TUTORIAL

What DDoS it all Mean?

Nick FitzGerald
Computer Virus Consulting, New Zealand

Unless you were well out of touch in early February this year, you must have heard about the day the Internet died. 'Cyber-attacks batter Web heavyweights' read one headline and the story ran endlessly in on-line, print and broadcast media for more than a week. Odd that the NASDAQ reacted by strengthening…

Distributed denial of service, or DDoS, attacks disrupted some of the largest Web sites – *CNN*, *MSN*, *Yahoo* and others – sites designed to serve millions of pages per day. So what are DDoS attacks? How might they affect you and what should you do to avoid them?

History Repeating

Network denial of service (DoS) attacks are easy to understand. A malicious user attempts to exhaust some limited resource – usually network bandwidth – to deny others access to a network-based service. Apart from bandwidth consumption, other forms of DoS attack are possible. Specific versions of some network software are known to have bugs that render them unstable when 'odd' packets, or packet sequences, are received. An attacker could utilize such a weakness to DoS a site known to run an affected version of the vulnerable software.

Historically, someone planning a DoS attack would obtain code to implement an attack the intended victim would be vulnerable to, or write an implementation of the chosen vulnerability from a description of it. One of the risks of discovery would be that the attacker could lose their account on the machine launching the attack (if, in fact, the attack was ever traced). Amelioration of that risk was often accomplished by the attacker cracking some other host first, then launching the DoS attack from there.

An easily compromised system, giving the attacker root or administrative privileges, has two advantages. First, it moves the attacker one step further from possible banishment since it is not the attacker's own system. Second, and more importantly, the attacker further reduces the chance of being discovered because if the site was easily compromised (say, with an old exploit), by definition it is a poorly administered site. Also, with root access, the attacker could alter system logs and the like, further obfuscating the real source of the attack, or at least the person responsible for it.

As widespread DoS'ing of sites became something of a sport among elements in the hacking underground, a new challenge arose. With the attacks becoming more common, some potential targets were increasingly armoured against one or more of the well-known attacks, through improved firewall and router configurations and use of network intrusion detection systems (NIDS). Further, the very large (and, therefore, most brag- and news-worthy) sites were daunting targets because of the sheer bandwidth a successful attack would have to use up.

Distributed DoS attacks were the obvious next step, solving both problems by implementing several attacks in one tool and providing a means to coordinate and synchronize attacks from very large numbers of machines. Given the alternative for an attacker having to maintain a motley crew of tools, and possibly accomplices to help launch attacks from a handful of compromised sites, the advantages of DDoS tools should be clear.

Are DDoS Tools New?

From the media coverage, you would probably assume the answer to this question is 'Yes', but they are not that new. The concept has been around for some time, but although there have been examples of DDoS and other distributed hacking tools, they certainly have not been common.

In September 1999's Editorial I mentioned a Trojan that had become widely distributed by mass-emailing. When the attached program was run, rather than installing the latest security patches to *Internet Explorer*, it installed a program to monitor whether an active Internet connection existed, and if so, sent a large amount of abusive email to the *Bulgarian National Telecommunications Company* and ISP.

Over the following few months, variants with different network-based, resource-wasting attacks were also seen. These reportedly caused a great deal of inconvenience to the real target – the Bulgarian ISP – but typically were of nuisance value only to those tricked into running them guilelessly. These Trojans may have implemented the first widespread, programmatic DDoS attacks.

Released shortly after Melissa, X97M/Papa contained not only a mass-email distribution mechanism, but a distributed 'ping' DoS attack directed at two machines of a well-known network security researcher. Perhaps fortunately for the target of Papa's ping flood, Papa did not become anywhere near as widespread as Melissa.

Between the appearance of these two early, simple, PC-based DDoS agents, W97M/ColdApe was released. As the target of that virus' email payload, it was very ineffective if it was intended as an email DoS attack against me or the magazine. So ineffective, in fact, I would not have considered this a possible motive for that part of its payload. However, several newsgroup posts by a virus writer affiliated with one of ColdApe's writers suggests that the pro-virus/VX underground saw it as such.

