

Protecting Electronic Commerce From Distributed Denial-of-Service Attacks

José Carlos Brustoloni
Networking Software Research Department
Bell Laboratories, Lucent Technologies
101 Crawfords Corner Rd., Holmdel, NJ 07733 — USA
jcb@dnrc.bell-labs.com

ABSTRACT

It is widely recognized that distributed denial-of-service (DDoS) attacks can disrupt electronic commerce and cause large revenue losses. However, effective defenses continue to be mostly unavailable. We describe and evaluate VIPnet, a novel value-added network service for protecting e-commerce and other transaction-based sites from DDoS attacks. In VIPnet, e-merchants pay Internet Service Providers (ISPs) to carry the packets of the e-merchants' best clients (called VIPs) in a privileged class of service (CoS), protected from congestion, whether malicious or not, in the regular CoS. VIPnet rewards VIPs with not only better quality of service, but also greater availability. Because VIP rights are client- and server-specific, cannot be forged, are usage-limited, and are only replenished after successful client transactions (e.g., purchases), it is impractical for attackers to mount and sustain DDoS attacks against an e-merchant's VIPs. VIPnet can be deployed incrementally and does not require universal adoption. Experiments demonstrate VIPnet's benefits.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General—*security and protection*; C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*packet networks*; J.1 [Computer Applications]: Administrative Data Processing—*business, financial*

General Terms

Security, Reliability, Performance, Algorithms, Experimentation

Keywords

Denial of Service, Quality of Service, Electronic Commerce

1. INTRODUCTION

In a denial-of-service (DoS) attack, a malicious client (called the *attacker*) performs operations designed to partially or completely prevent legitimate clients from gaining service from a server (called the *victim*).

DoS attacks are common and can cause significant losses. Measurements by CAIDA/UCSD detected more than 12,000 attacks against more than 5,000 victims during a 3-week

study in February of 2001 [22]. In a CSI/FBI study in March of 2001, 38% of the security professionals surveyed declared that their sites had been the object of at least one DoS attack in the previous year [15]. Well-known e-merchants, including Amazon, buy.com, E*Trade, and eBay, are among recent victims. DoS attacks can harm e-merchants in two ways. First, when an e-merchant cannot serve its clients, the e-merchant loses advertising and sales revenues. Second, the e-merchant's clients, advertisers, and investors are frustrated and may therefore seek competing alternatives.

Among DoS attacks, *congestive* ones are the most difficult to defend against. In a congestive attack, an attacker sends to a victim packets that exhaust the network's or the victim's resources, making the victim unable to receive, process, or respond to legitimate requests. Such attacks are easily enabled by the Internet, whose service model is non-authenticated, connectionless, and best-effort. Existing defenses against such attacks are weak and not widely deployed.

This paper contributes a novel DoS defense architecture that robustly and scalably limits the effects of congestive attacks against e-merchants, is consistent with existing Internet design principles, and compensates Internet Service Providers (ISPs) for the necessary investment.

Our solution, *VIPnet*, allows an e-merchant to request an ISP to carry the packets of certain clients in an elite class of service (CoS), called *VIP*. An e-merchant may grant VIP rights, e.g., to those clients that bring in a majority of the e-merchant's revenues. For this service, the e-merchant pays the ISP a fee. Quality of service (QoS) mechanisms commonly found in routers, such as differentiated services (diff-serv) [1], priority-based, or weighted fair queueing (WFQ), distinguish the VIP CoS from the regular best-effort CoS, which is used for other packets. Because VIP traffic is carried in its own CoS, it is insulated from congestion and DoS attacks that may occur in the regular CoS.

VIP rights are *term-* and *usage-limited*, i.e., each VIP right expires after a certain time or after the respective client has sent a certain amount of data using it. To obtain a new VIP right, a client must perform transactions (e.g., purchases) of sufficient value. Therefore, no host (with or without VIP rights, compromised or not) can sustain indefinitely a congestive DoS attack against an e-merchant's VIP CoS (unlike the regular CoS).

VIPnet is effective even if deployed only by select ISPs. ISPs that serve clients that have been granted VIP rights need to install a device called *VIP Gate (VIPG)* in the

ISP's *access gateways*, i.e., those nodes that terminate ISP customer layer-2 links and authenticated tunnels. VIPGs monitor packets coming in from access links, mark for the VIP CoS those packets whose destination is a VIPnet e-merchant and whose source has an active VIP right for the destination (or vice-versa), and mark for the regular CoS any other packets. Each VIPG also locally maintains a list of VIP rights, authenticates clients, and allows clients to activate or deactivate the respective VIP rights. On the other hand, intermediate ISPs on the paths between clients with VIP rights and the e-merchants who granted them need to support the VIP CoS, but need not deploy VIPGs. Finally, the rest of the Internet need not support or be aware of VIPnet at all. In peering points between ISPs that do and do not support VIPnet, a VIPnet ISP simply maps from *VIP* to *regular* the CoS markings (if any) in packets arriving from non-VIPnet ISPs.

Because VIPnet generates new ISP revenue streams, many ISPs may decide to support it. An intermediate ISP's compensation for carrying VIP packets received from another ISP can be negotiated between the ISPs much like any other form of peering. The mechanisms used by various ISPs to differentiate the VIP and regular CoS need not be the same. Furthermore, as long as work-conserving scheduling mechanisms are used, such as the ones cited above, VIPnet support does not reduce a network's total capacity, even if no VIP traffic is present. Resources that would be used for the VIP CoS, if VIP traffic were present, can be automatically used for the regular CoS, when VIP traffic is absent.

