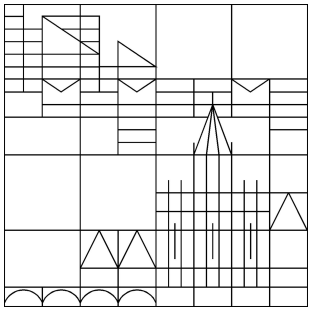

Universität Konstanz

Shifting the interests in high-speed DoS prevention

Marcel Waldvogel

Konstanzer Schriften in Mathematik und Informatik
Nr. 229, März 2007

ISSN 1430-3558

Konstanzer Online-Publikations-System (KOPS)
URL: http://www.ub.uni-konstanz.de/kops/volltexte/2007/2440/
URN: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-24400

Universität Konstanz
Fach D 188, 78457 Konstanz, Germany
E-Mail: preprints@informatik.uni-konstanz.de
WWW: http://www.informatik.uni-konstanz.de/Schriften/

Shifting the interests in high-speed DoS prevention

Marcel Waldvogel

***Abstract*—In early 2000, the Internet world was shocked: Several resource-rich commercial sites were unreachable for several hours, probably due to the actions of a single individual who previously had gained control over many thousand computers world-wide. This shock resulted in a series of proposals how to prevent future disasters. Six years have passed, there is still no consensus on how to improve the situation. In this paper, we propose a new mechanism which also shifts the interests: Involve different stakeholders, which might actually be interested in solving the problem; provide more immediate return-on-investment; focus on end-to-end mechanisms with minimal network involvement; and the absence of a modifications to a large installed base of network equipment characterise our new approach.**

I. Introduction

In early 2000, the Internet world was shocked: Several resource-rich commercial sites were unreachable for several hours, probably due to the actions of a single individual who previously had gained control over many thousand computers world-wide [2]. This shock resulted in a series of proposals how to prevent future disasters. In the past six years, it was tried to reach consensus on how to improve the situation, but to no avail [3]. We believe that the reasons do partly lie in the form of the proposals, as they address the wrong audience. To set the stage, we first identify five components of a DDoS attack:

Zombies. The perpetrator starts collecting nodes to use for his attack, the Zombies, typically hrough a worm, virus or trojan with a remote control interface.

Start signal. One component to influence or identify the perpetrator would be to listen into his control network and/or to inject commands. Our perpetrator thus sets up a series of intermediate agents, to which the Zombies connect and which can be used to conceal the source of the commands. Alternatively, the malware that was injected into the Zombies may include time-dependent instructions.

Attack. At some later stage, the hosts' new master issues the attack command, causing the Zombies to send predefined byte and packet sequences to the attacker, typically at maximum speed. These sequences are typically designed such that they cause maximum effect without being easily identifiable for filtering purposes.

Fake source. To conceal the IP addresses of the Zombies, many attacks try to spoof their source address.

Abort. The victim will try to abort the attack by identifying sources or packet properties and selectively shutting them down.

Distributed Systems Laboratory, Department of Computer and Information Science, University of Konstanz, 78457 Konstanz, Germany, <firstname>.<lastname>@uni-konstanz.de. The work was started while the author was with IBM Zurich Research Laboratory [1].

The manifold approaches at DDoS prevention try to hinder the first four components or improving the countermeasures in the Abort phase. We can classify these approaches into seven categories:

Anti-spoofing. Each router should be able to verify whether this source IP address could legitimately come in on that interface [4]. This also limits several legitimate uses of asymmetric routing, from link sharing to satellite-downlink-modem-uplink scenarios.

Packet information. Routers should include information in each passing packet [5, 6, 7]. Besides breaking fragmentation (or excluding fragments from traceability), it has been shown that attackers can more efficiently insert fake traceback paths than the system can insert real paths, potentially causing a DoS on the system used to analyse traceback information [8].

Router storage. Routers should store a fingerprint of each packet for a short period of time, allowing the return route to be identified when the receiver presents an unwanted packet to the system [9].

Automatic identification. Potential DoS activities should be directly identified at routers, potentially enabling them to directly take measures [10, 11].

