protocols may be viewed as implementing virtual leased lines between the geographically distributed LAN sites of an enterprise.

Although cost considerations clearly favor the use of inter-site VPNs over dedicated lines, the major impediment to the widespread employment of this technology today is its vulnerability to access link flooding attacks, a form of network borne distributed denial-of-service (DDoS) attack [4,5]. In an access link flooding attack, the attack traffic may be generated simultaneously from multiple points on the network from machines that have been "hijacked" or subverted by the attacker. This traffic flood, when directed at a victim LAN site, can inundate the access link connecting the site to its Internet service provider. By usurping access link bandwidth from the VPN tunnels operating over that link, the attack can cause partial or total denial of the VPN service and disrupt operations of any mission-critical application that relies on that service.

A number of techniques have been proposed recently to detect and counter access link flooding attacks. These techniques rely on mechanisms that must be partially or wholly implemented within the service provider network infrastructure to identify the source(s) of attack traffic. Once this is accomplished, generally manual actions are required to neutralize the effect of this traffic. This may involve, for instance, the installation of filters to discard attack traffic at the ingress to the service provider network. With this semi-automated approach, the time interval between the onset of an attack and its neutralization can be expected to be on the order of minutes at best and hours at worst. This interval represents a window of vulnerability for a VPN operating over the attacked access link.

This paper presents several automated techniques for building DoS-resistant (or survivable) VPNs that provide continuous, uninterrupted operation of the VPN services in spite of access link flooding attacks. In contrast to existing infrastructure-based techniques for detecting and countering these attacks, the techniques presented here are designed to be implemented within the enterprise edge networks (i.e., LAN sites connecting to the Internet). That is, the survivability mechanisms associated with this approach are implemented within the customer premises equipment and require no modifications or additions to equipment in the network infrastructure owned by the service provider. In that sense, the techniques presented here are *organic* survivability techniques for protecting VPNs from DoS attacks. They can be viewed as supplementing existing infrastructure-based techniques to overcome the window of vulnerability that exists with the existing techniques.

The rest of this paper is organized as follows. Section 2 describes the network environment targeted by this research and describes the threat model addressed by our survivability techniques.

Section 3 provides on overview of our approach for building survivable VPNs. Section 4 describes our current prototype implementation of a survivable VPN service. Sections 5 and 6 present related work and conclusions, respectively.

2. Background

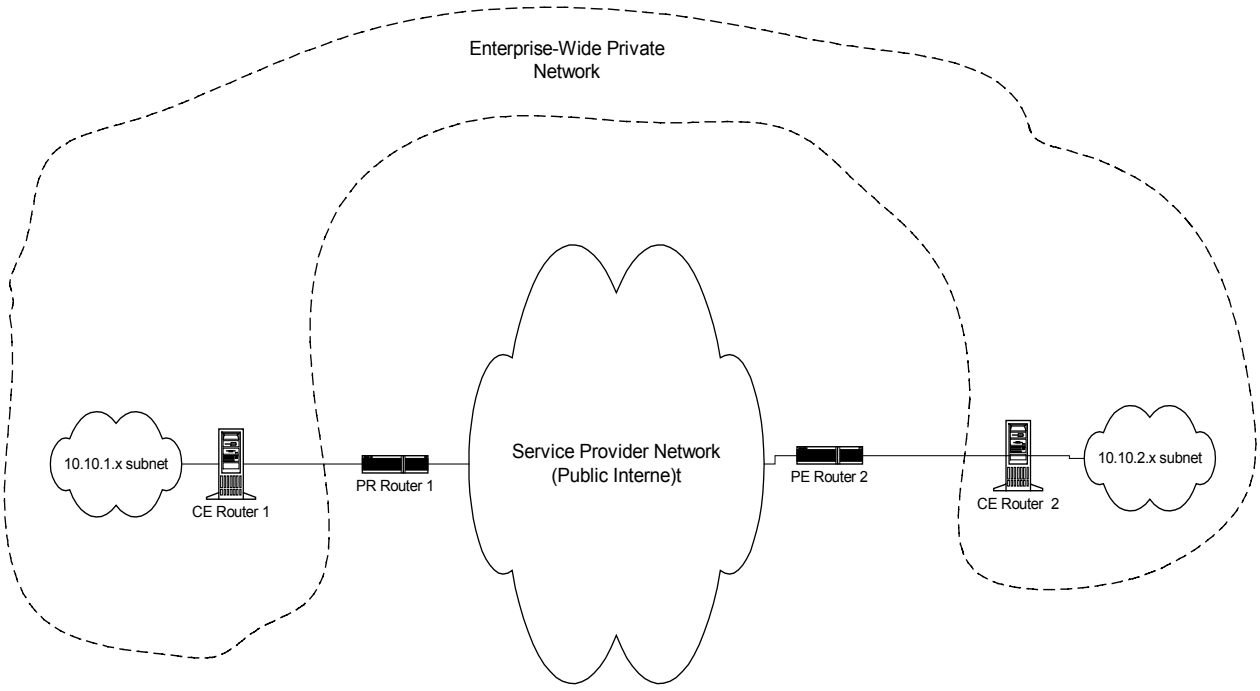

Figure 1: Example of VPN-based Network

Figure 1 depicts an example VPN networking environment typical of the kind targeted by the survivability techniques described in this paper. In the example scenario, the customer network (or enterprise network) is composed of two geographically distributed LAN sites that are connected together to form one virtual network using VPN tunnels implemented over the Internet. For simplicity, we assume IPSec-based VPN tunnels in this paper.

2.1 Definitions

The following paragraphs describe the terminology used in the rest of the paper.

Customer Network: A customer network is an edge network on the Internet that is owned and managed by the customer of the Internet service provider. Different sites of a customer network may be inter-connected through the service provider network via VPN services.

