Communications of the Association for Information Systems

Volume 11 Article 12

February 2003

Slammer: The First Blitz Worm

Raymond R. Panko
University of Hawaii, ray@panko.com

Recommended Citation
Panko, Raymond R. (2003) "Slammer: The First Blitz Worm," *Communications of the Association for Information Systems*: Vol. 11 , Article 12.
DOI: 10.17705/1CAIS.01112
Available at: https://aisel.aisnet.org/cais/vol11/iss1/12

Communications of the Association for Information Systems

SLAMMER: THE FIRST BLITZ WORM

Raymond R. Panko
University of Hawai`i
ray@panko.com

ABSTRACT

On January 25, 2003, the Slammer worm (also known as Sapphire) exploded on the Internet. Within ten minutes, it had taken over 90% of all unpatched computers running SQL Server or MSDE on the Internet. This article looks at several aspects of the Slammer infestation, including how it spread, the damage it caused, the crisis in vulnerability patching that it underscored, and the implications of the fact that Slammer probably was the first of a new class of worms predicted by Staniford, Paxson, and Weaver [2002]. These worms, which we will call blitz worms, can spread faster than human intervention can prevent, and radically new approaches will be needed to stop them.

KEYWORDS: Blitz Worm, Flash Worm, MSDE, Patch, Port 1434, Sapphire, Slammer, SQL Server, UDP, Virus, Vulnerability, Warhol Worm, Worm

I. INTRODUCTION

On Saturday, January 25, 2003, at 05:30 UCT, a worm began to spread through the Internet. This worm was called by several names, including Sapphire and the term we will use in this paper, Slammer.

The worm spread with astonishing speed. Slammer reached its peak packet attack traffic in an amazing three minutes [Moore, Paxson, *et al.,* 2003]. Within ten minutes, it infested about 90% of all vulnerable hosts on the Internet [Moore, Paxson, *et al.,* 2003]. Although Slammer was brought under control within hours, it achieved its aim of infesting nearly all vulnerable servers before the world even realized what was happening. The previous year, Staniford, Paxson, and Weaver [2002] suggested that new worms could spread almost instantaneously across the Internet. Although Slammer was slower than the suggested maximum spreading speed of such worms, it succeeded in infesting a large majority of all vulnerable systems before humans could react. It is time to begin asking how to address these new threats, which we will call blitz worms in this paper.

Slammer raises another important issue for the information systems community: patching. Slammer exploited a vulnerability that was known six months before Slammer hit; a patch from

Microsoft was available all that time. Microsoft labeled this patch as “critical.” Yet the rapid spread of Slammer indicates that large numbers of systems administrators failed to apply the patch—which was the only effective way to stop the worm. Was Slammer the result of widespread corporate negligence in applying the patch, or is the entire patching system fatally flawed?

In the same vein, initial reporting on the worm was confused and flawed, and the FBI’s National Infrastructure Protection Center (NIIPC) was silent during most of the crisis. January 25 was not a good day for Internet crisis management.

II. DAMAGE

Many articles described Slammer as a minor attack that did not do widespread damage. However, although Slammer did not delete files or do other deliberate damage, it took many critical database servers out of service, resulting in extensive damage worldwide.

Asia was hit particularly hard because Slammer spread at the beginning of the Asian business day. Most broadband ISP customers in South Korea lost their connections, and trading on the company’s highly automated stock exchange was greatly restricted, helping to send the market index down 2.7 percent. South Korean insurance officials said that they would pay out $860,000 for damages [Reuters, January 27, 2003]. South Korean President Kim Dae Jung called for government agencies to create contingency plans to thwart future attacks like Slammer [Reuters, January 27, 2003]. China cut most of its Internet connections to the outside world [Reuters, January 27, 2003], and Internet service also was hit hard in Japan and Taiwan [Chai, January 27, 2003]. All told, about 24% of the victim hosts were in Asia, the second worst-hit region after the United States, with 43% of the victim hosts [Moore, Paxson, *et al*., 2003].

Damage soon became worldwide, although it could have been much worse had the worm struck in the middle of a European or U.S. business day instead of on a weekend day. Most of the Bank of America’s 13,000 ATMs became unavailable [CNN.com, January 26, 2003]. Continental Airlines experienced some flight delays because they needed to revert to paper transaction handling [CNN.com, January 26, 2003]. A Boston medical center experienced slowdowns for about six hours, during which the center reverted to paper-based processing for patient orders and other processes [Roberts, January 27, 2003]. Police and fire dispatchers outside Seattle also reverted to slow paper processing at a 911 call center serving two suburban police departments and at least 14 fire departments [*InternetWeek*, January 28, 2003]. Other companies hit by the worm included Countrywide Financial Corporation, American Express, and several U.S. and Canadian banks [Bridlis, January 28, 2003]. In the UK, an unconfirmed report told of Hewlett-Packard sending some of its people home on Monday because of the worm, and service was badly affected at several e-commerce sites, including Borders.co.uk [Broersma, January 27, 2003]. Microsoft itself was hit hard by the worm, as evidence from leaked corporate e-mail from Microsoft’s internal security team [Leyden, January 28, 2003].