VIPnet allows clients to activate VIP rights and break through congestion even if the e-merchant is currently under a DoS attack. VIPnet ISPs carry requests to activate VIP rights in the regular CoS, but process them as close to the respective client as possible. Therefore, activation requests are not affected by DoS attacks against other parts of the respective client's ISP, Internet, or e-merchant.

For access from home or office, VIP rights are *location-specific*, i.e., valid only in a given ISP's point of presence (PoP). *Location-independent* VIP rights may be used for mobile access. A given client may have multiple VIP rights for the same e-merchant, one for each ISP or location.

The rest of this paper is organized as follows. Section 2 reviews some of the more common techniques used in DoS attacks and previously proposed defenses against them. VIPnet is described in detail in Section 3. Section 4 discusses VIPnet's advantages and limitations. Empirical results of a prototype implementation are presented in Section 5. Finally, Section 6 summarizes and concludes.

2. PREVIOUS TECHNIQUES FOR DOS ATTACK AND DEFENSE

This section summarizes techniques commonly used in DoS attacks and defenses proposed against them.

Some DoS attacks can be prevented by proper system administration [12, 16]. These include physical or remote *takeover* attacks and *death-pill* attacks. In a physical takeover attack, the attacker gains physical access to components of the ISP or e-merchant infrastructure (e.g., one or more links, routers, or servers) and compromises their functionality. In a remote takeover attack, the attacker exploits some bug in the infrastructure's software so as to gain privileged access and thus be able to modify the software remotely.

In a death-pill attack (e.g., *land* [8], *teardrop* [8], or *ping of death* [7]) the attacker sends one or a few packets to an infrastructure component (e.g., router or server) known to contain a bug, such that the packets cause the component to crash. Proper ISP and e-merchant physical security can eliminate physical takeover attacks. Likewise, prompt installation of patches or updates that fix software bugs can prevent future remote takeover or death-pill attacks exploiting those bugs.

On the contrary, congestive DoS attacks cannot be similarly prevented. In a congestive attack, an attacker floods a server with so many packets that the server is unable to respond to requests sent by legitimate clients. Four factors make it difficult to defend against congestive attacks. First, any host connected to the Internet can be used to sustain a congestive attack against any victim also connected to the Internet. By design, the Internet will forward packets from any host to any other host on a best-effort basis, without bounding packet rate or volume. Second, there are many hosts (e.g., in homes and universities) that are connected to the Internet and do not have the benefit of proper system administration. Such hosts often contain bugs or are configured in such a way that attackers can, without authorization, use them as *agents*, i.e., as the hosts that actually send attack packets to a victim. Agents provide *cloaking* and *leverage* to an attacker, i.e., respectively, hide the attacker's identity and multiply the attacker's resources (e.g., bandwidth). Third, attackers can *spoof* attack packets, i.e., falsify the packets' source addresses. Spoofing is possible because the Internet does not validate source addresses. Spoofing further enhances an attacker's cloaking. Finally, automated tools of increasing sophistication for mounting DoS attacks can be easily downloaded from the Web. Current examples of such tools include *stacheldraht* [18], *TFN2K* [10], and *mstream* [11]. Using such tools, even unskilled teenagers can mount successful attacks.

The two currently most popular DoS attack techniques, *smurf* [9, 20] and *TCP SYN flooding* [6], are both congestive. In a smurf attack, the attacker sends ICMP echo requests to a network's broadcast address. The attacker spoofs the requests with the victim's address. Therefore, each host in the network sends a reply not to the attacker but to the victim, thus unwittingly becoming an agent of the attack. In a TCP SYN flooding attack, the attacker or its agents send spoofed TCP SYN (i.e., connection request) packets to the victim. Each such bogus request causes the victim to unfruitfully tie up resources that could otherwise be used for requests from legitimate clients.

To prevent smurf attacks, the Internet Engineering Task Force (IETF) has changed the default treatment of directed broadcast packets by routers [26]. Instead of accepting and forwarding directed broadcast packets, routers should now by default drop them. Additionally, to thwart spoofing, the IETF now recommends *ingress filtering* [19], i.e., ISP ingress routers should drop a packet that arrives in a port if the packet's source address does not match a prefix associated with the port. Unfortunately, the IETF's recommendations need to be adopted by parties (networks unwittingly used in smurf attacks and ISPs) that are thereby burdened with new responsibilities and costs, but receive no compensation for solving what they may consider somebody else's (the e-merchants') problem. Moreover, these recommendations do not deter all possible congestive DoS attacks. Even without

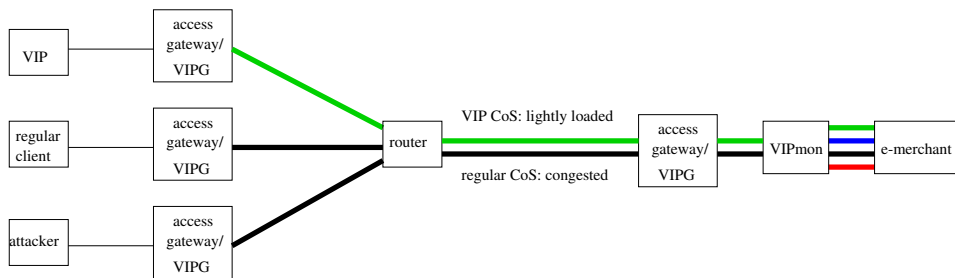


Figure 1: VIPnet allows an e-merchant to have the packets of the e-merchant’s best clients (VIPs) carried in an elite class of service. Thus, VIPs and a major part of the e-merchant’s revenues are insulated from congestion (whether malicious or not) in the regular best-effort class.

spoofing and directed broadcast, attackers can use agents to obtain the cloaking and leverage necessary for successful attacks. Therefore, adoption of these recommendations (particularly ingress filtering) has been spotty.

