Lack of Priority

"There is a difference between making something foolproof and reducing the number of fools."

We're in sore need of
critical Internet
infrastructure protection.

—Bill Barns

Queuing

Considered Harmful

VIJAY GILL, AMERICA ONLINE

Most modern routers consist of several line cards that perform packet lookup and forwarding, all controlled by a control plane that acts as the brain of the router, performing essential tasks such as management functions, error reporting, control functions including route calculations, and adjacency maintenance. This control plane has many names; in this article it is the *route processor*, or RP. The route processor calculates the forwarding table and downloads it to the line cards using a control-plane bus. The line cards perform the actual packet lookup and forwarding. Although individual vendors or models may differ slightly in implementation, the salient points remain the same.

Service providers have observed an increase in DoS (denial of service) attacks targeted at the network infrastructure (routers, routing protocols, etc.). These attacks result in instability in the network and disruption of service (the route processor becomes overloaded, and the routing protocols start to flap, so the router no longer has a complete view of the universe; also, providers have seen cases where the switching fabrics become unstable—this should never happen in theory, but I have seen this on some equipment).

This article assumes some knowledge of DoS attacks (see <http://www.ietf.org/rfc/rfc2827.txt> for more information). There are many different types of DoS attacks, including the so-called “smurf” attacks, using aggregate traffic to overwhelm a single link, and “killer” packets that cause the routers to crash, as well as floods targeted at content hosts and servers for DNS (domain name system), etc.

Lack of Priority Queuing Considered Harmful

Another class of packet consists of the so-called “Christmas tree” packets, which take advantage of the fact that most router-forwarding hardware is optimized for IP packets with no options. These Christmas tree packets consist of IP packets with several IP options set. Board space and cost constraints mean that most (not all) packet-forwarding hardware is designed for normal option-less IP packets and does not have enough processing power to deal with IP packets that have many options. As these packets pass through routers, the forwarding hardware will punt the packets to the route processor for further processing. This consumes route processor resources, as well as bandwidth on the internal control communication bus, starving legitimate control traffic and possibly causing routers to lose adjacencies and/or crash.

This article focuses only on infrastructure attacks that target the router control plane, as these are the hardest and most troublesome types of attacks and because they are almost impossible to address no matter how robust the hardware becomes. By definition, designing a cost-efficient router implies that the cross-sectional bandwidth capacity of a router will exceed the bandwidth of the control bus by several orders of magnitude, which makes the control bus especially vulnerable to attack.

Infrastructure attacks are particularly insidious and damaging to the network because routers are designed for moving packets from one interface to another; they are not optimized for dealing with large amounts of traffic that terminates or must be sourced from the router (route processors). Examples of traffic that must be sourced from the router include sending ICMP (Internet control message protocol) unreachable packets and generating IP fragments if the MTU (maximum transmission unit) of an outgoing link layer is smaller than the packet size destined for it.

It is very hard to distinguish valid control traffic—BGP (border gateway protocol) keepalives, updates, connection opens, etc.—from invalid packets that look like legitimate control traffic (e.g., fake BGP update with the source address of a configured BGP peer). This fact can be exploited to swamp the router CPU with a large amount of invalid control traffic, causing the router to become unstable and/or crash.

Crashing or malfunctioning routers result in network

instability, which leads to poor performance. As the backbone responses become more effective, the attackers also evolve newer and ever-more sophisticated attacking techniques. There will probably never be one solution that effectively ends all attacks; as such, the focus is on both tactical and strategic options that are flexible and can evolve to meet the threat level.

This article details the network architecture, principles of operation, and the protective methodology for dealing with DoS attacks targeted at critical Internet infrastructure. This involves implementing hardware features that are important for the hardening of routers and other network infrastructure. These features are hard to retrofit after the fact. Any changes at this level will involve software rewrites and may require board-level changes; therefore, the earlier these requirements are aired, the better.

OVERVIEW

Routers are optimized for traffic *through* the hardware; they are not optimized for traffic *for* the hardware. Designing a cost-efficient router implies that the cross-sectional bandwidth capacity dominates the budget allocation. There is simply no cost-effective way to engineer a router that can absorb and usefully process data at the rate it can forward it.

A simple analysis follows: A router, circa 2004, has 32

Speaking BGP

The Internet consists of a network of networks. Each group of networks under a common administration is known as an AS (autonomous system). Each AS commonly also has a common routing policy that defines how that AS behaves. The BGP (border gateway protocol) is used to exchange routing information between autonomous systems, so that the different networks can communicate with each other. End users of the Internet, such as corporations, usually employ an IGP (interior gateway protocol) such as OSPF (open shortest path first) for the exchange of routing information within their networks.