The attack hurt Internet service broadly. According to Keynote, which monitors download speeds for commercial webpages in the United States, download time at the average site in its index of 40 major U.S. home pages rose by more than 50 percent shortly after the attack [Broersma, January 27, 2003]. Matrix Systems, Inc. measured a 20% packet loss rate on the Internet, compared to 10% during the Code Red Attack [Sullivan, 2003]. Normally, packet losses are only about one percent. One company measured the packet loss rate for its Internet connection at 95% [Dyson, January 25, 2003].

Five of the 13 DNS root servers were taken down [Wagner, January 25, 2003]. In addition, numerous Internet routers encountered problems handling the increased traffic load, and quite a few reportedly tripped off because of a common software bug triggered by high traffic volume [Clarke, January 31, 2003]. Monitoring ISP newsgroups, Herbert [January 25, 2003] reported that Tier 1 (the largest) backbone carriers generally had instabilities, and one major Tier 1 service dropped its peering connections with other backbone carriers for a while. In addition, Cisco warned that a few Cisco applications running on some of its routers were vulnerable to Slammer [Cisco, January 26, 2003]; so some Cisco routers might have been taken over.

At least one company's switches failed over when traffic reached a little over 110 Mbps [Morton, January 25, 2003], and another company with only two infested servers found its Cisco PIX firewall's memory used up because of the large numbers of connections the firewall tried to establish [Kyle, January 25, 2003]. Hayne [February 8, 2003], who tested the Litchfield exploit (Section IV) in 2002, found that 75% of the capacity a 100 Mbps switch was used up in 5 seconds.

III. HOW SLAMMER SPREAD

A WORM

Although some news sources called Slammer a virus, it actually was a worm. Viruses infect (attach themselves to) programs or documents (as macros). Consequently, viruses can only spread if a human runs the program or opens the document, most commonly by opening an e-mail attachment.

Worms, however, spread on their own. When they infest (install themselves on) a machine, they actively seek to send themselves to other machines to infest those machines. Freed from the need for human action, worms can spread more quickly than viruses.

THE RATE OF SPREAD

When Code Red burst upon the Internet in 2001, it doubled its infestation rate every 37 minutes, eventually infesting over 350,000 hosts [Moore, Shannon, and Brown, 2002]. Code Red exploited a known vulnerability in Microsoft IIS webserver software. It was able to spread very rapidly in part because this software is so widely used. It did about $2 billion in damage.

Slammer spread far more rapidly [Moore, Paxson, *et al*., 2003]. Its doubling time was 8 seconds, allowing it to achieve its maximum scanning rate of over 55 million scans per second within three minutes (Section I). Although hosts continued to be infested after this period of maximum scanning, the bandwidth limitations of the Internet and corporate networks themselves limited the infestation rate (Section V). However, this slowing did not prevent the almost universal spread of the worm, which took place within the first ten minutes (Section I).

THE MECHANICS OF SLAMMER

Slammer exploited a buffer overflow vulnerability in Microsoft SQL Server 2000 (pronounced "sequel"), a popular database management system program. More than a million copies of SQL Server have been sold.

To take advantage of this vulnerability, Slammer sent an attack packet to UDP Port 1434. Under normal operation, when a client computer wishes to work with an SQL Server computer, it may send a UDP datagram to Port 1434 to learn what options it has in dealing with the server. The server sends back a reply detailing client options. Normally, the UDP datagram the client sends contains a single byte of data (02 hexadecimal) to indicate the type of information the client wishes. Slammer's datagram data fields instead contained a long overflow attack string. When the victim server attempted to process the string, the resultant stack buffer overflow attack gave Slammer local SYSTEM user privileges. Slammer installed itself and immediately began to attempt to install itself on other computers.

To infest many hosts, Slammer needed to send a large number of packets. Once installed on a host, Slammer sent a stream of messages to randomly selected destination IP addresses. These destination addresses included multicast addresses, which some routers probably sent to multiple machines. Slammer spoofed the source IP address with a random IP address to make infested hosts more difficult to detect. To send these messages, Slammer ran in an infinite loop that not only generated an enormous amount of traffic but also created a denial-of-service attack on the victim host.

Compounding this basic process, most destination IP addresses were unreachable; so many routers may have sent back ICMP error messages to the spoofed source address, further increasing the traffic load [Murphy, January 25, 2003].