IP traceback [24] is a recently proposed alternative to ingress filtering. Unlike ingress filtering, IP traceback can be effective even if not widely deployed. IP traceback modifies routers so that they probabilistically send traceback information to a packet’s destination. Statistical methods allow a victim to use such information to partly reconstruct the attack path (the reconstructed part is that closest to the victim). However, IP traceback has weaknesses that may conspire against its adoption. It appears that attackers can easily defeat IP traceback by making attacks *oblique*, i.e., by ostensibly targeting *neighbors* of the victim, rather than the victim itself. Moreover, traceback information sent by routers that are further from the victim than is the closest attacker can be spoofed and therefore needs authentication. The infrastructure necessary for such authentication may add considerable complexity and vulnerabilities of its own. Finally, like ingress filtering, traceback does not stop attackers from using agents, and may increase ISP responsibilities and costs without contributing to ISP revenues.

Congestive DoS attacks can be resolved by combining *input logging* and *rate limiting* [14].¹ To use these techniques, the victim must initially determine the *signature* of the attack, i.e., how the attack packets differ from legitimate packets. ISP personnel then install a filter matching the attack’s signature in the egress port of the router closest to the victim. The filter generates a log that reveals what ingress port the attack is coming from. Input logging is then iterated for the next upstream router, until the router closest to the origin of the attack is found. A rate-limiting filter matching the attack’s signature is then left installed in the ingress port from where the attack is coming.

Input logging and rate limiting have many limitations. First, attackers may perform an oblique attack, i.e. obfuscate the attack by ostensibly targeting a neighbor of the intended victim. Thus, the victim may not have the opportunity to examine attack packets. Second, even if attack packets reach the victim, the signature may be difficult to characterize. For example, an attacker may coordinate agents so that they send endless streams of seemingly legitimate

but fruitless requests to the victim, so as to crowd out requests from legitimate clients. Unlike smurf and TCP SYN flooding attacks, such *crowding* attacks do not cause easily identifiable anomalies at the network or transport layer, and therefore may be difficult to filter in routers. Third, filtering, logging, and rate limiting may not be available or may prohibitively slow down many routers, especially in the network core. Fourth, rate limiting may be unable to distinguish malicious and legitimate packets (e.g., TCP SYN packets) that arrive in the same ingress port. Thus, rate limiting may be ineffective if the attack is *evenly distributed* among ingress ports. Finally, input logging and rate limiting are often labor-intensive, tedious procedures performed under pressure and usually without adequate compensation to the ISP.

3. VIPNET

This section describes VIPnet, a new service that can limit the losses inflicted by congestion, whether legitimate or not. VIPnet allows an e-merchant to request that an ISP carry packets of a designated subset of the e-merchant’s clients in a class of service (CoS) that is privileged with respect to another CoS that is used to carry the packets of the e-merchant’s other clients. Members of the designated privileged subset are called *VIPs*, whereas other clients are called *regular* clients, as illustrated in Figure 1. An e-merchant may select its VIPs, e.g., among those clients that bring in a majority of the e-merchant’s revenues. An e-merchant turns a regular client into a VIP by granting it a *VIP right*. Devices called *VIP Gates* (VIPGs) monitor packets and mark for the VIP CoS those packets whose source has an active VIP right issued by the packet’s destination (or vice-versa). Quality of service (QoS) mechanisms protect such VIP packets from overload in the regular best-effort CoS. An optional device called *VIP Monitor* (VIPmon) may be used at an e-merchant for session admission control and dynamic prioritization. Clients, e-merchants, and ISPs use a *VIP protocol* to request, install, activate, and bill VIP rights.

The following subsections describe in greater detail VIP rights, VIPGs, VIPmon, QoS requirements, and the VIP protocol.

3.1 VIP rights

This subsection describes the main attributes of VIP rights.

In greater detail, a *regular* client becomes a *selected* client when an e-merchant grants a *VIP right* to the client. A selected client becomes a VIP when the selected client *acti-*

¹Another strategy that still works surprisingly often is to change the victim’s address in case of an attack. Of course, this solution is not robust against attackers that periodically check the victim’s current DNS mapping.

vates the respective VIP right. A VIP becomes a selected client when the VIP *deactivates* the respective VIP right. A VIP or selected client becomes a regular client when the respective VIP right expires or is revoked by the e-merchant.

The judgment of what clients should have VIP privileges for sending packets to a given e-merchant can be made only by that e-merchant, and not by a client. Therefore, a VIP right is always *client-specific*, i.e., a client cannot transfer to another client a VIP right. Because an e-merchant may not consider another e-merchant's VIPs particularly worthy, a VIP right is always *e-merchant-specific*, i.e., can be used to carry packets in a privileged CoS only to the e-merchant that granted the VIP right.

An e-merchant grants or revokes a VIP right by sending to an ISP a request that the ISP respectively insert the VIP right into or remove the VIP right from the ISP's *VIP list*. Thus, a VIP right is always *ISP-specific*. An e-merchant may grant VIP rights for the same client at different ISPs or at different points of presence (PoPs) of the same ISP. Typically, an e-merchant will grant a client no more than a few VIP rights, e.g., one that the client may use at home, another for use at work, and perhaps a third for use while traveling.

