W H I T E P A P E R

Mitigating Denial of Service

- **CRIPPLING E-BUSINESS**
- **HOW CAN WE BE SO VULNERABLE?**
- **TYPES OF ATTACK**
- **GENERAL ATTACK PREVENTION**
- **RESPONDING TO POISON ATTACKS**
- **RESPONDING TO STATEFUL ATTACKS**
- **RESPONDING TO RESOURCE ATTACKS**
- **SPREADING THE RISK**
- **CONCLUSIONS**

Alteon WebSystems, Inc.

50 Great Oaks Boulevard

San Jose, California 95119

408-360-5500

408-360-5501 fax

http://www.alteon.com

INTRODUCTION

Denial-of-service (DoS) attacks try to interrupt Web operations by overwhelming or interrupting servers. Such attacks have gained notoriety in recent months as visible Internet sites have fallen victim. These attacks are launched by new, more sophisticated software that can solicit the assistance of innocent third parties whose computers are infected.

This white paper describes the nature of these attacks, and the various steps that can be taken to prevent them or mitigate their effects through the use of Web switching equipment. Web switches have extremely high capacity that can thwart many attacks through the sheer volume of connections they can sustain when compared to server or appliance-based load-balancing systems.

CRIPPLING E-BUSINESS

DoS attacks can have a crippling effect on business. They prevent legitimate users from accessing the site, and degrade performance substantially; in today's point-and-click economy, this means potential customers look elsewhere. Detecting and correcting attacks is also costly, involving close interaction between service providers and IT personnel and interrupting normal business activities within the company. Attacks can have a negative effect in the press and the capital markets –buy.com sustained attacks in its first day of trading as a public company. Finally, these attacks can render servers temporarily unavailable, causing crashes that may destroy data and require downtime for corrective action.

HOW CAN WE BE SO VULNERABLE?

Why are attacks so hard to resist? In the rapid innovation of the Internet, why haven't wily engineers devised protection from these assaults? The answer comes down to four key factors: software bugs, anonymity, necessary weaknesses, and the involvement of innocent third parties.

Software bugs

Today's operating systems are complex and broadly available, often with millions of lines of code. Such complex systems cannot be perfectly tested for every possible condition and still get to market in a timely fashion. This means that hackers can find "boundary conditions"– unlikely or unanticipated circumstances in which an operating system stops working or grants a hacker access – and then apply these hacks to public systems.[1]

While vendors are constantly issuing patches and fixes to software, many organizations lack the knowledge or personnel to maintain the latest code on all of their systems.

Standardization

At the root of many vulnerabilities is standardization. Hackers and malicious users thrive in highly-standardized environments. On the Web, a few protocols link a few operating systems running a few servers. Back-end infrastructures such as databases and application servers are undergoing a similar standardization. Once hackers discover a weakness in a system, they can exploit it across a wide range of targets.

Anonymity

The Internet allows hackers to cover their tracks. Often, attacks come from "owned" systems to which hackers have gained access – so that when the machine performing the attack is finally tracked down, the hacker is long gone. The Internet's protocols also allow attackers to hide. Every packet of data sent across the network carries a

source and a destination address that uniquely defines a sender and receiver. But certain attacks allow the hacker to simply make up source addresses – a process known as spoofing-and render the attack nearly untraceable.

Underlying weaknesses

This lack of traceability is a function of some necessary weaknesses in the underlying protocols of the Internet. The net is a connectionless medium in which packets are sent back and forth across many paths. This is in sharp contrast to a connection-oriented system such as the telephone network, where a link is established in order to send data.

Because the Internet is connectionless, certain protocols – such as TCP – need to simulate a reliable, point-to-point link across it. Setting up this link happens in several stages, known as the TCP synchronization process, during which the sender and receiver agree to talk.

Since the Internet is an unreliable, best-effort medium, delays can occur in synchronization. The receiver responds to a synchronization request from a sender, and then waits patiently for the receiver to acknowledge the response. During this time, the TCP session is embryonic: it has not yet been established, but the receiver is keeping track of it.