Manual blocks. Some ISPs reportedly identify their network's ingress routers which provide a particularly large part of DDoS traffic to the given victim and block all traffic from these ingress routers to the victim in an attempt to minimise DDoS traffic without shutting off too many legitimate sources.

Peer-to-peer systems. Instead of controlling the problem, control the reaction: When a resource is in high demand, create more replicas [12, 13]. A close relative of this proposal is the Google way of throwing enough resources at the problem to handle any load.

Pricing. Let the market forces decide by increasing the price for packets to overloaded destinations [14]. Obviously, such a system would charge the owners of the Zombies, not the attacker. Even though it might be argued that this will teach the careless Zombie owners a lesson, an ISP would have a hard time obtaining the money from such customers or surviving the bad press the attempt would generate.

Besides technical issues, some of which are outlined above, there are also market issues: Those who should invest money in upgrading their equipment and risk more customer support calls or dropping customer satisfaction, among other things, are frequently not those that have an interest in setting up such a system. Typically, the victim's business partner is a hosting provider. This provider can at most do per-attack filtering and has no influence on the source of the traffic. Furthermore, the large consumer ISPs frequently offer only limited commercial hosting services and rarely have customers who form an attractive

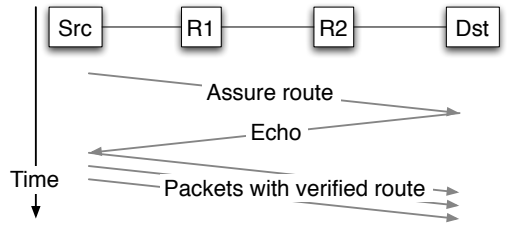

Fig. 1
RarQoS messages: Setup and data

target or who would complain loudly if such an attack occurred; many would not even notice.

Even ISPs which are active in both high-profile hosting and have a large consumer base may not feel the pressure or may be unable to force the responsible departments to team up. The lack of pressure is frequently caused by the minute minority of the attack traffic coming from the own network and the availability of tools and processes that can be used to trace traffic within a single network.

In this paper, we propose a new mechanism which also shifts the interests: Involve different stakeholders, which might actually be interested in solving the problem; provide more immediate return-on-investment; focus on end-to-end mechanisms with minimal network involvement; and the absence of a modifications to a large installed base of network equipment characterise our new approach.

Our lightweight scheme, router-assisted, receiver-driven QoS (RarQoS), allows the parties with vested interest to take action and as a result obtain better quality under heavy load. This not only presents a line of defence against rare events such as DoS, but at the same time can be used to improve QoS, a stronger driving force. The change necessary for the network providers is minimal, their role is essentially limited to only act as a third-party verifier of the sources' claims. All the important decisions remain with the end systems and their users or administrators. It further allows incremental deployment, where already a small deployment will show benefits, something which is generally lacking in other approaches. We believe that this better reflects the interests, market forces, and end-to-end design of the Internet.

II. Router-assisted, receiver-driven QoS

The design of RarQoS diverges from the established DoS prevention path. It was influenced by QoS ideas instead, noting that preventing DoS is just a special case of handling QoS. But as there is no need to provide QoS guarantees, but just a simple form of differentiating between multiple classes of best effort service, it does not suffer from the complexity and state explosion common to many QoS approaches, such as IntServ [15]. It also does not require establishing a mapping between different QoS parameters and contract negotiations, as necessitated by DiffServ [16].

RarQoS uses a simple setup protocol, which requires only the traffic source to maintain per-connection state and include it in every outgoing packet. It thus avoids becoming the target of DoS attacks, as has happened to other protocol features as the maintenance of TCP SYN state for half-open connections or IP fragment reassembly. During the setup, RarQoS-enabled routers record small snippets of data in a designated area of the packet, just enough to be recognised by the same router in later data packets (Figure 1, "assure route"). If the receiver would like to grant this sender elevated priority, it then echoes back these snippets to the source ("echo"). The receiver includes this data in future packets, which allows the routers to verify that the receiver is willing to receive these packets from this source along that path ("packets with verified route"). This setup process can easily be integrated into setup handshakes, such as TCP's three-way handshake.