Outside the world of personal computers, DDoS tools started to appear in the wild in early to mid-1999. The best known are Trinoo (or Trin00), Tribe Flood Network (TFN), Stacheldraht and a recent update to TFN known as TFN2K. These tools have gained quite some media coverage, probably because they have been closely analysed by security experts and source code for them is readily available. However, in a recent article, the hacker known as Mixter (author of TFN and TFN2K) claimed to know of four other DDoS tools, that he named. They have not been publicized, but may be in use, and how many other DDoS tools are in use that Mixter does not know of?

The Shields are Down Cap'n…

So how do these recent network DDoS tools work? Perhaps the most important thing to realize about them, which the mainstream media has mainly overlooked, is that there are really two separate targets in these attacks. Obviously the big-name Web sites in the early-February headlines were targets, but they could not have been targeted (as successfully) without the first set of targets – a large number of poorly secured and under-administered Internet servers.

Trinoo, TFN, TFN2K and Stacheldraht have similar general architectures, varying in implementation details. All four have two software components installed on compromised machines. Let us refer to these two components as 'master' and 'slave'. An attack with any of these begins with the attacker locating and compromising many suitable machines, on which the slave is installed. A few machines are also compromised and the master software is installed.

Together, these machines constitute an attack network. Launching an attack is simply a matter of contacting the master(s) and providing them with the address(es) to attack and the type of attack to use. Trinoo is the simplest of these well-known DDoS kits and it only implements one network DoS attack – a UDP flood. The others add ICMP and SYN floods, and the Smurf attack. Most of these attacks either depend on IP spoofing (sending packets with forged source addresses) or use spoofing to confuse and slow diagnosis and resolution attempts by the target further.

Captured and/or released source code for these kits shows various 'fingerprints' the tools leave in a system or on a network. Later tools, especially TFN2K, are more sophisticated in this regard, making several attempts to disguise their presence further. Some of these obfuscations include: the encryption of all control messages between masters and slaves with compile-time keys; depending on probabilistic delivery of control messages, so the slaves never respond to masters, and; use of ICMP packets which extant network tools have unsophisticated handling of and that generally are allowed through firewalls.

As this article was being completed, reports arrived of a US university discovering a Win32 port of the Trinoo slave installed and active on PCs in its student residence network. All the affected PCs had also been compromised with BackOrifice, suggesting that either BO has been 'bundled' with this Trinoo executable or Trinoo was installed once the PCs were accessible via the BO client.

Protecting Yourself

The bittersweet irony of these DDoS tools is that you cannot protect yourself. The best an individual site or firm can do is ensure its machines are as secure as they can be. After that, you can only hope the 'white hats' find the easy exploits in a timely fashion relative to the 'black hats', then install any security patches your vendor produces.

Having done all that, you are protected as best you can be against becoming a DDoS slave, but you can do little about attacks that may be launched against you with these tools. Depending on various technicalities, there are some newer router and firewall options that can reduce the impact of some of the DoS attacks the slaves launch without rendering your network unusable for its intended purposes.

NIDS have been updated to detect traces of Trinoo, TFN, TFN2K and Stacheldraht in the network. If you have a NIDS and have updated its profiles, do not be complacent that this is sufficient to detect these tools. They are available in source form and tend to be in the hands of more sophisticated users than the script kiddies. The source recommends users alter many of the defined constants precisely to avoid such 'signature' scanning methods. Evidence that attackers are heeding this advice is available in the Win32 port of Trinoo mentioned above. It does not use the 'default' ports described in the first detailed analysis of Trinoo, although from a rudimentary first look at the program it appears that the rest of the Trinoo protocol is fairly standard in this case.

Not to be left out, several anti-virus developers have added detection of the 'big four' DDoS tools to their products. This, of course, raises even more problems than the NIDS face. A good NIDS may be able to detect some tell-tale changes in traffic flow shapes, 'odd' packet types and the like, raising an alert for the network manager to apply some human intelligence to a trace. However, with the tools distributed as source, and intended for building on many systems, imagine the number of combinations of compilers, linkers and strippers. Cross that with the number of standard libraries, allow that two (or more) different sets of development tools are available for most of the likely target systems and factor in how many versions of these tools? We are talking a staggering number of potential binary variants, and that is before allowing that attackers may alter the known code, which they *are* doing.

The machines that most need detection added are the ones that responsible, concerned admins cannot affect. Your best defence is to secure your own sites and harden your network boundaries against the known attacks. Finally – and marketing departments will not like this – you had best hope that your Web site or company is not interesting or newsworthy enough to be targeted!