By default, servers track embryonic TCP sessions for several minutes. This allows clients and servers to establish sessions despite long network delays, and is necessary for a ubiquitous, global Internet. But the patience of the receiver can be exploited.

Attacking a server by establishing many embryonic TCP sessions is one example of taking advantage of a necessary weakness in networking protocols. There are many others.

Innocent third parties

The final reason that attacks are hard to defend against is that they often involve innocent third parties. Today's leading sites have huge amounts of capacity at their disposal. In order to hurt such sites, hackers need to enlist the help of other networks and devices. Some attacks trick other devices into responding to what appears to be a legitimate request from the victim – and these responses overwhelm the victim. Others infect systems with "Trojans," attack software that can later be used to target a victim. With the broad deployment of DSL and "always online" local loop connections from cable providers, hackers have an even broader base of unwitting third parties to exploit.

TYPES OF ATTACK

There are three basic forms of attack that can bring down a site.

Poison attacks

A poison attack sends toxic information to a target. This can take the form of a malformed packet that the receiver doesn't understand, or an oversized packet that exceeds the buffers of the target system. In some cases, a properly constructed poison packet can include code that the receiver inadvertently executes when buffers are exceeded.

State resource attacks

A state attack consumes resources on a destination server by forcing the server to expend much more effort than the receiver processing and tracking state. The TCP SYN attack is the best example of such an attack, although other attacks include flooding a server with SSL session requests that force it to perform heavy cryptographic computation.

Capacity resource attacks

Even if a site is relatively well-defended against poison and state attacks, attackers can still overwhelm bandwidth into the site. In recent Internet attacks, even when the target sites were aware of the problem it was too late: upstream providers' peering points were already handling more capacity than most countries. A capacity attack simply uses up more capacity than the victim owns.

GENERAL ATTACK PREVENTION

Preventing attacks requires work at both ends of the network. ISPs and system administrators can take steps to ensure that the devices under their control don't participate in an attack. This begins with a good security policy, but can also include steps to cripple would-be attacks from infected machines.

Preventing zombification

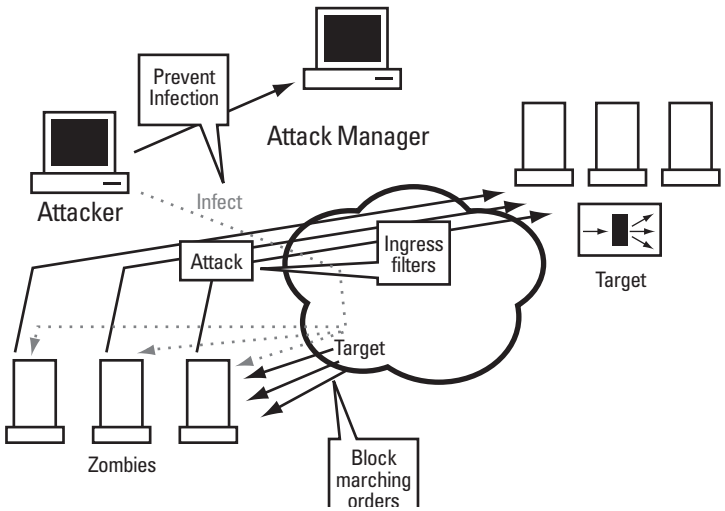
Machines can't participate in an attack if they aren't infected in the first place. Good security practices and clean servers are an essential preventative measure, but in today's complex server environments malicious code can masquerade as legitimate software and hide unnoticed in myriad subdirectories. Scanning tools are able to detect and remove some malicious code, but scanner vendors are in a constant race with hackers.

Blocking marching orders

Even if a system is infected, it still needs to get instructions so that it knows which systems to attack. These "marching orders" are stealthy, often encoded in seemingly innocuous traffic such as ICMP response messages. Some firewalls can block a portion of this traffic, but attack systems are surprisingly robust and hard to detect.

Ingress filters

Many of the most dangerous attacks involve spoofing the true source address of the attacker and replacing it with either random addresses (to hide identity) or the address of the victim (to trick devices into responding to the victim instead of the attacker.) By filtering downstream traffic, an ISP can prevent attacking machines from masquerading as another system. With the advent of "zombie" attacks, however, spoofing is less and less of an issue. Furthermore, inspecting traffic is costly in terms of processing and can discourage some service providers from implementing such mechanisms.