Slammer was especially damaging because it attacked database servers. Although some of the webservers attacked by Code Red were mission-critical e-commerce servers, most of the webservers hit were not critical to business operation. However, database servers are much more likely to be handling critical transaction processing chores.

In an important wrinkle, Microsoft SQL Server was not the only product hit. Many client PCs were running the Microsoft SQL Server Desktop Engine (MSDE 2000), which was also vulnerable. In addition, quite a few software packages included MSDE 2000 for their internal operation, so even if user organizations were looking for MSDE, they probably would not find its embedded version within software packages.

In summary, Slammer's main damage was creating a denial-of-service attack on both the victim hosts, whose capacity was taken over almost completely by the worm's infinite attack loop, and on transmission links, which wasted much of their capacity on the worm's traffic. In addition, the high traffic loads that Slammer generated caused many unpatched Cisco routers to malfunction due to the known netflow switching bug, increasing the damage to transmission capacity.

IV. IMPLICATIONS FOR PATCHING

A LONG VULNERABILITY

Code Red took advantage of a vulnerability that was only widely known for about a month. In contrast, the specific vulnerability that Slammer exploited was widely known since July 2002—six months before Slammer attacked. The vulnerability was one of several SQL vulnerabilities that Mark Litchfield of Next Generation Security Software, Ltd. (NGS) reported to Microsoft on July 17, 2002.

Johannes Ullrich, director of the Incidents.org site, noted that any SQL Server host that was vulnerable to Slammer was vulnerable to attack for at least six months, and quite a few of these servers may already have been hacked [Lemos, January 27, 2003]. Given the importance of information stored on database servers, this observation is sobering.

MICROSOFT PATCHES FOR HOST SERVERS

On July 24, 2002, just a week after receiving Litchfield's vulnerability report, Microsoft released a patch for the vulnerability as Microsoft Security Bulletin MS02-039. Microsoft labeled the patch as critical. The problem, then, was that tens of thousands of SQL Server 2000 database programs were not patched by the organizations owning them. Under these circumstances, many analysts blamed user organizations rather than Microsoft.

However, the patch was difficult to apply. Bruce Schneier [CNN.com, January 28, 2003] called MS02-039 and subsequent SQL Server patches "difficult, time-consuming, and error-prone." The patch required systems administrator to do many changes manually.

Schneier also noted that systems often fail after Microsoft patches are applied. For instance, Microsoft released a patch for a Windows NT vulnerability on December 11, 2002. Users found quickly that this patch could crash their computers and reported this to Microsoft. However, Microsoft did not pull the patch until February 2003 [Evers, February 4, 2003]. In Microsoft security bulletins, there is a standard disclaimer that the patch is offered as is, without warranty.

Perversely, historical problems made users reluctant to install Microsoft patches, especially to systems whose continued operation is crucial to the firm. Frank Beier, president of Web design firm Dynamic Webs, said, "seems like every time I install a system patch, something else goes wrong with my system." He went on to say, "In most cases, I'm better off just playing Russian roulette with the hackers until our servers are broken into." The same problem applies to the installation of service packs. These service packs make many changes at once, so the potential for adverse side effects is high. Unless the firm had previously installed Service Pack 2, however, it could not install the MS-039 patch.

Later, in October 2002, Microsoft offered a cumulative security path (MS02-061) that included the SQL Server patch. Unfortunately, like the MS02-039 patch, this patch was difficult to apply,

requiring a good deal of manual work [Deegan, January 30, 2003]. Worse yet, unless a certain option was taken during the installation, this patch would undo the protection offered by MS02-039 [CNN.com, February 1, 2003].

On January 17, 2003, a bit more than a week before Slammer hit, Microsoft released SQL Server Service Pack 3, which also included the SQL Server patch. This upgrade was much easier to install, but it is likely that many firms were still testing the service pack before installing it because Microsoft service packs make many changes, some of which may cause serious operational problems.

After the Slammer attack, Microsoft upgraded MS02-061 to include automated patch installation. Later, on February 6, Microsoft released three scanning tools to help companies identify vulnerable computers. Each tool could only be used on certain products.

Although the Microsoft patch development effort left a good deal to be desired, many systems administrators who took the time and effort to test and install the patch offered little sympathy for systems administrators who failed to patch their servers. They pointed out that the patch was labeled critical, that SQL Server is widely deployed, and that database servers can hold extremely important information. If Slammer possessed a malicious payload, the damage would have been enormous. As one systems administrator put it, “I totally disagree with your lazy readers who are refusing to take responsibility for their inability to keep their security and patches up to speed. This patch was released last summer, and within 72 hours of release all our serves were updated” [Wagner, February 4, 2003].

