The XenoService – A Distributed Defeat for Distributed Denial of Service

Jianxin Yan, Stephen Early, Ross Anderson

Computer Laboratory, Pembroke Street, Cambridge CB2 3QG, UK
forename.surname @ cl.cam.ac.uk

Abstract. Distributed Denial of Service (DDoS) attacks have become a serious problem since the second half of 1999. They are at heart a manifestation of what economists call the 'tragedy of the commons': while everyone may have an interest in protecting a shared resource (Internet security), individuals have a stronger motive to cheat (connecting insecure computers). So we doubt that some of the proposed technical countermeasures will work, as they take insufficient account of economic forces.

In this paper, we discuss a possible remedy. The XenoService is a distributed network of web hosts that respond to an attack on any one web site by replicating it rapidly and widely. In this way, a mom-and-pop antiquarian bookstore that comes under a DDoS attack can within a few seconds acquire more network connectivity than Microsoft, so that it can absorb a packet flood and continue trading. The XenoService uses Xenoservers, a technology developed at Cambridge for distributed hosting of latency- and bandwidth-critical network services. They can be deployed in such a way as to provide effective economic incentives for the principals to behave properly.

1 Introduction – Distributed Denial of Service

A distributed denial of service attack exploits a number of subverted machines (attack bots) to launch a large coordinated packet flood at a target. Since many bots flood the victim at the same time, the traffic is more than the target can cope with, and because it comes from many different sources, it can be very difficult to stop [1].

However, apart from the fact that they are launched in a distributed way, DDoS attacks exploit much the same vulnerabilities as previous service denial attacks. For example, Trinoo launches UDP flood denial of service attacks, and Tribe Flood Network can launch SYN floods, ICMP echo request floods, and ICMP directed broadcast ('smurf') denial of service attacks [2].

It is widely believed that at least three methods could help prevent DDoS attacks [1, 3, 4, 5]:

– Secure hosts: if all hosts on the net were secure, hackers would have nowhere to install their attack bots.

– Egress filtering: IP spoofing is a fundamental technique in DoS attacks, as it makes it difficult to locate and neutralize the attack bots. If network service providers stopped it, life would get very much easier.
– Fixes for specific vulnerabilities: for example, SYN floods can be made much less effective using SYNcookies, and smurf attacks are becoming less common as social and economic pressure on system administrators reduces the number of smurf amplifiers.

None of these will provide a complete solution.

First, DDoS is a system problem, not a matter of any specific Internet technology. It is not unknown for protesters who simultaneously send faxes to a politician to clog up his fax service; and a fuel shortage can cause so many calls to a rail timetable service that it becomes completely overloaded.

Second, Orange-book-evaluated systems are too expensive, have too few features, don't run popular applications, and are generally unavailable outside the defense community. Although there are strong economic incentives in favour of more secure platforms (principally so that copyright could be enforced more stringently), there are even stronger incentives in the other direction: the primacy of time-to-market dictated by the first-mover advantages conferred by network economics precludes long development and testing cycles, while the pressure to sell to value rather than to marginal cost of production dictates ever more features and product versions.

Third, the economic incentives needed to prevent DDoS attacks in particular are all but absent.

2 Incentive Issues

The first response that each user has, when facing a new vulnerability or attack, is to ask, 'How can I prevent this type of attack happening to me?' But in the case of DDoS, there is little users can do to protect themselves directly, as the attacks exploit vulnerabilities in other people's systems. Other types of attack, such as viruses, can typically be stopped by installing suitable defensive software, which might cost $100 per machine. This is fair enough; but stopping DDoS attacks using this kind of approach might involve most of the 200,000,000 machines on the net acquiring defensive software. This pushes the price up into the billions. It also changes the underlying economic incentives. I may be prepared to pay $100 to prevent myself being attacked; I am much less likely to spend this money to stop Amazon or Microsoft being attacked. It may be rational for me to keep my $100 in my pocket and hope that I am not one of the small minority that become a target.

This is an example of what economists refer to as the 'Tragedy of the Commons' [6]. If a hundred peasants are allowed to graze their sheep on the village common, where the grass is finite, then whenever another sheep is added its owner gets almost the full benefit while the other ninety-nine suffer only a very small disadvantage from the decline in the quality of the grazing. So they aren't

motivated to object, but rather to add another sheep of their own and get as much of the declining resource as they can. The result is a dustbowl. In the world of agriculture, this problem is tackled by community mechanisms, such as getting the parish council to set up a grazing control committee. One of the challenges facing the world of computer security is to devise the appropriate mix of technical and organizational mechanisms to achieve the same sort of result that was already achieved by a typical tenth-century Saxon village, only on the much larger and more diffuse scale of the Internet.