Customer Edge (CE) Router: This is the customer premise equipment that connects a particular customer site to a service provider network. For CE-based VPNs, assumed for this paper, a CE router also terminates VPN tunnels.

CE based VPN: In CE-based VPNs, the LAN sites of the customer network are connected together by VPN tunnels set up between CE routers.

Provider Edge (PE) Router: This is the provider edge (PE) equipment, which is connects to the CE router over the local access link. In our network environment, the PE device is an IP router.

Service Provider IP Network A service provider network is a network administered by one or more service providers that provides Internet connections to customer networks on its edge.

VPN tunnel: A VPN tunnel is a logical link between two CE routers that carries encapsulated packets between two CE devices. In the case of the IPSec VPNs assumed here, the encapsulation is carried out by using IPSec in the tunnel mode [1].

Thus, an IPSec VPN consists of a number of sites securely inter-connected through IPSec tunnels implemented over the Internet. Each site consists of the CE router, the customer network behind the CE router, and the access link to the PE router that connects the site to the Internet service provider network. On the customer network side, the CE router may be connected to one or more private IP subnets as well as globally reachable subnets. The geographically dispersed private subnets (10.10.1.x, and 10.10.2.x in Figure 1) are the ones that are interconnected into a virtual network (10.10.x.x. in the example) by VPN tunnels between them.

The CE router's interface to the public Internet may be configured with one or more globally reachable (or public) IP addresses, a subset of which can be reserved for use as VPN tunnel end points. This would enable the CE router to distinguish between VPN traffic received by it from its peer CE routers on the customer network and traffic destined for the globally reachable part of the customer network behind it. Hosts on the customer network may be connected to the public or private network or to both. Packets originating from one private subnet of the customer network and destined for a remote private subnet are encapsulated by the CE router at the originating subnet using an IPSec tunnel mode header. The encapsulated packet is thus encrypted before it is transported over the Internet to the destination CE router. The IPSec header also carries authentication data that can be used by the destination router to authenticate or verify the integrity of the received packet's content, including its headers. Upon successful authentication of a VPN packet, the CE router decrypts the encapsulated packet contained within and forwards it to the appropriate private subnet behind it. Thus, the CE router in our networking environment serves as a general IP router as well as a VPN gateway (for initiating and terminating VPN tunnels).

2.2 Threat Environment

The techniques presented in this paper for protecting VPNs from access link flooding attacks are targeted at a network environment that satisfies the following conditions or assumptions.

1. Flooding attacks are launched from the edge of the Internet. In other words, the attacker does not have access to facilities and equipment within the core of the Internet (within the service provider network) that would enable the attacker to snoop on and analyze the VPN traffic flowing between the CE routers.
2. The attack traffic does not originate from any of the customer equipment (i.e., CE routers, hosts) within the customer network that is being protected. That is, all attacks are outsider attacks.
3. Shared secrets between CE routers, such as those used to encrypt VPN traffic and control messages between them, are adequately protected against compromise or leaks to the attacker.
4. Although the volume of traffic generated by the attack source(s) may be sufficient to inundate an access link, it is not sufficient to disrupt the operation of the service provider network.

3. Survivable VPN Service

Our approach for VPN survivability relies on the fact that the globally reachable IP addresses associated with the two end points of a VPN tunnel are assigned by the CE routers at the two ends of that tunnel. A CE router provisions a set of alternate public IP addresses for each VPN tunnel terminating on it. One of these addresses is selected as the *current address* of the tunnel endpoint. The other addresses are denoted as *standby addresses*, any of which may be configured as the current address of the VPN endpoint when needed.

An IPSec-based VPN tunnel is uniquely identified by the current addresses of the two endpoints of the tunnel. A VPN tunnel carries two uni-directional flows between the endpoints. Each flow is uniquely identified by a *flow label*, an ordered pair (A,B) whose first element A represents the source IP address of a packet flow and whose second element B represents the destination of the flow.

For each VPN tunnel terminating on it, a CE router reserves an available fraction of the access link bandwidth for the IPSec packet flow arriving from the other end of the tunnel. It employs messages defined within IETF's Resource Reservation Protocol (RSVP) [6] standard to convey this bandwidth reservation request to the PE router. All PE routers are assumed to support the RSVP protocol. In the following discussion, we refer to the network shown in Figure 1 to describe how RSVP is used. The RSVP reservation is not an end-to-end reservation between the end points of the flow as is typical for RSVP usage. It only applies to the access link connecting the CE router (say CE Router 1) to the Internet.

The consequence of this access link reservation at the PE router for an IPSec flow is that packets carrying the flow label for that IPSec flow are guaranteed to receive a portion of the access link bandwidth no matter how saturated the link becomes. Conceptually, the CE router (i.e., CE Router 1) uses an RSVP reservation to set up a provisioned virtual access link between its PE router (i.e., PE Router 1) and itself as well a virtual firewall at the PE router to guard this virtual access link. The virtual firewall filters out all traffic flows except that specified by the CE router (i.e., CE Router 1) from using that virtual access line. What this implies is that the only way an attacker (or attackers) can disrupt the IPSec flow over the provisioned virtual link is by emitting packets carrying the same flow label as the VPN flow. Thus, spurious packet floods directed at a CE router with arbitrary source addresses will not be able to flood the virtual access line reserved for the protected VPN flow. The VPN service will only be affected by attack traffic directed at the CE router with a spoofed source address of the VPN flow.