MICROSOFT PATCHES FOR CLIENTS

As noted earlier, Microsoft SQL Server was not the only product hit. Many client PCs were running the Microsoft SQL Server Desktop Engine (MSDE 2000), and quite a few software packages included MSDE 2000 for their internal operation. (SQLsecurity.com provides a long list of affected products.) So even if users were looking for this program, they probably would not find its embedded versions within software packages.

Microsoft reported to its customers that “Typical home users’ computers are not affected,” but many ordinary user’ computers are not “typical” by Microsoft’s definition and do have MSDE 2000 or a product using it [Deegan, January 30, 2003]. MSDE is included in several Microsoft products. Sometimes, it is disabled by default but can easily be turned on. In other cases, it is turned on by default. The U.S. Department of Veterans Affairs also warned that some photographic, pharmaceutical, and medical equipment contained code affected by the vulnerability [CNN.com, January 30, 2003]. Particularly disturbing was that at least two security programs install MSDE by default [Laudat, January 29, 2003]. In both cases, these security products are redistributed in other products.

This spread of vulnerability beyond servers made Slammer very difficult to stop. Although Microsoft also supplied patches for MSDE 2000, patching a firm’s many client PCs is extremely difficult. Russ Cooper at TruSecure Corp found that half of the 4,000 computers on his corporate network turned out to be vulnerable [CNN.com, January 30, 2003]. The Beth Israel Deaconess Medical Center patched all of its servers in July 2002, yet infested clients with MSDE installed still caused heavy congestion [Roberts, January 27, 2003].

Although Microsoft outlined how to patch the vulnerability, these instructions were incomplete [Deegan, January 30, 2003]. They only covered Windows NT and 2000 clients—not Windows XP, Windows 98, or Windows ME clients. Deegan [January 30, 2003] noted that MSDE 2000 is a full database management system that requires a skilled administrator. Yet Microsoft provided only rudimentary administrative tools for MSDE. MSDE even violated proper security by not requiring a password.

EVALUATION COPIES

One source of problems, especially in universities, was vulnerabilities in evaluation copies of SQL Server. Many people running evaluation copies were hit because Microsoft remedies sometimes do not apply to evaluation copies.

IS PATCHING A BROKEN PROCESS?

Did Microsoft provide patches for SQL Server 2000 and MSDE 2000? The simple answer is that Microsoft did. However, the difficulty of applying these patches created major barriers to adoption. The MSDE patching process, in turn, could only be described as chaotic, including a lack of vulnerability assessment programs to identify vulnerable clients. That Microsoft itself was hit fairly hard [Leyden, January 28, 2003] indicates that Microsoft's patching process was broken, at least for this vulnerability.

The problem is not simply a Microsoft problem. Microsoft accounts for only a small fraction of the vulnerabilities found each year. In 2002, CERT (www.cert.org) reported over 2,000 vulnerabilities. The number of vulnerabilities doubled each year in recent years. Many of the resultant patches must be applied to multiple systems in a typical firm. This load is enormous.

In the end, however, although the patching process was less than desirable, the fact that large numbers of systems administrators failed to apply a security patch labeled as critical on important database servers indicates that the breakage is not entirely on Microsoft's end. Tens of thousands and perhaps millions of systems administrators left their corporations open to massive damage. Professionally, that is entirely unacceptable.

MONITORING

Bruce Schneier [February 15, 2003], who runs a company that provides monitoring service, argues that patch is inherently a problematic solution. It is simply impossible to keep up with the 20 or so patches that come out each week, many of which must be applied to multiple computers in a firm. Although patching should be done, it is unwise to create a security stance that is based on patching being effective. Such an approach is inherently fragile. If a company monitors its network actively, it will be able to detect attackers no matter what vulnerability they exploit. Patching plus strong monitoring would form a complete protection suite.

VULNERABILITY ASSESSMENT TOOLS

Another issue is the need for vulnerability assessment tools to examine a firm's hosts and applications in order to find vulnerabilities. Although a few high-end products do broadly ranging vulnerability assessment, these products are not economical for most firms. In addition, it would be ideal for these tools to be able to manage patch deployment fairly automatically (after testing). Although Microsoft and other companies provide such tools, they usually are fairly limited. For example, the Microsoft Baseline Security Analyzer only works on Windows NT, 2000, and XP, does not check all products for security patches, and looks at only some security configuration weaknesses [Panko, 2004].

V. A BLITZ WORM

In a seminal paper, Staniford, Paxson, and Weaver [2002] suggested that future worms might spread extremely rapidly through the Internet—far too rapidly for human reaction. Their proposed methods for the spread of such attacks, which we will call "blitz worms," were hotly debated. However, Slammer demonstrated that they were basically right, although even Slammer did not spread as rapidly as they said might be possible.