BGP is an inter-AS routing protocol. The primary function of a BGP-speaking system is to exchange routing information with other BGP systems. This routing information includes data on the list of autonomous systems that routes traverse. This information is used to construct a graph of AS connectivity for the purposes of forwarding packets to the correct destination. For more information please visit <http://www.cluecentral.net/bgp>.

ports capable of forwarding data in either direction at 10 gigabits per second (Juniper T640, Cisco 12416; <http://www.juniper.net>; <http://www.cisco.com>). To forward data the router must de-encapsulate frames, verify checksums if any, perform a lookup on the header, and then send the packet to an outgoing interface. All this is done using customized hardware/software combinations and can be done at line rate for all practical purposes.

Consider, however, what happens if a packet arrives for the router. The router must perform the de-encapsulation, checksum verification, and then pass the packet onto the route processor, the “brain” of the device, for further processing. This involves not just looking at the packet header, but also at the body of the packet, and trying to make sense of what it contains.

This is resource intensive and the processor, often a single CPU, is easily overwhelmed, especially since 32 line cards, each capable of accepting up to 10 gigabits each second worth of traffic with dedicated packet-forwarding engines, can send a very large amount of data to the route processor.

This causes legitimate processes to become starved for bandwidth and processor power, resulting in missed protocol keepalives and other housekeeping tasks, which often results in adjacencies being dropped and routers becoming unreachable, with the attendant disruptions in the service provided by the network.

To make matters worse, some router designs have a low-bandwidth bus between the route processor and the line cards (which is a perfectly valid design, because the route processor is not designed for passing or processing large amounts of data).

REFERENCE HARDWARE DESIGN

Figure 1 shows a reference hardware design of a modern-day router. This is a generic representation of a router with a distributed forwarding mechanism and a centralized route processor that maintains protocol state, exchanges routing updates, and builds a forwarding table that is distributed to the line cards. The line cards do autonomous forwarding based on the forwarding table. The route processor takes no part in forwarding packets.

THE PROBLEM

Attackers are seizing this weak link. DoS attacks targeted at infrastructure are increasing. We have seen port 179 attacks already—and MSDP (multicast source discovery protocol) can’t be far behind.

It is important to distinguish between invalid and valid control traffic (e.g., BGP updates). Although rate-limiting on control traffic is necessary, it is not sufficient. Enough false data will swamp legitimate data, causing connection flaps/resets as keepalives and protocol updates are lost in the morass of valid-looking, but invalid, updates.

This article focuses on BGP. Most other traffic consists of either housekeeping things such as time protocols or in-band access to the device, which will not cause control-plane issues. IGP (interior gateway protocols) are either not routable or can be safely blocked at the edges of the network; either there is no valid reason for IGP updates to enter a network from an external source or it can be compartmentalized to a few specific routers.

PRIORITY QUEUING

Network devices could implement simple, unambiguous

parsing of any traffic that is destined for the control plane (RP): what filters deem interesting goes in the high-priority queue; everything else goes in the low-priority queue.

As shown in figure 1, there is normally a limited amount of capacity between the line cards and the route processor. To be effective, this capacity must be shared among all the traffic that is going to the RP for further processing. The traffic using this capacity is split up among

Lack of Priority Queuing Considered Harmful

housekeeping tasks internal to the router, such as downloading new copies of the forwarding table to the line cards, maintenance tasks, etc., as well as routing control-plane traffic such as BGP keepalives and updates, as well as random traffic destined to the router, ICMP, Telnet/SSH (Secure Shell), time protocols, etc.

Clearly, some traffic is more important. If the RP cannot get BGP updates or keepalives, the forwarding of traffic will suffer. On the other hand, if the network time protocol doesn't make it to the RP for a while, forwarding will be unaffected. So it is important to set up two queues: a high-priority queue that is serviced quickly for important traffic; and a low-priority queue that can be serviced if nothing is waiting in the high-priority queue.

A discriminator such as an access list should be available to filter the traffic that is destined for the router. This discriminator will decide which traffic is important (to be placed in the high-priority queue) and which is not (to be placed in the low-priority queue).

It is very important that at least two priority levels are actually available. Without priority queuing, a massive packet flood would just overwhelm the single queue, causing legitimate control traffic to be discarded or delayed, with the attendant disruption. With two queues, the discriminator can quickly sift important traffic and place it into the high-priority queue for service, while everything else gets placed in the low-priority queue, to be serviced if there is time. If the low-priority queue cannot be serviced for a while, no major harm comes to the network; and if the low-priority queue gets full, the router can quickly throw traffic away without the potential for causing harm to the network control plane.