We now describe our approach for protecting the VPN service against such spoofed packet floods. As noted earlier, we assume that attack traffic can only be generated by outsiders with no knowledge of the shared secrets between the CE routers that are used to encrypt and authenticate IPSec encapsulated packets carried by the VPN tunnel. This implies that a spoofed packet arriving at a CE router from an attacker can be detected by its failure to pass authentication checks at the CE router. If the volume of spoofed traffic arriving at the CE router exceeds a certain threshold it is indicative of a flooding attack on the VPN service.

Upon detection of a spoofed packet flooding attack, the CE router invokes a failover (or recovery) mechanism for mitigating the impact of the attack on the VPN service. Conceptually, this failover mechanism reconfigures the source and/or destination addresses of the flow label associated with the victim VPN flow (i.e., the flow under attack). The new source and destination IP addresses for the VPN tunnel are selected from the set of standby addresses associated with the two endpoints of the VPN tunnel and conveyed to the two CE routers terminating the tunnel. Concurrently, the

failover mechanism at the victim CE router uses RSVP to cancel the reservation associated with the old flow and installs a reservation for the new flow for the VPN tunnel. Thus, the IPSec packets over the reconfigured tunnel carry the new label and are accommodated over this newly provisioned virtual link. The attack traffic still carries the old label associated with the VPN tunnel. The virtual firewall installed at the PE router by the new RSVP reservation protects this virtual access link and consequently the VPN traffic from the attack traffic.

For an attacker with no knowledge of the set of standby addresses associated with the two endpoints of a VPN tunnel, the VPN failover process described above appears unpredictable or "random". We refer to this VPN failover approach that reassigns the addresses of the two endpoints of a VPN flow upon detection of an attack as *randomized failover*. The term randomized is used here informally and is not meant to imply randomness in the strict statistical sense.

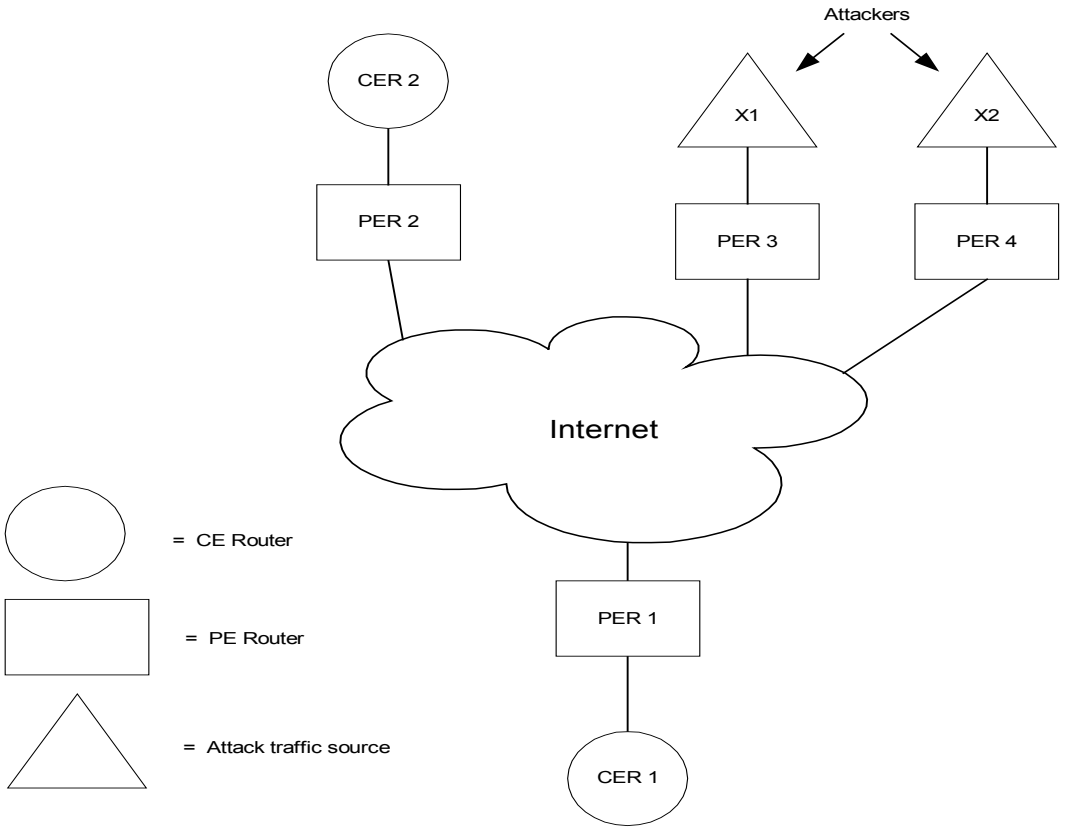

Figure 2: Notional VPN Networking Environment

Let the sets S1 and S2 represent all possible values that that may be assigned to the source and destination address components of a flow labeled (a,b). That is $a \in S_1$ and $b \in S_2$. Let $|S_1|$ and $|S_2|$ represent the cardinality of the sets S_1 and S_2, respectively. From the perspective of an external attacker of the VPN flow labeled (a,b), the new label assigned to the VPN flow by the randomized failover process can take any value from among $|S_1|*|S_2|$ possibilities. We use the term *address space diversity* to refer to the quantity $|S_1|*|S_2|$ that signifies the universe of possible values available to the randomized failover process for reconfiguring a VPN flow label when its flow is under attack.

The premise of the randomized failover approach for survivable VPN services is that, given sufficient address space diversity, it would be extremely difficult for an external attacker

operating with limited time and resources to discern the new label associated with the reconfigured VPN flow and adapt the attack to disrupt the new configuration of the VPN service.