3 Liability Issues

One of the most thought-provoking responses to the DDoS problem has come not from a computer scientist but from an economist, Hal Varian. His view, expressed in [7], is that the solution will be to force the owners of systems from which attacks originate to pay for the damage they cause. Whether this can be achieved through the courts, or will require fresh legislation, is an open question and may vary from one country to another.

Of course, the great majority of users have no idea how to defend their computers. So Varian proposes that the costs should fall first on the network operator, following the established principle in tort law that liability should fall on whoever is in the best position to manage it. But even if ISPs end up bearing primary liability for claims from DDoS victims, there still has to be some cost to users who get hacked, or they won't have any reason to take care. So an ISP's standard contract could say something like: "If your machine hosts a DDoS attack, we'll terminate your service. That means you'll have to go to the hassle of geting a new email address. It's up to you whether you buy security software or just take a chance".

But while this kind of approach might provide a solution in the long term (and in the even longer term someone might sue Microsoft over the vulnerabilities in their software), the problem facing web businesses is how to deal with the problem today. If your business plan calls for five nines availability, how can you deliver it in a chaotic and sometimes hostile world?

4 Replicated Solutions

One of the possible ways of blocking DDoS attacks is to replicate the service we wish to protect. But there is an economic problem here. If instead of one web server with a capacity of 10,000 transactions a second, we have 10 servers at 1000 tps or even 100 servers at 100tps, it will be very much more expensive. It also provides essentially no protection against flooding. A website typically just has one DNS name, and even if technical measures (such as dynamic DNS [8]) are used to distribute the load to different machines, DNS will deliver their IP addresses to the attackers just as well as to the genuine customers.

What's actually needed is that a site could respond to attack by acquiring very much more bandwidth. Huge web sites such as `www.microsoft.com`, with

hundreds of servers and megabits of network bandwidth, have the capacity to absorb and even ignore all but the largest DDoS attacks. A means for the small-to-medium-sized web site to acquire such a capacity could go a long way to solving the DDoS problem. It could also be useful in other ways. For example, from time to time our own lab website gets a huge number of hits – typically when a research result is cited in the mass media or on slashdot. Dynamic replication isn't entirely new – Freenet uses it as a censorship resistance mechanism, whereby frequently requested (and thus possibly controversial) documents become widely replicated [9] – but as far as we are aware this is the first proposal to use it as a distributed means of defeating DDoS attacks.

But dynamic replication of a website isn't enough on its own.

Let's go back to our pre-Internet example, of a politician's office being swamped by thousands of faxes from lobbyists. What the politician might want to do is to have all his faxes intercepted at some earlier point in the network. But even if it were possible for him to send out a message to every telephone exchange in his constituency saying 'please intercept all faxes for number X, tar them up, gzip them and ftp them to IP address Y' the problem isn't solved as his staff still have to wade through all the files. What's really wanted is some means of combining interception with filtering. It may be, for example, that of the 50,000 faxes sent that day to his office, 49,800 are the same document, which people have printed out from a lobbyist web site, signed, and faxed in. In this case, the incoming traffic should be separated into two bundles.

At this point the analogy starts to wear thin, as the common solution used by prominent people is to have separate public and private fax numbers. If the former jams, then so what: the politician may find it convenient to be able to claim that he received a lot fewer faxes.

Commercial websites have different requirements. To be effective, a web site should be visible to everyone on the Internet, and although 'obvious' attacks such as SYN and ICMP floods can be filtered automatically, it should continue to serve web pages in response to all well-formed requests if this can at all be managed.

5 The XenoService

The technology which we propose to adapt as a resilient platform for web service is the Xenoserver. Xenoservers were developed at our laboratory as a means of distributing temporally sensitive applications such as multimedia servers and multiplayer games so as to circumvent long communication latencies or avoid transferring data over congested or expensive network links [10]. The business model is that ISPs run a number of Xenoservers on which they rent capacity to service providers. The Xenoserver software enables the accounted execution of untrusted code; loosely, it may be thought of supporting mobile server code in the same way that Java supports mobile client code. As well as confinement (to prevent attacks or failures resulting from badly written CGI scripts), a Xenoserver must address the problem of resource management. This is achieved

in the Cambridge prototype by running it on top of Nemesis, an operating system designed to support quality-of-service guarantees and thus able to prevent applications mounting denial-of-service attacks on it.

This makes the Xenoserver concept well adapted for assuring quality of service of a different kind, namely resilience against flooding attacks.