Every VIP right is *term-limited*, i.e., expires at a certain time. This term limit reflects the frequency with which the e-merchant ranks clients for selection of VIPs. Every VIP right is also *usage-limited*, i.e., expires when the amount of information (e.g., number of packets or bytes) the client has sent using it reaches a specified limit. This usage limit is calculated to allow a VIP to perform, as a VIP, sufficient new transactions to remain a VIP.

A VIP right may also be *location-specific*, i.e., usable only in a certain PoP of an ISP. Location-specific VIP rights may be used, e.g., for access from home or from work. A location-specific VIP right allows the respective activation, deactivation, and usage accounting to be processed entirely locally at the PoP. This local processing provides straightforward scalability and robustness with respect to DoS attacks against most of the ISP's infrastructure. Alternatively, a VIP right may be *location-independent*, i.e. usable in any of the ISP's PoPs. A location-independent VIP right may be used, e.g., for mobile access. To activate, deactivate, read, or update the usage accounting of a location-independent VIP right, a VIPG needs to communicate with a remote database server. Because these operations cannot be performed locally, they are more susceptible to DoS attacks than is the case for location-specific VIP rights. Thus, location-independent VIP rights require more sophisticated distributed database processing techniques for high scalability and robustness.

3.2 VIP Gates

This subsection describes VIP Gates (VIPGs), the devices that put VIP rights into action.

Each VIPG maintains a VIP list and a Web site where clients can activate and deactivate the respective VIP rights. The VIP list contains the VIPG's location-specific VIP rights and replicas of location-independent VIP rights activated by clients connected to the VIPG. A VIPG dynamically binds a client with an address when the client activates a VIP right (i.e., client addresses need not be fixed). When a client activates a location-independent VIP right, the VIPG locks and reads the respective records from a remote database.

Conversely, when the client deactivates the right, the VIPG updates and unlocks the respective records in the remote database.

VIPGs monitor packets coming in from access links, mark for transmission in the VIP CoS those packets whose source is a VIP with respect to the packet's destination (or vice-versa), mark for the regular CoS all other packets, and maintain VIP rights' usage information. To determine whether a packet's destination has an active VIP right issued by the packet's source, a VIPG caches whether packets recently sent from the destination to the source had VIP CoS markings.

VIPGs are preferably implemented in an ISP's access gateways. This allows an ISP to separate VIP and regular traffic as early as possible. VIPGs can also be implemented downstream from access gateways, e.g. as stand-alone devices or integrated with diffserv edge routers. However, downstream implementation may leave VIP packets vulnerable to congestion caused by regular packets between an access gateway and the corresponding downstream VIPG.

IP address spoofing may in some cases allow regular clients to pose as VIPs. Therefore, mutually authenticated tunnels (e.g. using IPsec) between client and VIPG may be desirable in the following circumstances: (a) client is in an ISP customer's network that has many IP addresses (e.g. at a company or university); (b) access to the ISP is via a shared medium with layer-2 authentication that is either non-existent or deemed insecure (e.g., WEP in 802.11b wireless networks); or (c) VIPG is implemented downstream from an access gateway that does not implement ingress filtering [19].

3.3 VIP Monitor

This subsection describes the VIP Monitor (VIPmon), a device that may control and more finely prioritize client sessions.

VIPmon is an optional device that performs session admission control and prioritization at the e-merchant, so as to keep the performance of admitted sessions within desired performance bounds. VIPmon prioritizes each session according to CoS markings in client packets and load and revenues generated by recent sessions having the same client address. VIPmon favors *VIP*, *up-and-coming*, and *regular* sessions (in that order), in detriment of *disappointing* sessions. Disappointing sessions are those that have the same client address and CoS as one or more recent sessions that have consumed excessive e-merchant resources without generating revenues. Conversely, up-and-coming sessions are regular CoS sessions that have the same client address and CoS as one or more recent sessions that have generated significant revenues without unduly consuming e-merchant resources. VIP sessions are those that have the VIP CoS marking and are not disappointing. (However, if an excessive number of VIPs access the e-merchant at the same time, VIPmon may downgrade one or more VIP sessions to a lower priority, according to the recent load and revenues generated by each client.) Regular sessions are all other sessions.

VIPmon is typically implemented in a customer-premises front-end or Web switch. The advantage of using VIPmon is the ability to respond to client load and revenues more quickly and in more nuanced fashion than is possible using only VIPGs.