Zombie attacks might be stopped through infection prevention, ingress filtering, or blocking the "attack" signals that hackers send to infected systems.

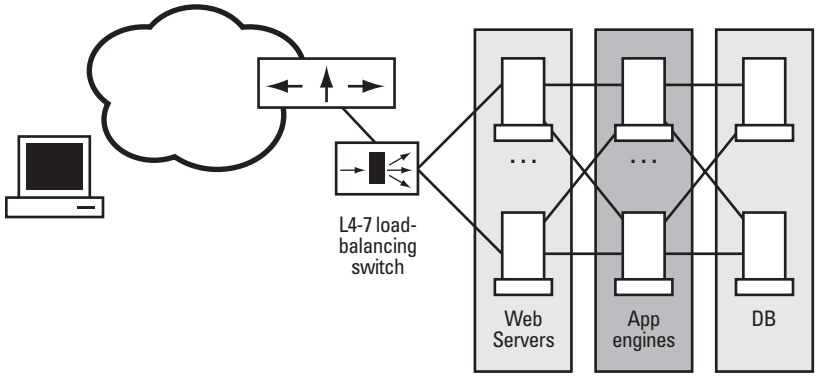
Don't bring a knife to a gunfight

These preventative measures are all wise, responsible steps to take. But they rely on altruism and assume that systems administrators have the time and knowledge to implement them. To make matters worse, several of them will simple be bypassed in later revisions of hacker code – with morphing Trojans or new attack messages, for example.

In other words, we can't rely on preventative measures to solve the problem.

RESPONDING TO POISON ATTACKS

A Web server can be protected against poison attacks through "hardening" the TCP/IP stack of the system. This involves disabling any unnecessary services or daemons, patching known bugs and weaknesses, and filtering incoming traffic to ensure that it is safe for the stack to handle before processing it. Since this is a complex process – further complicated by heterogeneous operating systems and multi-tiered server farms – a Web switch or firewall is an excellent "hard front" for attacks since it can be maintained and updated without touching individual servers.



RESPONDING TO STATEFUL ATTACKS

Sites can respond to stateful attacks in a number of ways. These are best implemented on a Web switch that sits between the servers and the outside world, since the Web switch is a network appliance that can perform the initial session establishment and "screen" the entire connection to ensure that it is legitimate. Web switches have the brains and brawn to deal with denial-of-service attacks:

- A Web switch can filter traffic as it arrives at the site to look for illegal addresses, bad packets, or too many concurrent connections that may signal an attack.

- An intermediate device can simply bring more resources to bear on a connection. While servers may only be able to handle a handful of concurrent TCP connections, Web switches are optimized to handle thousands every second. This means they have more state resources for an attacker to consume before the attack affects the site.

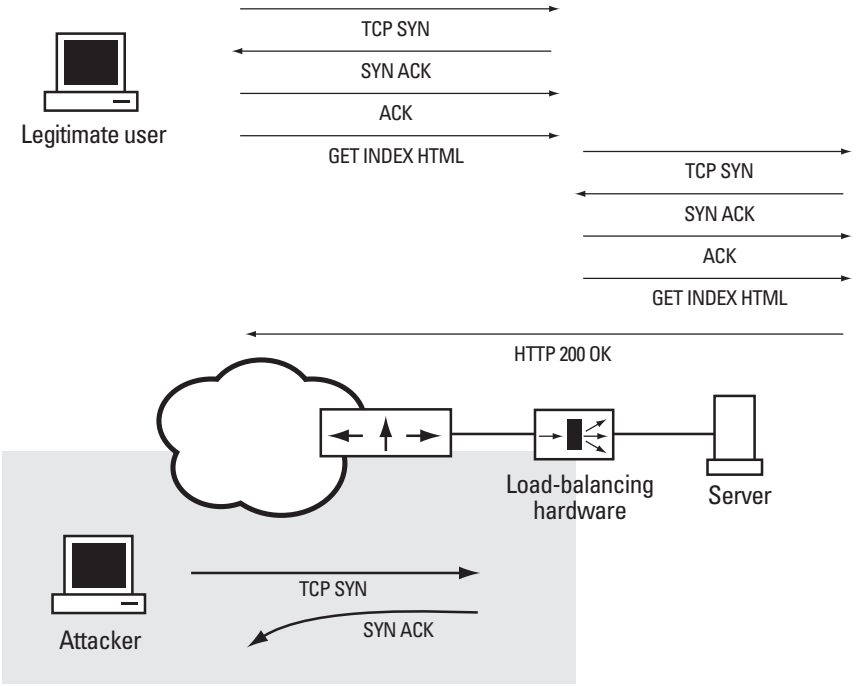
- The Web switch can tune its parameters more aggressively to overcome some of the default values that Internet protocols use. For example, the standard for TCP says to wait several minutes before declaring that an embryonic connection has vanished. In today's Internet, TCP setup times of more than 10 seconds are rare and will often cause users to go elsewhere. Consequently, Web switches can declare an embryonic session dead after a few seconds, and either remove the connection from their state tables (if they are terminating the session themselves) or reset the connection on the server (if they are forwarding sessions to a server immediately).