The following paragraphs present two different techniques for implementing the failover mechanism described above for survivable VPN services. This is followed by a comparison of these two techniques. We describe the operation the different techniques for VPN failover with reference to the notional VPN network environment shown in Figure 2. Although an actual deployment of VPN service may contain a number of dispersed customer LAN sites, for simplicity, Figure 2 shows only two customer LAN sites served by the CE routers CER 1 and CER 2. An IPSec-based VPN tunnel between the two CE routers connects these two networks into a single virtual private network. X1 and X2 represent sources of the attack traffic that are connected to the shared service provider network through the PE routers PER 3 and PER 4.

The VPN tunnel between CER1 and CER2 is uniquely identified by the current addresses assigned to the two endpoints of this tunnel by CER1 and CER2. Let A_1 and B_1 be the current addresses of the VPN tunnel endpoints and let A_2, A_3, …A_n and B_2, B_2, ..., B_m be the standby addresses for these endpoints, respectively. As part of the initial set up of the VPN tunnel between CER 1 and CER 2, the two routers exchange the list of standby addresses as well as the address to be used as the current address when the VPN tunnel starts operating. Thus, each CE router maintains the current address and the standby addresses of both ends of the VPN tunnel supported by them.

3.1 VPN Tunnel Reconfiguration with Unicast Addressing

As the name indicates, only unicast IP addresses are used for the VPN endpoints in this approach. During initialization, CER1 uses RSVP to reserve bandwidth on its access link for the IPSec packet flow from CER 2 with the flow label (B_1,A_1). CER 2 does the same for the flow from CER 1 with label (A_1,B_1).

Suppose an attacker directs a spoofed packet flood with the same flow label at CER1. Upon detection of this attack (as described earlier), CER1 selects a new label for the IPSec flow between CER2 and CER1. The new label is derived by replacing one or both components of the current label of the VPN flow (B_1,A_1) with standby addresses maintained by CER1 for both endpoints of the tunnel. The newly selected flow, say (B_3, A_4), is then conveyed to the far end CE router CER 2 using a secure signaling channel between the CE routers.

Subsequently, CER 1 cancels its initial access link reservations for the flows (B_1,A_1) and sets up a reservation for the newly configured flow (B_3,A_4). The attack traffic directed at CER 1 with the old label of the VPN flow, i.e., (B_1,A_1), is filtered out of the provisioned virtual link at PER 1.

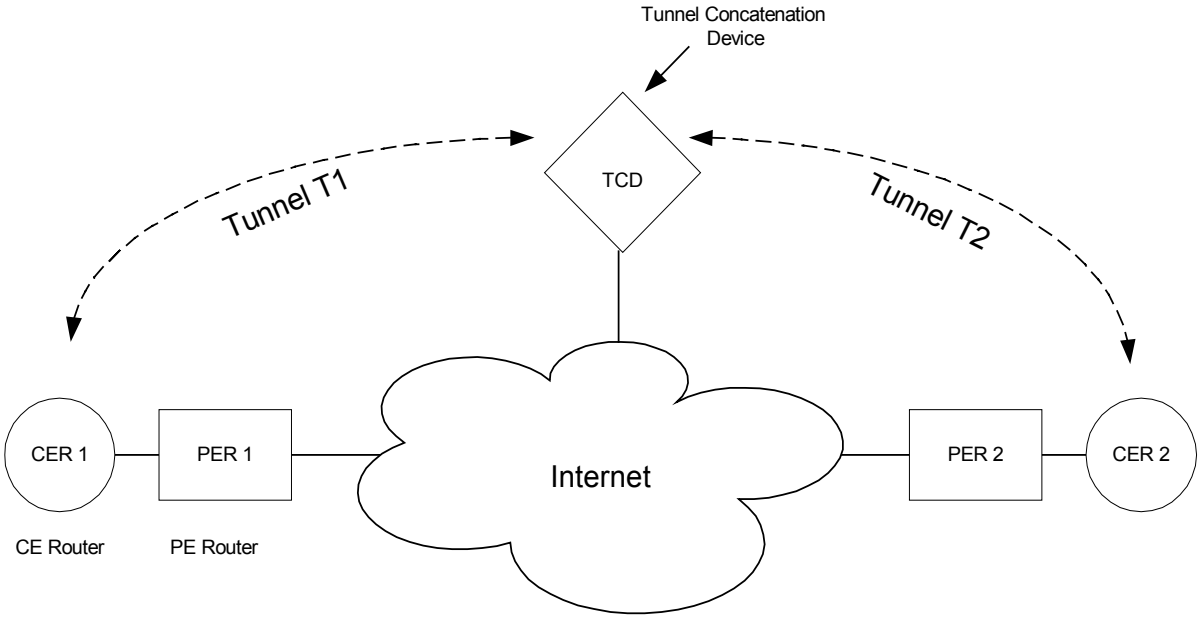

Figure 3: Networking Scenario with VPN Tunnel Splitting

The flow label reconfiguration technique described thus far limits the range of addresses that can be selected by each CE Router for the VPN tunnel end point. Consider an organization that has been allocated 256 Class C IP addresses for its LAN site. Only a subset of these addresses will be available for use by the CE router at that LAN site for use as VPN tunnel endpoint addresses. This limitation in address space diversity limits the degree of protection provided by the survivable VPN service from flooding attacks.

To overcome the problem of limited address space diversity, the technique above can be augmented with a mechanism called *VPN tunnel splitting*. When an attack on a VPN tunnel is detected, this mechanism splits the end-to-end tunnel between the LAN sites into two concatenated tunnels. Thus, instead of a direct IPSec tunnel between CER 1 and CER 2 that carries the VPN traffic, the VPN tunnel is composed of two IPSec tunnels that are concatenated by a *tunnel concatenation device (TCD)* as shown in Figure 3. Consequently, the VPN flow from CER 2 to CER 1 is redirected over an IPSec tunnel (Tunnel 1) to the TCD. The TCD is essentially another CE router potentially owned by a third party with which the VPN customer has an established trust relationship. The TCD de-encapsulates the IPSec packets received from CER 2 and re-encapsulates it in IPSec packets using one of the IP addresses allocated to the TCD as the source address and tunnels the packets over Tunnel 2 to CER 1. Any number of alternate TCDs may be configured between two VPN-enabled LAN sites with appropriate security associations between the TCDs and the end points.