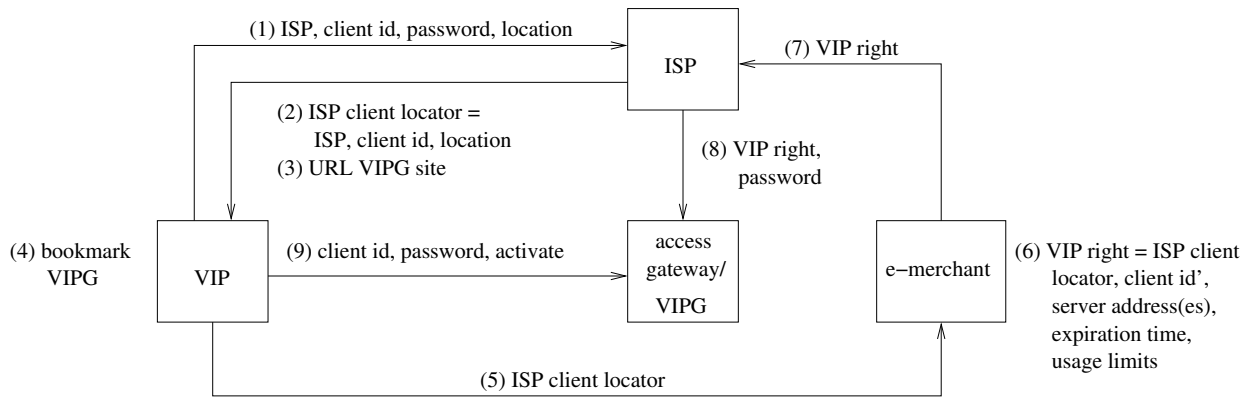


Figure 2: The VIP protocol allows a client to obtain and use VIP rights for accessing an e-merchant.

3.4 QoS requirements

This subsection discusses the QoS support that is necessary in ISPs, e-merchants, and clients for protecting the VIP class from overload in the regular class.

VIPnet assumes that ISPs can support at least two classes of service, called *VIP* and *regular*. The VIP CoS is used for packets sent by VIPs, whereas the regular CoS is used for packets sent by regular and selected clients. Different classes of service may be implemented using, e.g., diffserv [1], integrated services (intserv) [2], or other QoS scheme. Most routers currently sold support at least one such scheme. The regular CoS may be, e.g., the network's best-effort CoS. The VIP CoS may be privileged, e.g., by having higher priority than that of the regular CoS, or having a certain minimum share of each required resource, such as bandwidth and buffer space. The CoS may be marked, e.g., in the TOS (type of service) field of the packet's IP header.

More than one ISP may be involved in carrying packets from a client to an e-merchant. In such cases, each ISP must be able to differentiate VIP and regular traffic, but it is not necessary that all ISPs use the same mechanisms to achieve such differentiation.

At peering points, an ISP A that supports VIPnet needs to map CoS markings in packets received from each other ISP B. If B also supports VIPnet, then A needs to map B's VIP and regular CoS markings respectively into A's equivalent CoS markings. On the other hand, if B does not support VIPnet, then A needs to map any CoS markings into A's regular CoS marking. These mappings may require changing the TOS and correspondingly updating the checksum of the IP header of each packet. These operations are not expensive. IPsec, TCP, and UDP checksums do not depend on the TOS and do not need to be updated.

E-merchants also need to separate the resources used by each class (session class if using VIPmon, or client CoS otherwise). In large sites, this can be achieved by using separate hosts as servers for each class. A Web switch forwards the traffic of each class to the respective servers. Each server may run a conventional operating system.

Alternatively, in small sites, a single host running an operating system with QoS support may be used as server for all classes. Two examples of operating systems that may be used for such purpose are Eclipse/BSD [3] and QLinux [27].

ISP customer network and access links are outside the scope of VIPnet. However, it should be noted that con-

gestion or disruption in an ISP customer's network (e.g., at a company or university) or shared-medium access link (e.g., cable or wireless) can affect VIPs that access an ISP via such a network or access link. These vulnerabilities can be eliminated by using (1) customer-premises networks that support and enforce different classes of service, and (2) exclusive access links, such as DSL, dial-up, ISDN, or leased lines.

3.5 VIP protocol

This subsection describes the VIP protocol, which is used to request, install, activate, and bill VIP rights.

A simplified version of the VIP protocol is shown in Figure 2. (As discussed below, versions with stronger client authentication can be readily derived from this.)

The first stage in the VIP protocol comprises steps 1 through 4 in Figure 2. The client performs this stage only once per desired ISP and location. First, the client chooses an ISP, client identification (id), password, and location for using VIP rights. The location may represent one of the ISP's PoPs (for a location-specific right) or be left unspecified (for a location-independent right). The client may make these choices, e.g., using an ISP's secure Web site, which should check that the client id and password are not easily guessable. Second, the client obtains from the ISP an *ISP client locator*, which is a data structure containing the ISP name (in cleartext) and client id and location (both encrypted by the ISP using a secret key known only to the ISP, so as not to disclose them to the e-merchant or other parties). The client may download this data structure from the ISP's site as a file. Third, the ISP informs to the client the URL of the password-protected VIPG Web page where the client can activate and deactivate the client's VIP rights. This may be implemented by Web redirection. Fourth, the client bookmarks or writes down this URL.

The second stage in the VIP protocol comprises only step 5. The client provides the ISP client locator to an e-merchant that the client wishes to access as a VIP. This may be implemented by uploading the ISP client locator to a secure Web site maintained by the e-merchant. The e-merchant may preserve past client information, so that the client may need to perform this stage only once per location and e-merchant that the client desires to access as a VIP.

The third stage comprises steps 6 to 8, and is performed by the e-merchant each time the e-merchant grants new VIP

rights. In the sixth step, the e-merchant prepares a VIP right, which is a data structure containing the ISP client locator, e-merchant’s client id (encrypted by the e-merchant using a secret key known only to the e-merchant, so as not to disclose it to the ISP), e-merchant name or address(es), expiration time, and usage limits (the latter three items in cleartext). In the seventh step, the e-merchant requests the ISP to include the VIP right in the ISP’s VIP list. This may be implemented employing a mutually authenticated and encrypted channel between e-merchant and ISP, e.g. using TLS [17]. In the eighth step, the ISP decrypts the locator embedded in the VIP right and includes the VIP right in the corresponding VIP list. If the VIP right’s location is specified, then the ISP updates the respective VIPG’s VIP list, otherwise the ISP updates the ISP’s location-independent VIP list. The former update may be implemented employing a mutually authenticated and encrypted channel between the ISP and each of the VIPGs, e.g. using TLS [17].