- Some of the underlying protocols can change. While this is a matter for the Internet's standards bodies, a number of innovative mechanisms have already emerged such as SYN cookies. SYN cookies allow a device to set up TCP sessions without tracking session state by embedding session idenfitiers in the SYN ACK and ACK messages themselves – effectively using the datagram to store the state rather than consuming Web switch resources.

- When state resources, such as pending connections, begin to disappear under an attack, a Web switch can "reap" older connections from the state table in order to overcome the load. It can also terminate these connections from servers.

- Web switches can process network traffic far more quickly than a server might, particularly when the server is performing other work such as running applications or querying databases. By reducing the time it takes to process state information, these devices can handle more states more quickly and clear out attempted attacks.

A Web switch that terminates network connections before binding them to a particular server can detect and block incoming traffic when appropriate, making it an essential part of any attack prevention strategy.



Many of the sites that were hacked in recent months had implemented prevention mechanisms to overcome stateful weaknesses through the use of hardening and Web switching technology. The source of the outages was raw resource consumption – the hackers' zombie armies against the impressive capacity of leading sites.

RESPONDING TO RESOURCE ATTACKS

Resource attacks are harder to defend against than state or poison attacks because they overwhelm a site through sheer volume.

Capacity

The simplest response to a resource attack is to have more resources. By collocating with a service provider and ensuring that the provider has sufficient upstream bandwidth and peering agreements, site owners can simply overpower an attack. Unfortunately, this is an extremely expensive prospect open to only the largest of organizations.

SPREADING THE RISK

The best defense is to avoid putting all of a site's eggs in one virtual "basket" so that an attack might disable one part of the Web service – such as a single physical site – without interrupting the service as a whole. This approach is known as global server load balancing (GSLB).

GSLB means designing a system that will send attackers to only a portion of the service and then forward legitimate users to an alternate site once the attack is underway. This can be achieved through a two-tier DNS model that returns a list of multiple authoritative name servers for a service, and then returns one address for a single site from an availability-aware DNS server.

The service must be deployed across multiple sites, each connected to a different service provider's network and each having its own distinct virtual IP address (VIP).

When a client wants to connect to a service, they first request the VIP from their local DNS server. The local DNS then asks the root DNS server at Internic for one or more authoritative name servers that can resolve the URL for which the client asked. The client's DNS then asks the authoritative servers for the VIP, and ultimately returns the VIP to the browser so that the user can visit the site.