How are these snippets constructed? Traditionally, routes have been recorded using the IP "record route" option, which records the full IP address of all intervening routers. The probabilistic packet marking schemes [5, 6, 7] distributed this information over multiple packets, requiring additional information for reassembly of the fragments by the receiver. RarQoS avoids both the data expansion and the potential state explosion at the receiver by making the packet marking a mechanism which is in the interest of the involved parties to gain better service. This avoids the need to squeeze data surreptitiously into some hopefully unused areas of the IP header.

A first step would be to record tuples (*hop count*, *verifier code*) into the designated section of the packet. The verifier code consists of a value derived from flow information (e.g. the address/port/protocol five-tuple) and a secret known only to the issuing router. The derivation function must not be invertible by any other party than the router. Potential functions include keyed hashes or encrypting the flow information with the secret. Using 8 bits of verifier gives each router a 255-in-256 chance to identify forgeries, weakening the attacker's effort by a factor of 256.

Assuming both values in the tuple to consist of 8 bits, this would require 16 bits of information per RarQoS-enabled router along the path. Today, paths of $20 \ldots 30$ hops are quite common, this encoding would result in $320 \ldots 640$ bits to be included in each packet, plus some distinguishing header, clearly an undesirable cost-performance ratio.

The information can be halved by noting that the *hop count* fields do not provide 8 bits of information each. Options include delta-encoding the hop-count differences (more effort at the router) or having an index header pointing to the next field to use, to be updated at each RarQoS-step (requiring packet update). As a result, we get down to $8 + \varepsilon$ bits per step, but making the packet forwarding process more expensive.

The process can be further strengthened by having routers set a flag when they recognise a mismatch in the verifier code, as an alert to routers further down the road, that this packet has been tampered with and that it should be forwarded only when there is no congestion (Figure 2, "misbehavior detected bit"). Then, the efforts of all the routers are multiplicative, no longer just additive. For r RarQoS routers using b bits of verifier each, we reduce the chance of an attacker picking an invalid identifier from 1

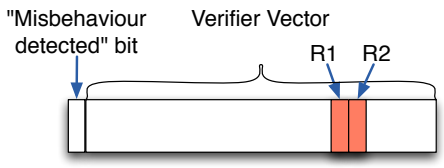

Fig. 2
MINIMAL RARQOS INFORMATION FORMAT

in $r \times 2^b$ to 1 in 2^{rb}.

This comes at a price: The attacker may now do a `traceroute` using the information and see when the "misbehaviour detected" bit is set as an indication to the downstream routers. One approach would be to prevent packets containing assurances (verifiers being built) from being returned. This would require including the assurance only in ICMP error packets, which will not cause further error messages. This would be a kludge at best, many would consider this a major abuse of the Internet protocol.

It turns out that this can be avoided by including the current hop count together with the flow information as an input to the keyed hash or encryption. For `traceroute` or other programs that would like to have packets returned at controlled locations in the network, it is essential to modify the time-to-live/hop-count field of the IP header, breaking the assurance of a packet actually coming from the designated source.

This multiplicative effort allows us to limit the number of bits per RarQoS step to one, the impact of an attack will be reduced by a factor of 2^r, with each router using just a single bit. This allows us to get rid of all counters: The bit is indexed by the time-to-live/hop-count field of the IP header, modulo the maximum number of expected hops, a number which can be determined during the setup message exchange.

The entities enjoying the greatest benefit from this system are now the users/administrators of the systems that communicate. The ISPs also do not have to fear a massive surge in support calls, when they enable this feature, as it will not affect normal operation, only benefit under high load, such as those caused by DoS attacks. The hosting providers will be under the biggest pressure to enable their routers, as they will be under pressure from both the DoS load generated and their hosting customers. But also other ISPs, such as those providing connectivity to home users will have no disincentive to enable this feature. If it requires upgrade of the router hardware, this can be done in the normal router replacement process, as already a few routers spread throughout the path provide a tangible benefit to all involved parties.