The fourth stage comprises step 9 only. When desired, the client goes to a VIPG’s password-protected Web page to activate or deactivate the respective VIP rights.

The fifth and final stage comprises steps 10 and 11. Periodically, the ISP verifies the usage of each VIP right and bills the e-merchant. The bill may, e.g., include a minimum monthly fee per VIP right that the e-merchant requests the ISP to install, plus an amount proportional to the actual network usage of the e-merchant’s VIPs.

The protocol shown in Figure 2 can be easily strengthened, e.g., by having the ISP give the client a hardware token (e.g., SecurID [25]) in step 3, and requiring the client to combine the hardware token with the client’s password in step 9. Certificate-based strengthening is also possible.

4. DISCUSSION

This section discusses VIPnet’s advantages and limitations.

4.1 Advantages

The key advantage of VIPnet is that it makes it very difficult for an attacker to mount a successful DoS attack against an e-merchant’s VIP clients. Currently, an attacker can easily scan for vulnerable computers to use as agents for a DoS attack: *Any* computer will do. VIPnet changes that. Attacks launched from regular clients do not affect VIPs because VIP packets are carried in a separate CoS. Consequently, attacks against VIPs need to be launched from VIP clients. Because VIP rights cannot be forged, the attacker can use *only* computers that have active VIP rights for the intended victim. Therefore, the universe of potential agents is smaller and more difficult to scan for. Moreover, the traffic that any one VIP agent might generate is limited, because VIP rights are term- and usage-limited. Consequently, the attacker cannot sustain an attack.

A corollary of the above advantage is that VIPnet protects a (perhaps major) part of an e-merchant’s revenues from the effects of congestion and DoS attacks.

The other main advantage of VIPnet is that it allows e-merchants to provide to select clients a superior quality of service. This can be proved as follows. Because VIP rights are usage-limited, a rational client will attach to each VIP right some value w that the client will not be willing to spend unless it improves his or her QoS by some amount $q > 0$. Let the QoS of regular clients be denoted Q_r and the QoS

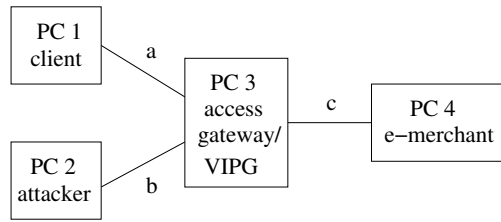


Figure 3: Experimental setup.

of VIPs be denoted Q_v . If $Q_r + q > Q_v$, then some VIP will prefer to deactivate and save the respective VIP right. This deactivation adds one client to the regular CoS, reducing Q_r , and subtracts one client from the VIP CoS, increasing Q_v . Further deactivations will follow until $Q_r + q \leq Q_v$, where $q > 0$. Therefore, the QoS enjoyed by VIPs will be greater than that enjoyed by regular clients.

4.2 Limitations

VIPnet assumes that the interaction between client and server is transaction-based, where normally, by the end of each transaction, the server obtains a payment from the client. This allows VIP rights to be usage-limited, which in turn guarantees that even compromised VIPs cannot sustain an attack against the VIP CoS. Although a transaction-based model is natural for e-merchants (e.g., B2B, B2C, stock brokerage, and auction sites), it is not characteristic of many other Web sites that may also need protection. For example, Web sites may have flat-fee subscription-based revenues (e.g., WSJ and many other online publications), advertising-based revenues (e.g. Yahoo and other portals), or no revenues at all (e.g., the White House and other governmental or nonprofit sites). This paper leaves open the question of how to protect such other sites from DoS attacks.

VIPnet also assumes that, on average, the cost of providing VIP services during a transaction does not wipe the transaction’s profitability. Whether this assumption holds depends on how VIPnet is priced.

4.3 Related work

Like VIPnet, Paris Metro Pricing (PMP) proposes a limited number of classes of service for the Internet, differentiated by price [23]. However, in PMP, any host can use the higher class to send any number of packets to any other host. Therefore, unlike VIPnet, PMP allows attackers to easily find agents and sustain attacks: Any computer on the higher class can be used as an agent indefinitely.

5. EMPIRICAL VALIDATION

In order to validate our design empirically, we implemented a VIPG prototype and report its performance in this section.

5.1 Experimental setup

Figure 3 illustrates our experimental setup. We used four personal computers (PCs) connected by Fast Ethernet links at 100 Mbps. The characteristics of each PC are shown in Table 1. The network interface cards (NICs) used were Intel EtherExpress PRO/100+. ² The first PC was used as a legitimate client, the second PC was used as an attacker,

²These NICs are poor choices for implementing a VIPG.

PC	Role	CPU	Memory	Operating System
1	Client	Pentium II 233 MHz	64 MB	FreeBSD 4.2-R
2	Attacker	Pentium II 266 MHz	64 MB	FreeBSD 4.2-R
3	Access gateway	Pentium II 300 MHz	32 MB	FreeBSD 4.2-R (modified)
4	E-merchant	Pentium II 400 MHz	128 MB	FreeBSD 3.4-R (modified)

Table 1: Characteristics of the computers used.