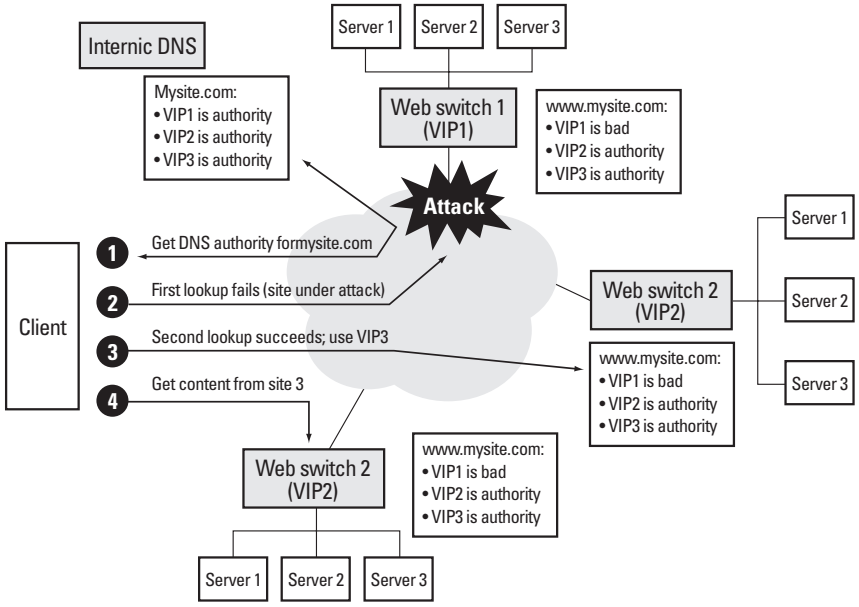
The Internic DNS has four IP addresses pointing to the four authoritative name servers for the service. Each authoritative DNS is a WAN-enabled Web switch running at one of the four sites. A query proceeds as follows:

- The client's DNS iterates through the four entries (ns1.mysite.com to ns4.mysite.com), connecting with one of the four authoritative name servers.
- If the client's DNS query to the authoritative name server times out, it tries another one of the four as part of the default behavior for DNS.
- When client's DNS tries to resolve www.mysite.com from the chosen name server (such as ns3.mysite.com)

that name server responds with a single VIP for a chosen site. It "prefers" its local site unless the local site is heavily loaded.

- If an attacker tries to attack www.mysite.com, the zombies receive only one IP address as their target (since the Web switch hands out only one at a time.)

- Once the attack is underway, resolutions go to an unattacked site. This happens for two reasons. First, the Web switches detect the congestion at the attacked site and hand out other VIPs to subsequent requests. Second, when DNS clients try to resolve a name with a DNS server that is at an attacked site, the resolution times out and the DNS client tries another name server in the list.



The following diagram illustrates this process for a sample site with three Web switches. Each switch is both an authoritative DNS for the site and a switch for server traffic.

This model presents multiple targets to an attacker, which increases the aggregate bandwidth the attacker must overwhelm and increases the complexity of the attack software and the number of zombies needed to mount an assault.

Database complexities

The aforementioned model works well for services that can have many physical sites. This is not always possible. In today's dynamic, database-driven environments, all sites need access to the same content instantaneously. This is often achieved by real-time synchronization between many sites – but the synchronization can fall victim to a denial-of-service onslaught.

The typical approach to database-driven sites is to collocate all equipment in one location. This gives attackers a single target. By employing the aforementioned techniques and having multiple inbound links to multiple peering

points on multiple networks, a Web switch can make a single server cluster seem like many physical locations while keeping database content in a single location.

To do this, each Web switch must have both an overlapping address block and a unique address block. DNS runs on the shared, overlapping address block while multiple instances of the Web service – with distinct VIPs – run on each unique block.

CONCLUSIONS

With e-business under siege from anonymous hackers, companies need to address the denial-of-service threat to retain customers, avoid liability, and protect themselves from long-term damage. This can be accomplished through the deployment of Web switching equipment to block poisoned packets and mitigate state resource attacks, and by careful configuration of an availability-aware, multi-tiered WAN Web switching solution such as GSLB.

Alteon's Web switches form a distributed network processing engine that can handle huge volumes of traffic and shield sites from attack through advanced global load distribution architectures. The systems are resilient to SYN floods and other resource attacks, making them ideal for mission-critical e-commerce. ■

1 The Open Source movement holds that by making the source code publicly available, hackers and engineers will uncover all known loopholes and the system will become more robust as a result. By contrast, closed-source operating systems such as Microsoft's Windows NT are not freely open to inspection, but loopholes are sometimes less evident as a result. The debate over which model works best rages on.