Referring to the VPN networking scenario above, let TCD_1, TCD_2, …., TCD_n be the alternate TCDs configured to support the survivable VPN service between the two LAN sites. Consider normal operating conditions where the VPN connection between CER 1 (with address A_1) and CER 2 (with address B_1) is a direct IPSec tunnel between these two devices, as described previously. That is, using RSVP CER 1 and CER 2 provision bandwidth on their access links for the flows (B_1,A_1) and (A_1, B_1), respectively.

Consider the attack scenario above, where spoofed attack traffic with label (B_1, A_1) is directed at CER 1 to flood the access link associated with that LAN site. Upon detection of the attack, CER 1

selects one of the TCDs configured for the VPN service, say TCD_1, as the tunnel concatenation point for the IPSec flow between it and CER 2. It also selects one of the IP addresses assigned to TCD 1, say α_1, for the tunnel concatenation service. It then securely notifies CER 2 to begin operating in the split tunnel mode and provides it the IP address α_1 for the tunnel concatenation service. Subsequently, CER 2 tunnels all VPN flows for LAN site 1 to TCD_1 (at its IP address α_1). The latter redirects this traffic over an IPSec tunnel between it and CER 1 (Tunnel 2 in the figure). This traffic flow has the label (α_1, A_1). Note that this IPSec flow now carries the stream of VPN packets between the LAN sites that was carried by the direct tunnel, i.e., (B_1,A_1) previously.

In addition to initiating actions to reconfigure the IPSec tunnel, CER 1 cancels its existing RSVP reservation for the flow (B_1,A_1) on its access link. It now provisions bandwidth on the access link for the redirected IPSec arriving from TCD 1, i.e., the flow with label (α_1, A_1). The attack traffic that was directed at CER 1 with the spoofed label (B_1,A_1) is filtered of this newly provisioned virtual link between PE router 1 and CER 1 that has been established for the VPN traffic between the two LAN sites.

Conceptually, tunnel splitting may be viewed as facilitating the reconfiguration of the label of the flow from LAN site 2 to LAN site 1 without limiting the address space diversity that is available for performing this reconfiguration. The address space diversity with tunnel splitting is 2^{32}*(size of the destination address space). Thus, it greatly increases the survivability of the VPN service compared to label reconfiguration with direct tunnels. However, this comes at the price of additional per packet overhead incurred by the tunnel concatenation service in de-encapsulating and re-encapsulating the packets at the TCD. Also, redundant hardware in the form of one or more TCDs are needed.

To prevent the TCDs from being exploited by potential attackers to amplify or reflect attack traffic, each IPSec packet arriving at the TCD is authenticated to verify that it originated at a source that is authorized to use this TCD. Packets that fail this authentication and authorization check are discarded by the TCD.

To enable continued operation of the VPN service between the remote LAN sites in spite of a disruption of an active TCD's operation (either because of a benign hardware failure or an intrusion-induced degradation), CER 2 and CER 1 periodically exchange *VPN heartbeat messages*. Each VPN heartbeat message carries with it the sequence number of the last IPSec packet that was transmitted by the CE router sending the heartbeat message. Using this information as well as the sequence number field of the received IPSec encapsulated packets, a CE router continually tracks the packet loss rate between consecutively received heartbeat messages. The loss or unacceptable degradation of an existing tunnel concatenation service is detected by the CE routers by the occurrence of one of the following events: (1) failure to receive VPN heartbeat messages over some period of time (specified at system configuration); (2) observed packet loss rates over a specified threshold.

Upon detection of a degradation or loss of the tunnel concatenation service for an existing split tunnel between the LAN sites, the CE router reconfigures the split tunnel. It does this by selecting an alternate TCD from the list of TCDs maintained by it for the VPN service, choosing an IP address from the candidate addresses for this TCD, and notifying the far end CE router of the address of the new TCD. Also, it cancels its RSVP reservation for tunnel from the existing TCD and provisions bandwidth for the new tunnel from the selected TCD.

3.2 VPN Tunnel Reconfiguration with Multicast Addressing

In this case, each of the two unidirectional IPSec encapsulated packet flows within a VPN tunnel uses a multicast IP address as the destination of the flow and a unicast address as the source IP address. Each CE router maintains a list of alternate multicast IP addresses to assign to IPSec flows terminating on it and a set of alternate unicast IP addresses for IPSec flows originating from it.

The tunnel reconfiguration technique described here relies on the use of *source specific multicast (SSM)* [7]. This is an extension of the traditional IP multicast service defined by IETF RFC 1112 [8]. SSM is currently supported within some commercially available IP routers. The service provided by SSM is a "channel" that is uniquely identified by the SSM address M and a source IP address S. A range of IP multicast addresses, i.e., 232.0.0.0 to 232.255.255.255, has been reserved by the IANA for use by this service. A source S transmits IP datagrams to a destination M. To receive these datagrams a receiver must subscribe to the channel (S,M). Version 3 of IGMP supports mechanisms for such channel subscriptions by a receiver.