Operating Systems		TCP throughput 4 → 1 (Mbps)			
PC 3	PC 4	without attack		with attack 2 → 4	
		Average	Std. dev.	Average	Std. dev.
Generic	Generic	72.6	0.2	1.8	0.1
Generic	SRP	72.6	0.2	1.6	0.2
VIPG	Generic	72.8	0.2	54.4	0.6
VIPG	SRP	72.7	0.2	55.6	0.3

Table 2: The VIP Gate (VIPG) introduces negligible overhead and greatly improves performance for a VIP client when the e-merchant is under attack.

the third PC was used as an access gateway/VIPG, and the fourth PC was used as an e-merchant.

We modified the operating system of the third PC (access gateway) as follows, so as to support two classes of service and VIPG. First, we replaced IP’s input queue by two input queues, one for the VIP class and the other one for the regular class, where the VIP class was configured with priority higher than that of the regular class. Second, we inserted in the `ether_input` routine a call to a new `vip_classify` routine. For each packet received from an interface configured as an `access` interface (all links in Figure 3), `vip_classify` verifies if the packet’s source is a VIP with respect to the packet’s destination (or vice-versa), encodes the result in the IP header’s TOS (type of service) field, updates the IP header’s checksum, and returns a pointer to the corresponding input queue. For each packet received from an interface that is not configured as an access interface, `vip_classify` returns the input queue selected by the IP header’s TOS field. Third, we used the ALTQ patch [13] to implement two output queues per output link, selected by the TOS field in each packet’s IP header. (Ordinarily, FreeBSD uses a single FIFO queue per output link.) The VIP class was configured with priority higher than that of the regular class. Each output link was configured with a token bucket rate limiter, with average rate equal to 98 Mbps and bucket size equal to 4 KB.

Also to support different classes of service, we modified the operating system of the fourth PC (e-merchant) as follows. We used the SRP patch [4, 3] to change how the protocol processing of received packets is scheduled. Ordinarily, FreeBSD performs such processing in the context of a software interrupt, with priority higher than that of any application. This can generate scheduling anomalies and allows a sufficiently high reception rate (whether malicious or

First, they interrupt the PC’s CPU whenever a packet arrives. Thus, arrival of regular packets can interrupt processing of VIP packets. Second, they internally use a FIFO transmit queue and achieve maximal transmission rates only if fed several packets in advance. Thus, it is not possible to schedule transmission strictly according to priority — a newly arrived VIP packet may need to await transmission behind previously fed regular packets. Therefore, some deviation from ideal results can be expected.

not) to easily knock out a FreeBSD server. (The same vulnerability is present in most operating systems derived from or inspired by Unix.) SRP modifies the operating system so that each socket gains its own queue of unprocessed input packets, and protocol processing of each packet occurs only when the respective receiving process is scheduled.

The operating systems of the first two PCs (client and attacker) were not modified at all.

5.2 Experimental results

We measured the e-merchant’s TCP throughput under a simulated smurf attack with or without a VIPG. In this experiment, PC 1 (client) was configured as a VIP of PC 4 (e-merchant), while PC 2 (attacker) was configured as a regular client. All results reported here are the averages of five measurements. There was no other load on the computers or network links.

To simulate a smurf attack, we modified `sing`, a utility available from FreeBSD’s port collection [21]. Our modifications cause `sing` to generate an unending flood of ICMP echo reply packets. We ran this utility in PC 2 (attacker) and directed its output to PC 4 (e-merchant). The measured rate of the attack was about 51,300 packets per second. The ICMP process in PC 4 (e-merchant) was configured with a time-sharing priority.

We used the `ttcp` utility with default parameters to measure the TCP throughput from PC 4 (e-merchant) to PC 1 (VIP client). Like `sing`, `ttcp` is available from FreeBSD’s port collection. We ran the e-merchant’s `ttcp` at a real-time priority.

We repeated the TCP throughput measurements under the following cases: (1) presence or absence of the simulated smurf attack; and (2) presence or absence of the operating system modifications discussed in the previous subsection (VIPG in the access gateway, and SRP in the e-merchant). The results are shown in Table 2, where “Generic” denotes an unmodified operating system.

The TCP throughput in the absence of an attack was essentially the same, regardless of whether VIPG or SRP were used. Therefore, the overheads introduced by VIPG and SRP are negligible. SRP’s effect on the TCP throughput when the e-merchant was under attack was negligible when SRP was used by itself, and very small when SRP was

used with VIPG. (This illustrates that victims often cannot thwart DoS attacks without ISP cooperation, as many losses occur before packets reach the victim.) On the other hand, VIPG greatly improved the TCP throughput when the e-merchant was under attack. Without VIPG or SRP, the attack caused the throughput to drop by 97.5%. With VIPG, the attack caused the throughput to drop by only 25.2% (without SRP) or 23.6% (with SRP).³

6. SUMMARY

DoS attacks are a major threat to e-commerce and if unabated may curtail use of the Internet for business purposes. Congestive DoS attacks are particularly challenging because victims cannot protect themselves without other parties' cooperation. Moreover, the current Internet architecture makes it easy for attackers to mount and remain unaccountable for such attacks. This paper introduces a new QoS-based defense architecture that limits the effects of congestive DoS attacks on e-merchants. VIPnet allows e-merchants to have ISPs carry the traffic of the e-merchants' best clients, called VIPs, in a privileged class of service. The VIP class enjoys better quality of service and is insulated from congestion, whether malicious or not, in the regular class. DoS attacks against VIPs are difficult to mount and sustain because attackers cannot forge VIP rights, attackers cannot easily find and infiltrate computers with active VIP rights for an intended victim, and VIP rights are term- and usage-limited. Consequently, VIPnet can protect a (perhaps major) portion of an e-merchant's revenues from the effects of congestion and DoS attacks. For this service, e-merchants pay a fee to ISPs, thus amortizing the necessary investment.