Natural extensions of the basic RarQoS described include the support for multiple levels of QoS.

III. CONCLUSIONS AND FUTURE WORK

We described a lightweight scheme where the parties with vested interest need to take action and then obtain better quality under heavy load. The change necessary for the network providers is minimal, their role is essentially limited to only act as a third-party verifier of the sources' claims. All the important decisions remain with the end systems and their users or administrators. It further allows incremental deployment, where already a small deployment will show benefits, something which is generally lacking in other approaches. We believe that this better reflects the interests, market forces, and end-to-end design of the Internet.

Our next steps are to prototypically implement this scheme and gain experience from it. We will also work on addressing issues related to network changes, source mobility, managing the time-driven change of secrets, more compact encoding of the verifier vector, improved revocation of QoS grants, and other refined features.

REFERENCES

[1] Sean Rooney, Christopher J. Giblin, Marcel Waldvogel, and Paul T. Hurley, "Identifying a distributed denial of service (DDoS) attack within a network and defending against such an attack," European Patent Application EP04405438.5, 2004.

[2] Jelena Mirkovic, Janice Martin, and Peter Reiher, "A taxonomy of ddos attacks and ddos defense mechanisms," Tech. Rep. 020018, Computer Science Department, University of California, Los Angeles, 2002.

[3] Rich Pethia, Alan Paller, and Gene Spafford, "Consensus roadmap for defeating distributed denial of service attacks," `http://www.sans.org/dosstep/roadmap.php`, 2000.

[4] Jun Li, Jelena Mirkovic, Mengqiu Wang, Peter Reiher, and Lixia Zhang, "SAVE: Source address validity enforcement protocol," in *Proceedings of IEEE Infocom*, 2002, pp. 1557–1566.

[5] Dawn X. Song and Adrian Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings IEEE INFOCOM*, 2001.

[6] Drew Dean, Matt Franklin, and Adam Stubblefield, "An algebraic approach to IP traceback," in *Proceedings of the Network and Distributed System Security Symposium*, Feb. 2001.

[7] Stefan Savage, David Wetherall, Anna R. Karlin, and Tom Anderson, "Practical network support for IP traceback," in *Proceedings of ACM SIGCOMM*, 2000, pp. 295–306.

[8] Marcel Waldvogel, "GOSSIB vs. IP traceback rumors," in *18th Annual Computer Security Applications Conference (ACSAC 2002)*, Dec. 2002, pp. 5–13.

[9] John Ioannidis and Steven M. Bellovin, "Implementing pushback: Router-based defense against DDoS attacks," in *Proceedings of Network and Distributed System Security Symposium*, Reston, VA, USA, Feb. 2002, The Internet Society.

[10] João B. D. Cabrera, Lundy Lewis, Xinzhou Qin, Wenke Lee, Ravi K. Prasanth, B. Ravichandran, and Raman K. Mehra, "Proactive detection of distributed denial of service attacks using MIB traffic variables – a feasibility study," in *Proceedings of International Symposium on Integrated Network Management*, 2001.

[11] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, "Controlling high bandwidth aggregates in the network," Tech. Rep., AT&T Center for Internet Research at ICSI, July 2001.

[12] Jianxin Yan, Stephen Early, and Ross Anderson, "The XenoService – a distributed defeat for distributed denial of service," in *Proceedings of CERT Information Survivability Workshop 2000*, Oct. 2000.

[13] Marcel Waldvogel, Paul Hurley, and Daniel Bauer, "Dynamic replica management in distributed hash tables," Research Report RZ–3502, IBM, July 2003.

[14] David Mankins, Rajesh Krishnan, Ceilyn Boyd, John Zaho, and Michael Frentz, "Mitigating distributed denial of service attacks with dynamic resource pricing," in *Proceedings of Annual Computer Security Applications Conference (ACSAC 2001)*, 2001.

[15] Robert Braden, David Clark, and Scott Shenker, "Integrated services in the Internet architecture: An overview," Internet RFC 1633, June 1994.

[16] Steven Blake, David Black, Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss, "An architecture for differentiated services," Internet RFC 2475, Dec. 1998.