The operation of this technique is now described with reference to the notional networking scenario of Figure 2. Let MA_1, MA_2, MA_2, …., MA_n and A_1, A_2, A_3, …., An be the sets of alternate SSM multicast and unicast IP addresses maintained by CER 1 and let MB_1, MB_2, MB_3, ……., MB_m and B_1, B_2, B_3, …., B_m be the SSM multicast and unicast addresses at CER 2. Suppose the VPN tunnel starts operation using the labels (A_1, MB_1) and (B_1, MA_1) for the two unidirectional flows between CER 1 and CER 2. In this case, these flows are carried on the SSM channels (A_1, MB_1) and (B_1,MA_1) respectively. Thus the flow labels also identify the SSM channels.

In setting up the survivable tunnel between the two LAN sites, CER 1 and CER 2 subscribe to the SSM channels (B_1,MA_1) and (A_1,MB_1), respectively. Using RSVP, CER1 provisions bandwidth on its access link for the flow (B_1,MA_1) and CER 2 does the same for the flow (A_1,MB_2).

Suppose the spoofed attack traffic originating from X1 and/or X2 is directed at CER 1. That is, the attack traffic carries the label (B_1,MA_1) which is currently assigned to the flow from CER 2 to CER 1. Upon detecting the attack (using the mechanism described earlier), CER 1 chooses a new label for the flow by selecting an alternate address for either or both of the components of the original flow label. It then unsubscribes from the SSM channel (B_1,MA_1) and subcribes to the channel associated with the newly configured flow label, say (B_3, MA_1). Also, CER 1 cancels its RSVP reservation for the flow (B_1, MA_1) and makes a reservation for the newly configured flow, i.e., (B_3, MA_1).

When CER 1 cancels its subscription to the SSM channel (B_1,MA_1), the multicast routing protocol implementing the SSM service prunes the multicast tree to remove all branches that do not have subscribers under them. This pruning process results in the attack traffic originating at X1 (and X2), directed at CER 1, to be filtered out at PER 3 (and PER 4). Thus, this organic technique enables the system to automatically squelch the spoofed packet flood close to the source of the attack traffic. The spoofed packet flood thus gets filtered out before it even enters the service provider backbone network. Any attack traffic directed at the unicast addresses of CER 1 however are still handled by the virtual firewall at PER 1 that protects the provisioned virtual link for the VPN service.

The address space diversity of this technique is 2^{24}* (size of unicast address space of source). The unicast address space for the source is determined by the number of addresses allocated to the

source LAN site by the service provider. The limitation on the size of the unicast address space of the source can be overcome by augmenting the technique described above with the tunnel splitting mechanism described earlier. With tunnel splitting, the direct tunnel between CER 2 and CER 1 is split into two tunnels that are concatenated by a TCD. The tunnel between the source of the flow (CER 2) and the TCD uses unicast addresses for both end points. The second tunnel for the flow between the TCD and the destination (CER 1) uses an SSM multicast address for the destination. Tunnel splitting increases the size of the unicast address space of the source to that of the unicast address space of the Internet (i.e., approximately 3.8 billion addresses). The address space diversity with this technique is therefore approximately $63*10^{15}$.

The technique described above for tunnel reconfiguration using SSM multicast address can also be adapted for use with ordinary IP multicast addresses. In this case, the destination multicast IP address component of a flow must be changed during tunnel reconfiguration if attack traffic is to be filtered out close to the source. If only the source IP address component of the flow label is changed to accomplish tunnel reconfiguration, attack traffic filtering occurs at the PE router of the victim network as in the case of tunnel reconfiguration with unicast addressing described in Section 3.1.

4. Prototype Implementation

We built a prototype implementation of a survivable VPN service using tunnel reconfiguration with unicast addressing. For testing and demonstrating the implementation, we configured a network testbed consisting of two private subnets (i.e., 10.10.10.x and 10.10.20.x) connected together by a VPN tunnel over a backbone LAN simulating the Internet cloud, as shown in Figure 4. A simulated T1 link implemented over a point-to-point Ethernet connection implements the access link connecting the CE router of each private subnet to the PE router on the backbone LAN. Attached to the backbone LAN is an attack tool that can generate packet floods directed at a given IP address. The source address of the packets generated by the tool can be configured by the user.

The 10.10.10.x and 10.10.20.x networks are configured with a video streaming server and a video player from Real Networks, respectively. The video traffic between the two private networks is thus carried over the IPSec tunnel between the CE routers of these networks.