7. REFERENCES

- [1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss. "An Architecture for Differentiated Services," IETF, RFC 2475, Dec. 1998.
- [2] R. Braden, D. Clark, S. Shenker. "Integrated Services in the Internet Architecture: an Overview," IETF, RFC 1633, June 1994.
- [3] J. Bruno, J. Brustoloni, E. Gabber, B. Özden, and A. Silberschatz. "Retrofitting Quality of Service into a Time-Sharing Operating System," in *Proc. Annual Tech. Conf.*, USENIX, June 1999, pp. 15-26. Software available at <http://www.bell-labs.com/project/eclipse/release/>.
- [4] J. Brustoloni, E. Gabber, A. Silberschatz, and A. Singh. "Signaled Receiver Processing," in *Proc. Annual Tech. Conf.*, USENIX, June 2000, pp. 211-223. Patch available at <http://www.bell-labs.com/project/eclipse/release/>.
- [5] CERT. "CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack," CERT/CC, available at <http://www.cert.org/advisories/CA-1996-01.html>.
- [6] CERT. "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks," CERT/CC, available at <http://www.cert.org/advisories/CA-1996-21.html>.
- [7] CERT. "CERT Advisory CA-1996-26 Denial-of-Service Attack via ping," CERT/CC,

available at <http://www.cert.org/advisories/CA-1996-26.html>.

- [8] CERT. "CERT Advisory CA-1997-28 IP Denial-of-Service Attacks," CERT/CC, available at <http://www.cert.org/advisories/CA-1997-28.html>.
- [9] CERT. "CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks," CERT/CC, available at <http://www.cert.org/advisories/CA-1998-01.html>.
- [10] CERT. "CERT Advisory CA-1999-17 Denial-of-Service Tools," CERT/CC, available at <http://www.cert.org/advisories/CA-1999-17.html>.
- [11] CERT. "CERT Incident Note IN-2000-05," CERT/CC, available at http://www.cert.org/incident_notes/IN-2000-05.html.
- [12] CERT. "CERT Security Improvement Modules," CERT/CC, available at www.cert.org/security-improvement/.
- [13] K. Cho. "Managing Traffic with ALTQ," in *Proc. FREENIX Annual Tech. Conf.*, USENIX, June 1999, pp. 121-128. Software available at <http://www.cs1.sony.co.jp/person/kjc/kjc/software.html>.
- [14] Cisco. "Characterizing and Tracing Packet Floods Using Cisco Routers," Cisco, available at <http://www.cisco.com/warp/public/707/22.html>.
- [15] Computer Security Institute and Federal Bureau of Investigation. "CSI/FBI Computer Crime and Security Survey 2001," CSI, Mar. 2001, available at <http://www.gocsi.com/>.
- [16] J. David et al. "Results of the Distributed-Systems Intruder Tools Workshop," CERT/CC, Pittsburgh, PA, Nov. 1999, available at http://www.cert.org/reports/dsit_workshop.pdf.
- [17] T. Dierks and C. Allen. "The TLS Protocol Version 1.0," IETF, RFC 2246, Jan. 1999.
- [18] D. Dittrich. "The "stacheldraht" Distributed Denial of Service Attack Tool," available at <http://staff.washington.edu/dittrich/misc/stacheldraht.analysis>.
- [19] P. Ferguson and D. Senie. "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing," IETF, RFC 2827 (also BCP 0038), May 2000.
- [20] C. Huegen. "The Latest in Denial of Service Attacks: "Smurfing" Description and Information to Minimize Effects," available at <http://www.pentics.net/denial-of-service/white-papers/smurf.cgi>.
- [21] G. Lehey. "The Complete FreeBSD," 773 pp., 3rd. ed., Walnut Creek, CA, June 1999.
- [22] D. Moore, G. Voelker and S. Savage. "Inferring Internet Denial-of-Service Activity," to appear in *Proc. Security Symp.*, USENIX, Aug. 2001.
- [23] A. Odlyzko, "Paris Metro Pricing for the Internet," in *Proc. 1st ACM Conf. Electronic Commerce (EC'99)*, ACM, 1999, pp. 140-147.
- [24] S. Savage, D. Wetherall, A. Karlin and T. Anderson. "Practical Network Support for IP Traceback," in *Proc. SIGCOMM'2000*, pp. 295-306, ACM, Stockholm, Sweden, Aug. 2000.
- [25] SecurID. Homepage at <http://www.rsasecurity.com/products/securid/>.
- [26] D. Senie. "Changing the Default for Directed

³NICs that interrupt the PC's CPU less than do the NICs used and support different priority output queues should allow the TCP throughput to fall even less under attack.

Broadcasts in Routers,” IETF, RFC 2644 (also BCP 0034), August 1999.

- [27] V. Sundaram, A. Chandra, P. Goyal, and P. Shenoy.
“Application Performance in the QLinux Multimedia
Operating System,” Tech. Rep. UM-CS-2000-011,
Univ. Massachusetts, Amherst, MA, Mar. 2000.
Software available at
<http://www.cs.umass.edu/~lass/software/qlinux/>.