Staniford, Paxson, and Weaver [2002] began their analysis with an examination of the spread of the Code Red I worm on July 19, 2001. They found that the growth of Code Red I could be described adequately with a fairly simple model that they called random constant spread (RCS). Quite simply, this model assumed that the worm started with a single instance of the infestation and that infested hosts sent take-over packets to random addresses at a constant rate. The RCS model (which is a simple logistics model) involves only a single parameter—the rate at which a host can compromise others. RCS will create slow initial exponential growth, followed by rapid exponential growth and then quick saturation. In the case of Code Red I, the number of scans was comparatively small for the first nine hours after it began. Then the worm began to grow explosively, reaching saturation at 500,000 scans per hour in about 14 hours. Although Code Red turned itself off at midnight on the day it started, it had already largely completed its spread.

The Staniford, Paxson, and Weaver [2002] analysis of the RCS model showed two problems with RCS propagation. First, as with all exponential processes, growth is very slow initially. Second, even before a RCS infestation begins to reach saturation, most worms waste most of their efforts attacking already-infested computers. This led Staniford, Paxson, and Weaver [2002] to suggest that future worms would add two innovations. First, they would begin with a hit list, that is, a pre-built list of vulnerable hosts. They would then begin their infestation with a large number of computers, overcoming the slow initial spread of simple RCS. They would also use permutation scanning instead of simple random scanning to reduce the probability that worms would attack already-infested hosts. Staniford, Paxson, and Weaver [2002] estimated that such a worm, which they called a Warhol worm, could infest most vulnerable hosts in less than 15 minutes.

Staniford, Paxson, and Weaver [2002] posited an even more virulent worm, which they called a flash worm. Like the Warhol worm, the flash worm would begin with a hit list. However, the flash worm hit list would be extremely large. The list would have the IP addresses of a large percentage of all susceptible hosts on the Internet. This approach, plus permutation scanning or something like it, would allow a flash worm to hit almost all vulnerable hosts within about 30 seconds.

What was Slammer? From the speed of its spread, infesting about 90% of all vulnerable hosts within ten minutes, it sounds like a Warhol worm. However, Slammer really was a simple RCS worm [Moore, Paxson, *et al*. 2003]. There is no evidence that it used an initial hit list, and its code shows that it did not use permutation scanning or any other method to decrease the problem of attacking already-infested hosts as the worm neared saturation.

How could Slammer spread so fast, then? The answer is that it used UDP rather than TCP, as Code Red I did. Although UDP did not solve all problems, it did allow much faster spread.

Code Red's infestation rate peaked because of latency (delay) [Moore, Paxson, *et al*., 2003]. It simply took too long to attack another computer. Using TCP at the transport layer required the attack computer to open a connection to the victim computer before sending the Code Red worm to the computer. Making a connection required the attacker to send a SYN segment, the victim to send back a SYN/ACK segment, and the attacker to respond with an ACK segment. Only then could the worm send its take-over message. This three-way open took considerable time—especially after the attack was well under way and latency was beginning to grow because of traffic congestion. This TCP-driven latency limited the speed at which Code Red could spread, despite the fact that Code Red opened 99 threads so that it could process many openings in parallel.

In contrast, Slammer used UDP at the transport layer. UDP is connectionless, so the attacker can simply send a UDP attack datagram to the victim without waiting for a connection to open. This tactic gave hosts infested with Slammer a vastly faster scanning rate than hosts infested with Code Red I.

UDP datagrams are sent independently of one another. Consequently, the entire Slammer program had to fit into a single UDP datagram, which allows a maximum data field size of about 400 bytes. Slammer was 376 bytes long.

Although 376 bytes is a short message, Code Red's TCP-based attack openings usually only sent brief packets with TCP headers and no TCP data fields [Murphy, January 25, 2003]. Although Code Red I sent a 4 KB attack message if it found a vulnerable computer, successful connections were rare. Hosts infested by Slammer not only sent more messages per second than Code Red; their messages were longer on average, so congestion was much worse.

Small size was no impediment to Slammer's simple loop execution. According to the Internet Storm Center [2003], a single infested server could generate in excess of 50 Mbps in attack traffic.

However, small attack packet size limited what Slammer could do. Slammer could not include a destructive payload nor could it hide itself. Although it could have downloaded damaging programs and stealth programs from other sites, this capability would reduce its ability to execute

its primary function. Removing Slammer simply required rebooting the host, although given the rate of infestation messages, most hosts would be reinfested within minutes of rebooting unless a patch was applied before reattaching the server to the network.

Moore, Paxson, *et al*. [2003] found that what finally limited the maximum scanning rate of Slammer was limited bandwidth on the networks it was attacking and on the Internet. Quite simply, beyond some point, newly infested hosts could not get their attack packets out because of traffic from so many other infested hosts.

