RIK FARROW

musings

Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security* and *System Administrator's Guide to System V,* and editor of the SAGE Short Topics in System Administration series.

■ *rik@usenix.org*

IT'S A BLUSTERY SPRING DAY AS I write this column. I've been reading a great new book about Internet Denial of Service (see details in [1]), and I have to confess I am depressed by what I am reading. The outlook just isn't very good, for so many reasons.

To top things off, some people in the German branch of the Honeynet Project have published a new paper about their experience collecting bots [2]. Their paper describes some of the bots captured, as well as information about the use of IRC for command and control (C&C). The sizes of botnets uncovered were not that enormous (well, just tens of thousands in some cases), but as the paper points out, a botnet of 1000 agents, with an average Internet connection speed of 128Kbps, can easily swamp a target with a 100Mbps connection to the Internet.

I haven't really studied the newer distributed denial of service (DDoS) agent software much in recent years, not since the beginning of the century. Back in late 1999, people were worried about big DDoS attacks being used to take out large parts of the Internet. When New Year's Eve passed without incident, people breathed a sigh of relief. But that relief was short-lived, as heavily publicized attacks on commercial Web sites in February 2000 showed.

The early DDoS tools, like Trinoo and Tribe Flood Network, have been replaced by newer, much more flexible tools. Chief among these are descendants of Agobot [3], a bot written in excellent C++ that can be compiled for either Windows or Linux. Agobot uses IRC channels for communications, unlike the earlier DDoS agents that relied on receiving commands from handlers. The commands used with older DDoS agents had easily recognizable signatures that were soon included in IDS software. Researchers also wrote tools that could probe for DDoS agents. But the use of IRC implies that any network that permits outgoing IRC connections will also permit DDoS agents to receive commands and report to the person running the botnet.

I do need to step back for a moment and define my terms. In the world of Internet Relay Chat (IRC), bots, short for robots, were originally network programs that would stay connected to an IRC server and thus stay active in a particular channel. The bot would perform services for its owner—for example, bestowing special privileges that the owner would lose if he left the channel. Bots run on systems other than the bot owner's, so that a denial of service attack against the bot owner's system would leave the bot still running [4].

Over time, bots gained additional abilities. Those who frequent IRC channels like #hack often fight over control of the channel, and one proven method of knocking someone out of a channel is to use DDoS attacks. A person who can amass a large network of bots (a botnet) can use this network to flood any adversary's Internet connection at will.

But why stop there? The Honeynet paper goes on to describe the many other features of advanced bots. They include the ability to execute any command, update their own software, hide themselves, sniff the network, log keystrokes, launch worms (like the Witty worm), and relay spam. Bots can be used to steal identity information, as well as license information for games and software (Agobot is big on this). Agobot, when running on Windows systems, also attempts to disable firewall and anti-virus software.

Botnets have been used for other financial purposes than simple identity theft. People have used them as a blackmail threat, one that can easily be demonstrated by launching a short flood against the target. Some botnet owners will hire out their botnets for DDoS attacks, or to spammers for relaying email through the use of SOCKS (the proxy server by David Koblas announced during the 1992 USENIX Security Symposium).

The German Honeynet Project paper collected information about bots by setting up several GenII [5] honeynets with Windows victims. They had decided on Windows systems as targets based on the amount of scanning detected for Windows-specific services. In their paper, the authors claim that Windows XP (SP1) and Windows 2000 were, by far, the most popular hosts for bots, followed by other Windows versions. They would collect software installed on the honeypot and reinstall the OS each day. On average, the Windows box would be owned in 10 minutes. In one instance, a box was compromised within three seconds after being connected to the network.

The group later designed a special program, mwcollect2, that simulated vulnerabilities and would download malware when commanded to do so by the exploit. This tool made it much easier to collect bots and other malware.

Over four months of research, the German Honeynet group tracked over 100 botnets that used IRC for C&C. Through the use of IRC JOIN messages, they saw 226,585 unique IP addresses of bots connecting to the IRC channels the group was tracking. This number is deceptive, in that there is no way of knowing if the bot's host was assigned a different IP address over time by DHCP. Also, many of the IRC servers used, especially by the more sophisticated botnet owners, were hacked so that they would not provide JOIN messages or accept commands that could list the IP addresses of participants. But the researchers estimate that at least one million hosts have bots installed.