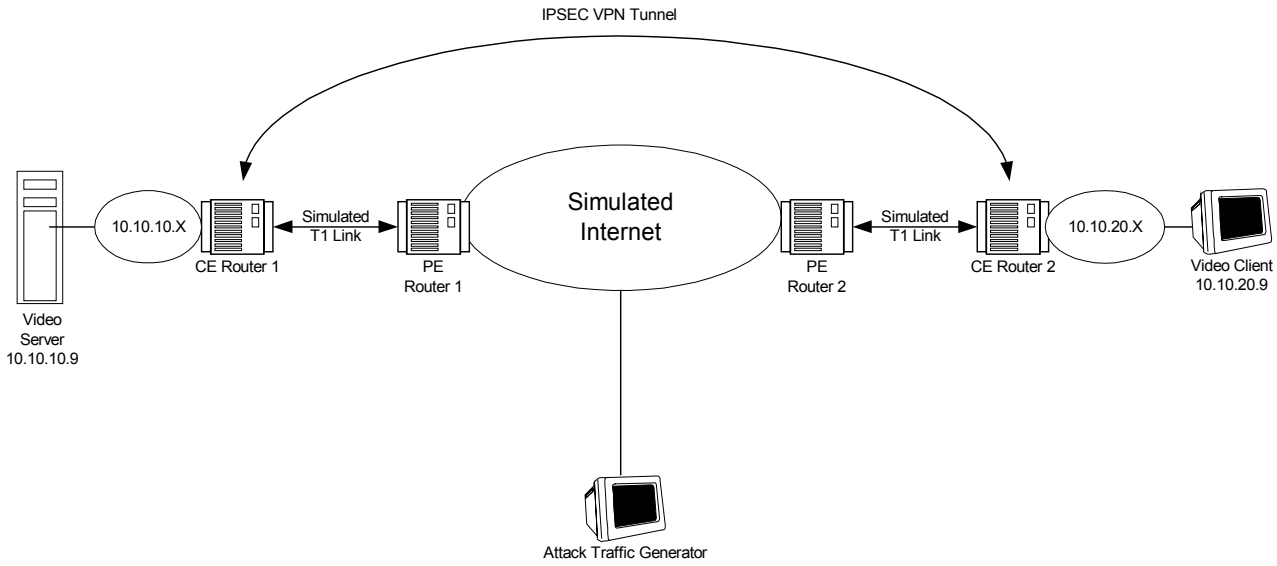

Figure 4: Demonstration Testbed Configuration

To demonstrate the operation of the survivable VPN service, we used the attack tool to generate packet flood directed at the CE Router 2. The packets in the attack traffic were configured to spoof the current source address of the IPSec encapsulated packets from CE Router 1 to CE Router 2. This simulates a packet flooding attack targeted at the access link connecting the 10.10.20.x private network (i.e., the network with the video client) to the simulated Internet. The packet arrival rates at the video player were monitored using a tool that is integrated within the Real Networks player.

Figures 5 and 6 depict the impact on the flooding attack on the video stream without and with the VPN survivability techniques, respectively. Without the survivability techniques, the packet arrival rate at the video client dropped by over 50% following the onset of the attack. The packet loss rate encountered under these conditions precluded the video client from playing the received video stream. In contrast, as shown in Figure 7, the attack traffic had no deleterious impact on the packet arrival rates at the video client when the survivable VPN services were used. We notice a small downward spike representing a momentary drop on the packet arrival rate following the onset of the attack. However, this had no adverse impact on the observed video playback quality.

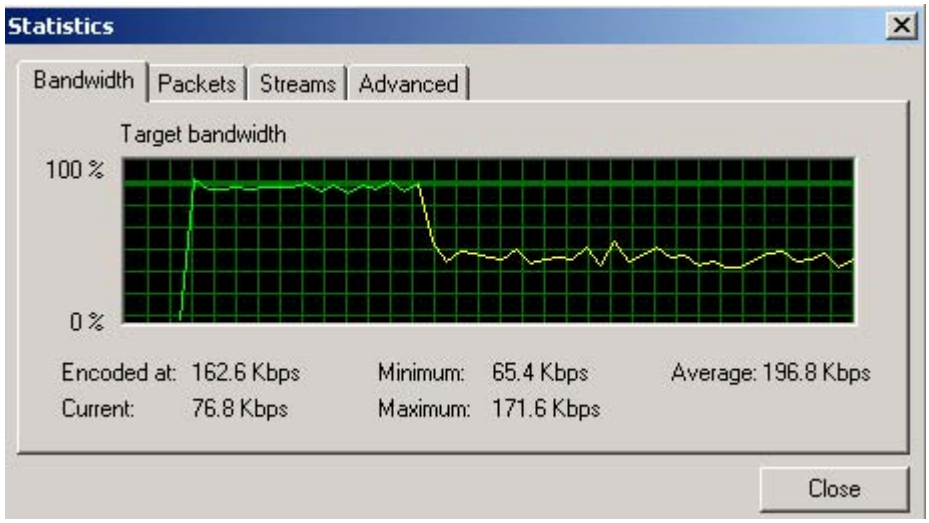

Figure 5: Impact of Flooding Attack on Packet Arrival Rate at Video Player using Traditional VPN Service

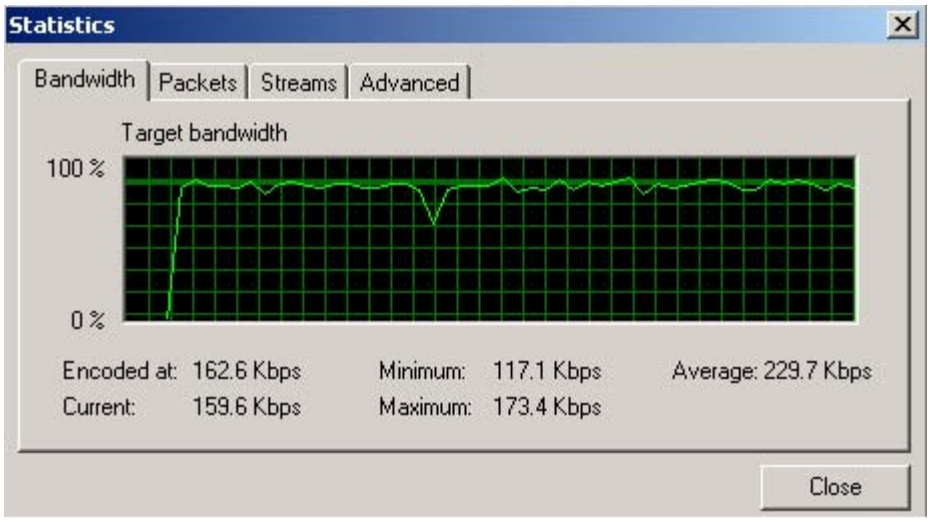

Figure 6: Impact of Flooding Attack on Packet Arrival Rate at Video Player using Survivable VPN Service

5. Related Work

Existing work on approaches for dealing with DoS flooding attacks primarily focus on one of the following two areas: *fail-soft* mechanisms and *traceback* mechanisms. Fail-soft mechanisms are aimed at mitigating the deleterious effects of flooding attacks on the victim hosts [9,11,15]. They are designed to enable a victim host to provide at least a minimal level of acceptable service in the face of flooding attacks targeted at it. As noted by Savage *et al* [13], this approach serves at best as a stopgap measure. It does not eliminate the problem to allow the victim host to resume providing full service nor does it serve as a deterrent to attackers. More importantly, these fail-soft mechanisms focus on the end hosts only and do not address the problem of attacks that are designed to disrupt the operation of entire edge networks by inundating the link connecting the edge network to the shared IP backbone infrastructure.