The news media made a major point of saying that Slammer was easy to stop. Companies and some ISPs simply filtered packets containing UDP datagrams addressed to Port 1434. However, this procedure often also stopped legitimate traffic going to these ports. More importantly, Moore, Paxson, *et al*. [2003] noted that by the time humans could intervene, the damage was already done. Within the first ten minutes of the attack, 90% of all vulnerable hosts were taken over. So, by the time humans noticed the attack, it was already "game over." If Slammer had executed a malicious attack script, hundreds of thousands of database servers around the world would be damaged heavily by the time any human even noticed the attack.

Another point that Moore, Paxson, *et al*. [2003] made was that previous worms needed to attack large populations of vulnerable hosts. With the type of methods used by Slammer, even 20,000 vulnerable hosts spread across the Internet might be enough to constitute the critical mass needed for attacks like Slammer to succeed.

In the end, as fast and damaging as Slammer was, Slammer could have spread much more rapidly using an initial hit list and more intelligent scanning; and Slammer could have been vastly more damaging if slammer clones downloaded and executed malicious attack scripts.

VI. PUBLIC POLICY ISSUES

In addition to raising concerns about patching and the realization of flash attack theory, Slammer raised two public policy issues: how to do vulnerability reporting and whether there is a need for government action to respond to similar threats in the future.

VULNERABILITY REPORTING

Computer software vendors engage in a love-hate (actually, more like tolerance-hate) relationship with vulnerability reporters who find vulnerabilities and report them to vendors. The SQL vulnerability that Slammer exploited was discovered by David Litchfield (Section IV) of Next Generation Security Software. Microsoft thanked Litchfield in their security bulletins.

However, Litchfield was criticized for posting a detailed exploit for the vulnerability. Litchfield [January 29, 2003] acknowledged that the attack code in Slammer was similar to the exploit he published in August 2002, at a Blackhat Security Briefing. However, he argued that systems administrators needed his published exploit so they could see how they were likely to be attacked. [Litchfield, January 29, 2003]. He added that "Anybody capable of writing such a worm would have found this information without my sample code." He also noted that Slammer's creator knew about buffer overflow exploits, and he estimated that he only saved the attacker about 20 minutes of code writing. Jason Coombs [January 29, 2003] supported disclosure, noting that when vendors release fixes, this information usually gives attackers sufficient knowledge to create their own exploits.

In an interesting analogy, Schneier [February 15, 2003] notes that most locks with master keys contain a vulnerability known to criminals and locksmiths for about a hundred years. For example, in 1994, Schneier notes, a thief stole $1.5 million in jewels using this vulnerability. Designs that defeat this vulnerability are known but they are not being produced in large numbers. Although informed consumers might demand such locks, lock makers did not inform customers about the danger. Ignorant of the danger and solutions, lock purchasers cannot make informed purchase decisions or demand safe locks. Yet when a security researcher published a paper detailing the vulnerability in January 2003, the lock companies "went ballistic."

The amount of detail in published vulnerability reports is one major issue in reporting. The other is timing. Vendors complain that vulnerability reporters sometimes report the vulnerability publicly at the same time they report it to the vendor or publish it very soon after reporting it—before the vendor issues a patch. On the other side of the picture, vulnerability reporters often complain that vendors sometimes take far too long to create a patch after a vulnerability is reported or even fail to acknowledge the vulnerability report at all. This attitude leaves potential victims unprotected for an unconscionable length of time. This timing issue was not a factor in the SQL Server vulnerability, but it is in many other vulnerabilities. Knowing that hackers and virus writers may also discover the vulnerability, vulnerability reporters consider it unprofessional to delay too long in publishing the vulnerability.

A NEED FOR GOVERNMENT ACTIONS?

A public policy concern is the need for government leadership. In fast attacks, corporate security officers need a source of information to which they can turn. In the past, CERT/CC (www.cert.org) was the "prime source" for vulnerability information for many end user organizations. However, CERT/CC's reputation was diminished by its creation of an alliance of organizations that pay to support CERT in return for receiving earlier warnings than CERT issues publicly. This preference raises the concern that CERT will not respond promptly at its public information site. One vulnerability reporter, David Litchfield (who discovered the SQL Server vulnerability (Section IV)) said that he will no longer submit vulnerability reports to CERT/CC because it released one of his vulnerability reports to its paid alliance before releasing it to the public. CERT/CC may no longer be the prime source for many firms.