The business model we propose is as follows. A number of ISPs worldwide install Xenoservers and offer a resilient web hosting service at a premium price. The quality of the service is monitored and, if it starts to deteriorate as a result of a surge in demand, the web site is at once replicated to other Xenoservers whether locally or in other ISPs. In the latter case, distributed DNS techniques can be used to eliminate bottlenecks.

As ISPs get paid for the resources they supply, there is an incentive for them to install and maintain a sufficient number of Xenoservers to meet whatever service levels are set in their contracts with the system. As the ISP hosting the attacked system can rent additional capacity as required, it needs less installed capacity to cope with everyday demand fluctuations and should in any case be able to insure against the costs of absorbing particularly severe DDoS assaults. There is a potential perverse incentive if a large ISP could earn significant resource rental income as a result of attack traffic from bots hosted on its own network; we propose that this be dealt with by contractual mechanisms such as by making each participating ISP liable for some of the traffic from its own network.

The use of Xenoservers can be justified in any event as a means of efficiently distributing and multiplexing surges in web traffic across participating servers. As the installed communications capacity of the Internet continues to grow much more rapidly than computing capacity (with a doubling time of perhaps five months rather than the eighteen months commonly cited as Moore's law), such mechanisms will become increasingly economic. As a means of absorbing attacks, we feel that they may be exceptionally effective.

In electronic warfare, it is an established principle not to let the jammer know whether, or how, his attack is succeeding. So in military communications, it's usually better to respond to jamming by dropping the bit rate rather than boosting power; if the bit rate is invisible to the enemy (as it's masked by a spreading sequence) then with luck he will not know he's having any effect, so he'll give up rather than continue to expose the jamming platform to physical attack. In the DDoS context, if the only effect of an attack is a rapid replication of the service, with effects that are all but invisible to the attacker, then hopefully DDoS attacks will go rapidly out of fashion.

Of course, it is not necessary for a XenoService to be operated as a public commercial service. It can be used in closed environments too. One that comes naturally to mind is government. While there may be frequent flood attacks on the web servers of the Department of Justice (for example), a typical government has a vast range of other web servers and thus has the natural capacity to replicate a service under attack to huge numbers of other machines.

In many cases, only parts of a corporate web site will be designed for rapid

XenoService replication. The static parts of a site, such as catalogues and price lists, are intrinsically easy to replicate while interactive services such as order entry are harder. There are many options for the designer, ranging from a centralized credit card transaction capture service through a message saying something like: 'Sorry, the items you have requested are temporarily out of stock. Can we email you in a few days when we get another delivery?' A programmer selling software online might want the former service, so that if he got 'slashdotted' he could ship a million copies of his code and collect the cash without having to hire new staff. A mom-and-pop antiquarian bookstore would probably prefer the latter as there's only so many books they can wrap properly and ship in any working day.

6 Conclusion

We believe that web site replication may provide a large part of the solution for distributed denial of service attacks, provided it's done in an intelligent way that uses appropriate technology and a sensible business model. We've proposed what we believe to be a workable solution.

References

1. CERT, Results of the Distributed-Systems Intruder Tools Workshop, Software Engineering Institute, Carnegie Mellon University, `http://www.cert.org/reports/dsit_workshop-final.html`, December 7, 1999
2. CERT, Distributed Denial of Service Tools, CERT Incident Note IN-99-07, `http://www.cert.org/incident_notes/IN-99-07.html`, updated on December 8, 1999
3. Terry Dawson, Preventing Distributed Denial of Service Attacks, `http://oreilly.linux.com/pub/a/linux/2000/03/10/netadmin/ddos.html`, Mar 9, 2000
4. SANS Institute, Consensus roadmap for defeating distributed denial of service attacks, `http://www.sans.org/ddos_roadmap.htm`, Feb 23, 2000
5. SANS Institute, Help Defeat Denial of Service attacks: step-by step, `http://www.sans.org/dosstep/index.htm`, Mar 23, 2000
6. WF Lloyd, *'Two Lectures on the Checks to Population'*, Oxford University Press (1833)
7. H Varian, Managing Online Security Risks, Economic Science Column, The New York Times, June 1, 2000, `http://www.nytimes.com/library/financial/columns/060100econ-scene.html`
8. P Vixie (ed.), S Thompson, Y Rekhter, J Bound, RFC-2136: Dynamic updates in the domain name system (DNS UPDATE), 1997
9. I Clarke, The Freenet Project `http://freenet.sourceforge.net//`
10. D Reed, I Pratt, P Menage, S Early, N Stratford, Xenoservers; Accounted execution of untrusted code, Hot topics in Operating Systems, 1999; `http://www.cl.cam.ac.uk/Research/SRG/netos/xeno/`