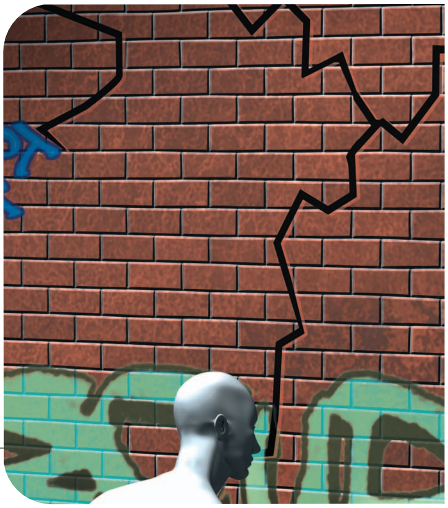
The Blaster Worm: Then and Now

The Blaster worm of 2003 infected at least 100,000 Microsoft Windows systems and cost millions in damage. In spite of cleanup efforts, an antiworm, and a removal tool from Microsoft, the worm persists. Observing the worm's activity can provide insight into the evolution of Internet worms.



Michael Bailey, Evan Cooke, Farnam Jahanian, and David Watson
University of Michigan

Jose Nazario
Arbor Networks

On Wednesday, 16 July 2003, Microsoft Security Bulletin MS03-026 (www.microsoft.com/security/incident/blast.mspx) announced a buffer overrun in the Windows Remote Procedure Call (RPC) interface that could let attackers execute arbitrary code. The flaw, which the Last Stage of Delirium (LSD) security group initially uncovered (http://lsd-pl.net/special.html), affected many Windows operating system versions, including NT 4.0, 2000, and XP.

When the vulnerability was disclosed, no known public exploit existed, and Microsoft made a patch available through their Web site. The CERT Coordination Center and other security organizations issued advisories over the next several days.[1] Almost immediately, discussions of the vulnerability began appearing on security lists. By 26 July, HD Moore had published a working exploit, dcom.c, on the Full Disclosure mailing list (http://lists.netsys.com/pipermail/full-disclosure/2003-July/007092.html).

Scattered reports of attackers reusing the exploit emerged during the next several weeks; then, on Monday, 11 August, the first Blaster worm variant struck. Also known as MSBlast or Lovsan, the worm copied code directly from the dcom.c exploit, added its own code, and launched a coordinated denial-of-service (DoS) attack to exhaust Windowsupdate.com's resources using a Transmission Control Protocol (TCP) port 80 SYN flood. It also used the backdoor mechanism from the example exploit to transfer the worm payload to newly infected systems.

Within its first week, the Blaster worm infected more than 100,000 Microsoft Windows systems. In spite of eradication efforts, the Blaster worm was alive and continued to infect new hosts more than a year later. By using a wide area network-monitoring technique that observes worm infection attempts, we collected observations of the Blaster worm during its onset in August 2003 and again in August 2004. This let us study worm evolution and provides an excellent illustration of a worm's four-phase life cycle, lending insight into its latency, growth, decay, and persistence.

How the Blaster worm attacks

The initial Blaster variant's decompiled source code reveals its unique behavior (http://robertgraham.com/journal/030815-blaster.c). The Blaster worm can be launched in one of two ways: as the result of a successful new infection or when a user reboots an already infected machine. Once launched, the worm immediately starts the setup for further propagation by choosing an address from the same local /16 (class B) address as the infected host. Next, it picks a random number to determine whether to use the local /16 address it just generated or a completely random one. The bias is 60 percent toward a random address. Next, the worm randomly chooses the offset to determine whether to infect Windows 2000 or XP, with an 80 percent bias toward XP.

On certain system dates, the initial variant, Blaster.A, then starts a thread to launch a DoS attack against Windowsupdate.com. It makes no further calls to the random number generator, but repeatedly seeds the random number with the number of milliseconds since boot time. This indicates that the worm author significantly lacks understanding of random number