An option is for corporate security officers to turn to the public website of the FBI's National Infrastructure Protection Center (NIPC). However, NIPC did not release its first advisory until a full 13 hours after the spread of Slammer [Roberts, January 28, 2003]. NIPC announced that it simply waited until it knew the full details and so could issue accurate information [Roberts, January 28, 2003]. However, Marcus Sachs of the White House Office of Cyberspace Security said that NIPC did not have the right people on staff on the weekend when the worm hit and found it difficult to bring in the staff members it needed [Roberts, January 28, 2003]. More generally, NIPC suffers from a long history of slow response, including criticism from the U.S. General Accounting Office in a 2001 audit [Thibodeua, May 22, 2001].

A policy and fiscal issue is that the Internet does not use a broad system of sensors and an always-staffed operations center to detect attacks. Although many call for creating such an Internet operations center (IOC), civil liberties concerns are severe, given the power such a center would possess. In addition, for the IOC to be effective in flash attacks, it would need to be able to command traffic shaping actions on core Internet routers worldwide within a few seconds. That is a dangerous power, and if attackers could impersonate the IOC, they could use this capability to create a devastating attack that would make Slammer seem like a pinprick.

MARKET FORCES

One possibility is that market forces will require vendors to create more secure software that would require much less patching. Unfortunately, market forces face the problem of monopoly or oligopolies in most areas [Banisar, 2002]. Microsoft dominates the market for client operating systems and office suites, and Oracle dominates the market for enterprise-class database processing. Under these conditions, switching costs can be enormous.

VII. CONCLUSION

Although Slammer was bad enough, future blitz worms will spread even faster and probably will be deliberately malicious. Malicious blitz worms will be able to do severe damage to a large fraction of all vulnerable hosts within seconds or minutes. They will be able to do cyberwar-scale damage before they are even noticed. Worst of all, it is easy to create such massively damaging blitz worms, and we should expect to see them in the near future.

The first line of defense against blitz worms must be patching. Given the ability of blitz worms to attack almost all vulnerable hosts in seconds or minutes, patching is a "do it or lose it" reality for

firms today. In this context, the fact that large number of corporations failed to apply a critical update to many of their important database servers is deeply frightening. In this threat environment, not patching because patching is difficult to do and must be tested carefully before deployment can only be called professional malfeasance.

Given the reality that patching is far from universal, the next line of defense may be rapid response by ISPs, perhaps under the coordination of what Staniford, Paxson, and Weaver [2002] call a Cyber-Center for Disease Control. However, even if civil liberty issues and other problems can be overcome, it will take some time before such an Internet operations center exists. It the meantime, patching is the only feasible line of defense today and can no longer be considered optional.

Editor's Note: This article was received on February 9, 2003 and was published on February 19, 2003. An earlier version of this article appeared in ISWorld. Because of the importance of its contents, the author was invited to publish the paper in the Communications of AIS so that it becomes a paper of permanent record.

REFERENCES

EDITOR'S NOTE: The following reference list contains the address of World Wide Web pages. Readers who can access the Web from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that

1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.
4. the author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Bridlis, Ted (January 28, 2003), "Worm's Disruptions Shake Preconceptions," Washingtonpost.com,http://www.washingtonpost.com/wp-dyn/articles/A53442003January28.html. Now archived.

Broersma, Matthew (January 27, 2003), "Slammer Worm Crashes into the UK," ZDNet, http://zdnet.com.com/2100-1105-982191.html.

Chai, Winston (January 27, 2003), "Worm's Home Ground May be Asia," ZDNet News, http://zdnet.com.com/2100-1105-982167.html.

Cisco Systems (January 26, 2002), "Cisco Security Advisory: Microsoft SQL Server Vulnerabilities in Cisco Products,". http://securityfocus.com.

CNN.com (January 26, 2003), "Computer Worm Grounds Flights, Blocks ATMs," http://www.cnn.com/2003/TECH/internet/01/25/internet.attack/index.html.

CNN.com (January 28, 2003), "Microsoft Also Gets Slammed by Worm," http://www.cnn.com/2003/TECH/biztech/01/28/microsoft.worm.ap/index.html.

CNN.com (January 30, 2003), "Techie Rethinks Disclosing Flaws," http://www.cnn.com/2003/TECH/internet/01/30/attack.disclose.ap/index.html.

Coombs, Jason (January 29, 2003), "Response to David Litchfield on Responsible Disclosure and Infosec Research," BugTraq, January 29, 2003, http://securityfocus.com.

Deegan, Peter (January 30, 2003), "Latest Worm Isn't Just for Servers," *Woody's Office Watch*, 8(3).

Dyson, Jay D. (January 25, 2003), "Re: MS SQL Worm is Destroying Internet Block Port 1434!" BugTraq, http://securityfocus.com.

Evers, Joris (February 4, 2003), "Microsoft Pulls Patch that Crashes NT 4.0," Computerworld.com, http://www.computerworld.com/securitytopics/security/holes/story/0,10801,78171,00.html.