Defenses

The Honeynet Project paper discusses the attackers. The Internet DoS book talks about attack tools, but focuses on defensive techniques. Imagine that you want to defend your site against DDoS attacks? What will you do if the attacker wants to send, on average, 1Gbps of reasonable seeming traffic at your network. Remember that this volume requires 10,000 bots and does not appear to be impossible for some attackers. The German researchers monitored one botnet with 50,000 hosts.

If you look at the information about the duration of DDoS attacks [2,6], most, but not all, attacks are short-lived. Most bots and agents accept time periods measured in seconds, with many attacks lasting only minutes. But some attacks can go on for weeks. I find this distressing to consider.

Chapter 5 of the DoS book includes an excellent discussion of where to locate DDoS defenses. While it would be ideal to filter out attacks at the source, for example, you will quickly realize, as the authors point out, that you do not control ISPs. If edge networks and ISPs would, at the very minimum, enforce ingress filtering by permitting only non-spoofed source addresses from their clients' systems, we could end most source address spoofing. Ingress filtering is one of the simplest technologies, one embodied in an RFC back in 1998, but generally not implemented even today.

Stopping attacks that don't use spoofed source addresses is much harder, even at the source. If the botnet owner decides to launch an attack that uses spidering of a Web site, just having thousands of clients attempting to walk your Web directories will, in itself, be a devastating attack (for most servers). And this attack uses legitimate appearing traffic, not something that an ISP would be able to filter out, if the ISP even noticed.

The core routers appear to be a logical place to stop floods. And while there is some research into this, as well as some working examples, the Internet is a loose federation of networks, and the companies that control the core are competitors. Expecting these companies to cooperate is expecting a lot. You might think that it would be in the interest of the owners of core routers to reduce floods, but the normal traffic seen by their routers is a flood, and the noise of extra traffic gets buried in the background.

That leaves defenses close to the target—your own site. Like just about everything in security, it comes down

to you protecting your own site. The authors suggest many things, including looking for network choke-points, having excessive bandwidth, adding more servers when the load is heavy, and installing patches (some DoS relies on buggy software). But you should also be prepared for a DoS attack. You need to be able to monitor and analyze network traffic at the edge of your networks. Monitoring implies that you have practiced doing this and can easily capture this information and know how to interpret it.

With this information in hand, you can communicate your plight accurately to your upstream ISP, who should be willing to install temporary filters for you. The ISP might even want to communicate with its own providers, pushing back the attack even further.

I will not attempt to duplicate the information in the Internet DoS book here. It has sections appropriate for managers, as well as more technical chapters. I was pleased with the clear and logical prose, even as I was often depressed by the logical implications.

The Internet, as it is designed, accepts any traffic, as long as it complies with minimum standards (a functional IP header). It is fruitless to hope that there are any easy solutions in sight. And that certainly includes solutions that suggest revising IP to defeat DDoS. For the most part, the Internet just works. DDoS and spam are certainly enormous nuisances, but not ones that will by themselves destroy the greatest network ever.

But they sure do make me wish that we had better tools for combating these attacks.

REFERENCES

[1] Jelena Mirkovic, Sven Dietrich, David Dittrich, and Peter Reiher, *Internet Denial of Service* (Prentice Hall, 2005), 372 pp.

[2] Honeynet Project, "Know Your Enemy: Tracking Botnets," http://www.honeynet.org/papers/bots/.

[3] I picked out one variant of Agobot. There are many others: http://www.sophos.com/virusinfo/analyses/w32agobotli.html.

[4] David Brumley, "Tracking Hackers on IRC," *;login:*, http://www.usenix.org/publications/login/1999-11/features/hackers.html.

[5] Honeynet Project, "Know Your Enemy: GenII Honeynets," http://www.honeynet.org/papers/gen2/index.html.

[6] David Moore, Geoffrey Voelker, and Stefan Savage, "Inferring Internet Denial of Service Activity," http://www.caida.org/outreach/papers/2001/BackScatter/.