;login:

THE MAGAZINE OF USENIX & SAGE

July 2001 • Volume 26 • Number 4

inside:

MUSINGS
by Rik Farrow

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

musings

by Rik Farrow

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*.

<rik@spirit.com>

When you get to read this, it will be in the depths of summer. But, today it is May Day, and the Chinese have proved to be a disappointment.

You might recall the ruckus stirred up when the US refused to say "we're sorry" about the death of a Chinese fighter pilot who collided with a US spy plane 70 miles off the coast of the Chinese mainland in April. As an afterthought, some Chinese hackers threatened a coordinated attack against the US between May 1 and May 7. Both days are important: May 1 is International Workers' Day, and May 7 is when the US sent three cruise missiles into the Chinese embassy in Serbia (oops). At least that time, the US said "sorry" right away.

I had thought that perhaps something would really happen. After all, the "1i0n worm," a worm that was (and perhaps still is) exploiting Linux systems sends /etc/shadow off to an address at china.com (which is registered in the Asia Pacific, at least), and was written by a Chinese hacker group of the same name. The Lion worm exploits BIND 8.2 (not again!!), but only on Linux systems running on x86. A portion of its install includes the setup of more automatic scanning, the t0rn rootkit, as well as DDoS agents, like trinoo and TFN2K.

If the Chinese were really using a worm to exploit and collect Linux systems worldwide, and these systems now had DDoS agents installed, they might have "systems to burn" and create a really interesting set of floods in the US. But, nothing has happened (so far). Also, the tools installed, especially trinoo, are really primitive compared with later DDoS agents, which include the ability to update themselves and execute any command as root.

More FUD

The other interesting event of the day was the issuance of CERT advisory CA-01-09 (*http://www.cert.org/advisories/CA-2001-09.html*). For one thing, this was more of a paper than an advisory. I wrote to a friend who works for CERT and suggested that they add an "Executive Summary," so that people will have at least a clue of what it is about and how important it is.

And really, I wondered, how important can it be? CA-01-09 explains some issues in how Initial Sequence Numbers (ISN) are generated, and how studies by Tim Newsham (*http://www.guardent.com/comp_news_tcp.html*) and Michal Zalewski (*http://razor.bindview.com/publish/papers/tcpseq.html*) show that the methods used by many vendors are insufficient to guard against even a "weak" attack that relied on guessing the ISN. I have no argument with the analyses, which date back to seminal papers by Robert Morris (1985, *ftp://research.att.com/dist/internet_security/117.ps.Z*) and Steve Bellovin (1989, *http://www.research.att.com/~smb/papers/ipext.ps*). How could I argue against these guys?

But what I was wondering about is just how relevant this issue is to current configurations of various servers' OSes. What got people's attention in 1994 was the very interesting attack against Tsutomu Shimomura's home network on Christmas day, the very attack that led to the search for Kevin Mitnick (who in all likelihood did not create the attack and might not even have launched it). The attack actually turned the warnings of Morris and Bellovin into something real for the first time. You can find a variant of this attack named rbone2 on some exploit sites even today.

The attack relies on finding a pair of systems that have a trust relationship and rsh servers that are reachable by the attacker. Now, anyone with any security training should be aware that using rsh is a very bad idea. This lapse led me to believe that Shimomura could not have been a security guru since he was using rsh. What Shimomura had done was use TCP wrappers to provide access control based on the IP source address, and then allowed root access between his pair of trusted Sun systems. That is what the attacker exploited using a tool like rbone2.

The tool first uses a SYN flood to "disable" a port on the trusted server. Before this attack, a SYN flood was something that happened to busy Web servers (also a new thing in 1994). Most servers had a connection queue for each port that was only about ten entries deep. If an attacker sent TCP SYN packets with spoofed source addresses, the server's TCP stack would send SYN-ACKs, then wait for responses that would never come, eventually timing out after 75 seconds. So, all an attacker had to do to SYN flood a service was to send enough packets to keep the connection queue filled. The attack tool sent 20 packets.

Next, the tool sent more SYN packets, but this time to the trusting host, and using the tool's real IP address. This allowed the tool to collect ISNs from the soon-to-be victim. Before UNIX vendors bothered to generate pseudo-random ISNs, ISNs were VERY predictable, with each subsequent ISN being 128,000 greater than the previous (unless more than one second had passed, which would also increment the value by 128,000). The attack tool collected a sequence of ISNs, resetting the connection each time instead of letting it complete (which prevents TCP wrappers from ever being invoked and logging the failed connection).

Now, the attack tool has a potential value for the next ISN and can proceed. The attack tool spoofs the IP address of the trusted server and sends a SYN packet to the victim. The victim responds to the trusted server, but the port used is the one that was SYN flooded, and the packet is discarded. Then, the attack tool sends a spoofed reply, which must include the victim's ISN (plus one) in the response if it is to succeed. If the attack tool has guessed the victim's ISN correctly, it can finish the attack.

Keeping in mind that the attacker never sees any response packets, what can be done to the victim? The attack tool sends packets to execute commands as root to open a hole in the victim's defenses, allowing remote access as root (for example, `echo ++ >> /.rhosts` is part of this attack). The only way the attacker can tell if the attack has succeeded is to try an rsh to the victim, and if this works, the attack has succeeded. If not, well why not try again until it does work?