Traceback mechanisms [10,13,14,16] are infrastructure-based mechanisms that attempt to trace the attack traffic towards their origin. This is motivated by the fact that the effectiveness of

techniques such as packet filtering would be greatly enhanced by applying them as close to the generation points of the traffic flood. Traceback mechanisms, however, provide no means for a network to easily differentiate spurious packets from legitimate packets originated by a spoofed source so that selective filtering of traffic can be performed. Consequently, all packets from a source IP address destined for the victim are filtered out, including legitimate traffic. Furthermore, for high-availability mission-critical network applications that cannot tolerate disruptions of network service beyond a few seconds, the latency associated with network recovery using traceback mechanisms may be unacceptable. For large-scale distributed DoS flooding attacks, the amount of time taken by traceback mechanisms to trace all the distributed attack sources and squelch the spurious traffic may be in the order of minutes to hours, even assuming that the entire traceback and network response process is automated. Thus, traceback mechanisms by themselves are insufficient for meeting the needs of such network applications, such as VPN services. Traceback must be augmented by other mechanisms, such as those we present, that provide immediate restoration of the network services upon detection of an access link flooding attack.

6. Conclusions

This paper presented an approach for building a survivable inter-site VPN service that employs a set of organic techniques to protect VPN flows from access link flooding attacks. The service is designed to ensure continued, uninterrupted operation of VPNs in spite of packet flooding attacks. The key ideas underlying the organic VPN survivability techniques presented here are the use of a truncated RSVP reservation to provision a virtual access link for a VPN and the use of a randomized failover process for reconfiguring a VPN tunnel when it is under attack. Alternative implementation techniques for tunnel reconfiguration were also presented.

In order to launch a successful DoS attack on a VPN service, the attacker needs to know the current configuration of the VPN service. That is, the attacker needs to know the globally reachable IP addresses of one or both of the endpoints of the VPN tunnel(s) to which spoofed packets can be sent. In the VPN survivability approach presented here, the randomized failover mechanism that is invoked by the VPN service upon detection of a flooding attack reconfigures the VPN tunnel. The intent is to render the attack that was directed upon the old configuration of the VPN tunnel ineffective through this reconfiguration.

In order for packet flooding DoS attacks to be successful in such a dynamically reconfigurable VPN service, the attacker needs to be able to accurately track VPN tunnel configurations as they change and adapt the attack accordingly. By implementing sufficient address space diversity through tunnel splitting techniques, the survivable VPN service makes it extremely difficult for an attacker to determine the new configuration of the VPN tunnel. Of course, given sufficient time and unlimited resources, an attacker might potentially be able to deduce this information. However, the goal of the organic survivability mechanisms is to deter the attacker long enough to allow other complementary techniques such as traceback to pin-point the source of the attack so that actions can be taken to neutralize the attacker before further damage is inflicted on the system.

References

[1] S. Kent, and R.Atkinson, “Security Architecture for the Internet Protocol,” IETF RFC 2401, October 1998.

[2] K. Hamzeh, et al., “Point-to-Point Tunneling Protocol,” IETF RFC 2637, July 1999.

[3] W. Townsley, et al., “Layer Two Tunneling Protocol (L2TP),” IETF RFC 2661, August 1999.

[4] K.J. Houle, and G.M. Weaver, "Trends in Denial of Service Attack Technology," Technical Report, CERT Coordination Center, October 2001.

[5] D. Moore, G.M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," Proceedings of the 2001 USENIX Security Symposium, Washington, D.C., August 2001.

[6] R. Braden, et al., "Resource Reservation Protocol (RSVP)," IETF RFC 2205, September 1997.

[7] S. Bhattachayya, et al., "An Overview of Source-Specific Multicast (SSM) Deployment," IETF Internet Draft: draft-ietf-ssm-overview-01.txt (work in progress), August 2001.

[8] S. Deering, "Host Extensions for IP Multicasting," IETF RFC 1112, August 1989.

[9] G. Banga, P. Druschel, and J. Mogul, "Resource Containers: A New Facility for Resource Management in Server Systems," in Proceedings of the 1999 USENIX/ACM Symp. on Operating System Design and Implementation, Feb. 1999.

[10] S.M. Bellovin, "ICMP Traceback Messages," Internet Draft:draft-bellovin-itrace-00.txt, Mar. 2000.

[11] Cisco Systems, Configuring TCP Intercept (Prevent Denial-of-Service Attacks), Cisco IOS Documentation, Dec. 1997.

[12] D. Dittrich, "Distributed Denial of Service (DDoS) Attacks/tools," http://staff.washington.edu/dittrich/misc/ddos/.

[13] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," IEEE/ACM Trans. on Networking, vol. 9, no. 3, June 2001.

[14] D. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," Proceedings of IEEE INFOCOM 2001, Anchorage, AK, Apr. 2001.

[15] O. Spatscheck and L. Peterson, "Defending Against Denial of Service Attacks in Scout," Proceedings of the 1999 USENIX/ACM Symp. on Operating System Design and Implementation, Feb. 1999.

[16] R. Stone, "CenterTrack: AN IP Overlay Network for Tracking DoS Floods," Proceedings of the 2000 USENIX Security Symposium, Dec. 2000.