Hayne, Stephen C (February 8, 2003), personal e-mail communication.

Herbert, George William (January 25, 2003), "Re: MS SQL Worm is Destroying Internet Block Port 1434!" BugTraq, http://securityfocus.com.

Internet Week (January 28, 2003), "SQL Server Worm Slammed Police, Fire, Microsoft Internal Servers," http://www.internetweek.com/story/showArticle.jhtml?articleID=6406475.

Kyle, Tom (January 25, 2003), "Re: MS SQL Worm is Destroying Internet Block Port 1434!" BugTraq, http://securityfocus.com.

Laudat, Stephan (January 29, 2002), "Re: MSDE Contained in …". http://securityfocus.com.

Leyden, John (January 28, 2003), "MS Struggles to Contain Slammer Worm," *The Register*, http://www.theregister.co.uk/content/56/29073.html.

Lemos, Robert (January 27, 2003), "SQL Worm Feeds on Apathy, MS Flaws," ZDNET News, http://zdnet.com.com/2100-1105-982135.html.

Microsoft (July 24, 2002), Microsoft Security Bulletin MS02-039, "Buffer Overruns in SQL Server 2000 Resolution Service Could Enable Code Execution (Q323875)," http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms02-039.asp.

Microsoft (October 16, 2002), Microsoft Security Bulletin MS02-061, "Elevation for Privilege in SQL Server Tasks (Q316333)," http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS02-061.asp. (Updated on January 28, 2002.)

Moore, David, Colleen Shannon; and Jeffrey Brown (2002), "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," *Proceedings of the Second ACM Internet Measurement Workshop*, http://www.caida.org/outreach/papers/2002/codered/codered.pdf. Cited in Moore, Paxson, *et al*., 2003.

Moore, David, Vern Paxson, Stephan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, (2003) "The Spread of the Sapphire/Slammer Worm," http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html.

Morton, Byron (January 25, 2003), "Re: MS SQL Worm is Destroying Internet Block Port 1434!" BugTraq,. http://securityfocus.com.

Murphy, Matthew, (January 25, 2003) "Re: MS SQL Worm is Destroying Internet Block Port 1434!" BugTraq,. http://securityfocus.com.

Panko, Raymond (2004, to be published in early 2003), *Corporate Computer and Network Security*, Upper Saddle River, NJ: Prentice-Hall. http://pankosecurity.com.

Staniford, Stuart, Vern Paxson, and Nicholas Weaver, (2002) "How to Own the Internet in Your Spare Time," *Proceedings of the 11th USENIX Security Symposium (Security '02)*, http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html.

Reuters (January 27, 2003), "Internet Worm Slithers On."

Reuters, (February 1, 2003). "Experts: Microsoft Security Gets an F," CNN.com, http://cnn.com/2003/TECH/biztech/02/01/microsoft.security.reut/index.html.

Roberts, Paul (January 27, 2003). "Businesses Dig Out From Slammer," IDG News Service, http://www.pcworld.com/news/article/0,aid,109007,00.asp.

Roberts, Paul (January 28, 2003), "Was the FBI's Response to Slammer Too Slow?" PCWorld.com, http://www.pcworld.com/news/article/0,aid,109036,00.asp.

Schneier, Bruce (February 15, 2002), "The Security Patch Treadmill," Crypto-Gram Newsletter, Counterpane Internet Security, http://www.counterpane.com/crypto-gram-0103.html.

Sullivan, Bob (January 25, 2002), "Virus-Like Attack Slows Web Traffic,".No longer online.

Thibodeua, Patrick (May 22, 2001), "GAO: NIPC Late on Cyberattack Alerts, Lacks Expertise," Computerworld. com, http://www.computerworld.com/securitytopics/security/virus/story/0,10801,60773,00.html.

Wagner, Mitch (January 25, 2003), "SQL Server Worm Slows Internet Traffic to a Crawl," Internetweek, http://www.internetwk.com/story/INW20030125S0001.

Wagner, Mitch (February 4, 2003), "Letters: Readers Take Peers to Task on Security Readiness", InternetWeek, http://www.internetweek.com/story/showArticle.jhtml?articleID=6511906.

ABOUT THE AUTHOR

Ray Panko is a professor of IT Management at the University of Hawai`i. Before coming to the University, he received his doctorate at Stanford University and was a project leader at Stanford Research Institute (now SRI International). He is the author of textbooks on networking (http://Panko.info) and IT security (http://pankosecurity.com), both published by Prentice-Hall. His research focuses on unreliability because of human error, legal and policy aspects of security, vulnerabilities in existing technologies, hardening the Internet, and understanding reasons for existing corporate security practices.

Communications of the Association for Information Systems

ISSN: 1529-3181