My own problem with this is twofold. How many people would consider using the r commands and trust relationships today? SSH provides a drop-in replacement for the r commands that includes digital signatures that positively identify both the client and the server to each other. So spoofing the source IP address won't work anymore. But perhaps people are still using the r commands.

The advisory also mentions TCP hijacking, but this requires more than guessing the ISN. You must also know the socket information, presumably by sniffing, in which case you don't need to guess the ISN.

The CERT advisory points out that even if the ISN gets incremented with pseudo-random numbers, it is still possible to guess the increment, given enough attempts. The advisory points out that given a series of ISNs, they will tend to fall around an "average" value over time, and given enough attempts, the attack tool could still guess an ISN. Sure. Why not?

What is easier to do, and more likely, is to guess an ISN and send RESETs to break existing connections. In the case of shutting down an existing connection, you still need the socket information. If the attacker has that, the ISN (actually the acknowledgment value) only needs to be within the TCP window in order to be accepted, and then the RESET flag will cause the connection to be terminated. If you are curious about typical TCP Window values (used by the receiver to control how many bytes of data the sender may transmit without receiving an acknowledgment), check out the nmap-os-fingerprints file (get nmap from *www.insecure.org*), and you can see that this value varies from as small as 512 bytes to as much as 32K-1 (Windows NT/SP3, W=7FFF).

But this is still a problem. How does the attacker know the socket information (source port, source IP address, destination port, destination IP address) without sniffing packets? The destination info is easy, as you can port scan the destination server and glean that info. But the source info is much more difficult, as (at the very least) the client port will be some value between 1 and 65535.

There are cases where the client port is easy to guess. Communications between DNS servers may use the same port at each end (53/tcp). Is this the thing that CERT is so worried about? Maybe. A determined attacker could disrupt updates to root servers, with the effect that new DNS info could not be received. Remember what happened when Microsoft's DNS servers could not be reached in January? All of their domains, and related Web and email servers, effectively disappeared from the Internet.

Well, sorry, but this still seems pretty far-fetched to me. I also wondered about BGP4, used to exchanged routes between core routers. BGP4 requires that peer routers keep a live TCP connection at all times, so breaking a connection (and keeping it down for some time) would force the peers to announce new routes. The problem with ISN guessing affecting routers using BGP4 is that they use RFC2385, and sign their TCP headers, preventing spoofing. To be honest, I found this interesting, as someone decided that spoofing was enough of a threat to BGP4 that RFC2385 was considered that solution (*http://www.faqs.org/rfcs/rfc2385.html*, published in 1998).

So, should you worry about the ability of attackers to guess ISNs? If you are using the r commands, and have no firewall to protect the servers running these commands, yes. If you are worried about people hijacking or resetting TCP connections, no. The true solution is to use encrypted connections, such as SSH, SSL, or IPSec. While improving the methods used for ISN generation is important, encryption is the better countermeasure (as the CERT advisory notes).

Also, congratulations to OpenBSD and the Linux developers for having already implemented strong ISN, a la RFC1948. Note that FreeBSD has included the OpenBSD solution, and that other OS vendors, like Sun, support more secure ISN generation as a tunable kernel parameter, but not by default.

MS Horror Story

I just spent the last couple of hours installing Windows 2000 Professional. I guess Microsoft thought that if you got the Pro version, it should be difficult to install (for pros only, get it?). The fifth reboot just completed, soon after I elected to "Finish configuration of this server later." Well, I was sure that was what I selected, but Win2K has a "mind" of its own, and went about setting up Active Directory and other services after I told it "later."

I have seen a preview of the SAGE 2000 Salary Survey, and it seems that lots of you have to manage Windows servers (second only to Solaris). Sorry to hear that. The other inter-

I have seen a preview of the SAGE 2000 Salary Survey, and it seems that lots of you have to manage Windows servers (second only to Solaris). Sorry to hear that.

esting tidbit was that people who only manage Windows servers get paid less (a LOT less) than people who also manage UNIX systems. I really wondered about that. Is this merely because people consider UNIX so much more difficult than Windows, or because basic Windows administration is based on carrying about a CD pack, and reinstalling software frequently? Sorry, didn't mean to criticize anyone, but I did quote a UNIX sysadmin in a past column who described Windows sysadmin as exactly that. At least it would explain the salary gap.

Ah, Win2K is finally rebooted. After installing the (!#$!!) software that I need that will only run under Windows (yes, I know about and have installed StarOffice, and yes, I still use VMWare on my notebook, thanks), I have to reboot again. I notice that Win2K has a nifty new feature where your menu choice fades out after you have selected it. I call this the "LCD screen simulation effect," so that you can get some of the feeling of having an LCD screen without spending the money.

Before I conclude this column, I would like to thank the three Libertarians who were so annoyed by my February column that they wrote me letters many times longer than the actual column itself. I do read my email and will respond to any letters you care to send me, and am still wondering exactly why my cynicism only annoyed Libertarians.