The other half of the solution is rate-limiting. The low-priority queue should be rate-limited in a configurable manner, and should discard traffic aggressively.

For reasons stated earlier, by the time the traffic makes it to the route processor, it is often too late. This implies that the queuing and the discriminator application should happen on the line card/forwarding engines, before it gets passed onto the route processor.

As to what to filter, the router should be able to implement the GTSM (Generalized TTL Security Mechanism) as defined at <http://www.faqs.org/rfcs/rfc3682.html>, as well as dynamic filtering as mentioned at <http://www.nanog.org/mtg-0210/gill.html>.

DYNAMIC FILTERING

Internet inter-domain routing relies on BGP (a work in progress: <http://www.ietf.org/html.charters/idr-charter.html>). BGP uses TCP/IP (<http://www.rfc-editor.org/rfc/rfc793.txt>) as the underlying transport mechanism. Using some of the underlying properties of the transport mechanism, the routing hardware can be made more robust in the face of attack. (See figures 2 and 3.)

Each session can be uniquely identified by the standard TCP/IP 5-tuple (for brevity, I will focus on BGP):

- Protocol
- Source IP address (src ip)
- Source Port (src port)
- Destination IP address (dst ip)
- Destination Port (dst port)

The 5-tuple for BGP consists of: (tcp, local ip, local port, peer ip, peer port). Once you have this information, matching packets then becomes a simple task that can be executed on each line card. Packets destined for the router are matched against the filter. If the packet matches the filter, place that packet into the high-priority queue; otherwise, place that packet into the low-priority queue.

The IP Header Format

FIG 2

At session establishment time the 5-tuple looks like this: (tcp, local ip, local port, peer ip, 179). Assuming the router building the filter wins the collision detection, nothing needs to change. If the router loses the collision detection, update the filter as follows: (tcp, local ip, 179, peer ip, peer port). Reverse the filter, because the incoming packet will have local ip and local port that is the peer ip and peer port, and then push the filter out to the line cards.

How does this actually perform in practice? A simple analysis follows.

Any valid BGP packet arriving on any line card will have the right 5-tuple and should be placed in the high-priority queue. Most spoofed DoS BGP packets will not match the filter and will be placed in the low-priority queue. The route processor CPU services the high-priority queue first. This mitigates packet flooding.

Now, assume an intelligent attacker who can fill in the 5-tuple with some knowledge about the router under attack. The following information can be easily gained by tracerouting through the network a few times. This will lead to knowledge of the 5-tuple to the extent of being able to fill in: (tcp, local ip, p, peer ip, p'); p and p' must be guessed to pass the filter and get placed in the high-priority queue. Assuming an attacker with some BGP knowledge, the attacker can guess—depending on how the BGP collision detection is resolved—that either p or p' must be 179. The attacker now must guess the BGP collision winner's local port, which can be anywhere in the range between 1,025 to approximately 64,000.

The total possibilities an attacker must try in picking a valid local port are on the order of $2^{16}-1$, or approximately 64,000. This means that, on average, the attacker will need to try 32,000 times to find the correct 5-tuple, or alternatively only 1 in 32,000 packets sent will make it past the filter and hit the RP CPU. This means now that the attacker must marshal on average 32,000 times greater resources to adversely affect a router. The cost of attacking infrastructure has risen dramatically, while the cost to defend is minor.

The TCP Header as Defined in RFC 793

FIG 3

Stability is most important. Place only the high-priority queue filter for a neighbor once the session is established. Before the session is established, place neighbor packets in the low-priority queue. Trading time for a session to come up over knocking existing sessions down is an operationally correct choice.

There are some shortcomings with the filters as described here. These have to be implemented as close to the line cards as possible. They will serve no purpose if implemented on the RP, as the choke point is well before the RP is even in the equation. There is also an added cost and complexity of implementation.

The Internet has put the control plane in the data plane. Though this has been useful when trying to debug the connectivity between various pieces, the downside has been that the control plane is vulnerable to attacks via the data plane. Some of the suggestions made here serve to harden the control plane from trivial data-plane-based attacks. Network service providers should work with all incumbent and potential vendors to ensure that all devices implement control-plane priority queuing. ☺

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

VIJAY GILL is senior network architect for global network operations at AOL. His previous positions include systems development and analysis at the University of Maryland and senior network architect at UUNET and MFN/AboveNet.

© 2004 ACM 1542-7730/04/1100 \$5.00