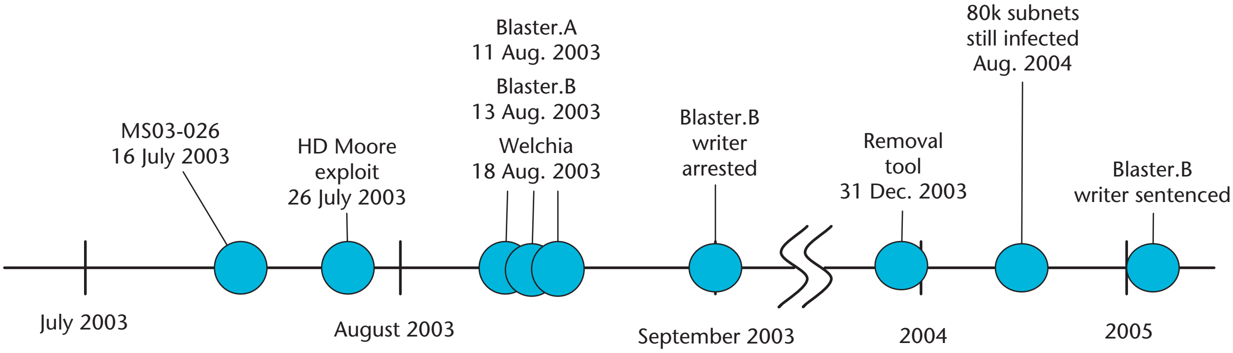


Figure 1. A Blaster worm time line. Although a rapid succession of activity occurs around the worm's initial release, its impact continues to be felt more than a year later.

generators. (Others have discussed the impact of these poorly seeded random generators.[2]) The propagation setup is complete when the worm uses the previously generated starting address and exploit offset to attempt to infect 20 sequential addresses using 20 threads on TCP port 135. It repeats the target infection attempt on the next 20 sequential addresses, indefinitely scanning IPv4 space in sequential order. If a connection attempt to TCP port 135 is successful, the worm sends an RPC *bind* command and an RPC *request* command containing the buffer overflow and exploit code. The exploit opens a backdoor on TCP port 4444, which waits for further commands. The infecting system then issues a command to the newly infected system to transfer the worm binary using Trivial File Transfer Protocol (TFTP) on UDP port 69 from the infecting system and execute it.

Numerous Blaster variants—as well as several new families of worms that exploit the same initial RPC vulnerability—have appeared since its release, many of them emerging within a few weeks of Blaster. Perhaps the two most notable are the Welchia (http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html) and SDBot (http://securityresponse.symantec.com/avcenter/venc/data/w32.randex.e.html) worms. Welchia, or Nachi as it's sometimes called, was an antiworm[3] that attempted to patch the vulnerability and ended up causing significant damage of its own. SDBot was notable in that it used the same RPC vulnerability to install the SDBot kit, which creates a backdoor on the system that enables remote control of the infected system through Internet Relay Chat (IRC).

The Blaster worm's impact wasn't limited to a short period in August 2003. A published survey of 19 research universities showed that each spent an average of US$299,579 during a five-week period to recover from the Blaster worm and its variants.[4] The cost of this cleanup effort has helped solidify a growing view of worms not as acts of Internet vandalism but as serious crimes. Although the original Blaster.A author was never caught, authors of several other variants have been apprehended.[5,6]

Worm measurement infrastructure

We measured of the Blaster worm by using a globally announced but unused /8 (class A) network, which represents roughly 1/256 of the Internet, or approximately 16 million addresses. This monitor is itself part of the Internet Motion Sensor (IMS; http://ims.eecs.umich.edu), a network of distributed blackhole sensors that monitor blocks of unused address space. Because no legitimate hosts exist in an unused address block, any observed traffic destined for such addresses must be the result of misconfiguration, backscatter from spoofed source addresses, or scanning from worms and other network probing. Prefiltering traffic in this way eliminates many false positives when identifying malicious traffic and helps us avoid the scaling issues of other monitoring approaches (for a discussion of such approaches, see the sidebar on p. 30). This technique goes by several names, including network telescopes,[7] blackholes,[8,9] and darknets (www.cymru.com/Darknet/index.html).

The Blaster life cycle in August 2003

One of the Blaster worm's more interesting elements is that it provides an excellent example of a worm's life cycle. Although not all worms follow this cycle in its entirety, it's still informative for understanding broad behaviors. As mentioned earlier, a four-phased worm life cycle consists of *latency*, *growth*, *decay*, and *persistence* (see Figure 2).

Latency describes the time period between discovering a vulnerability and observing the appearance of a worm in the wild. This period might include vulnerability publication, patch release, and theoretical or working

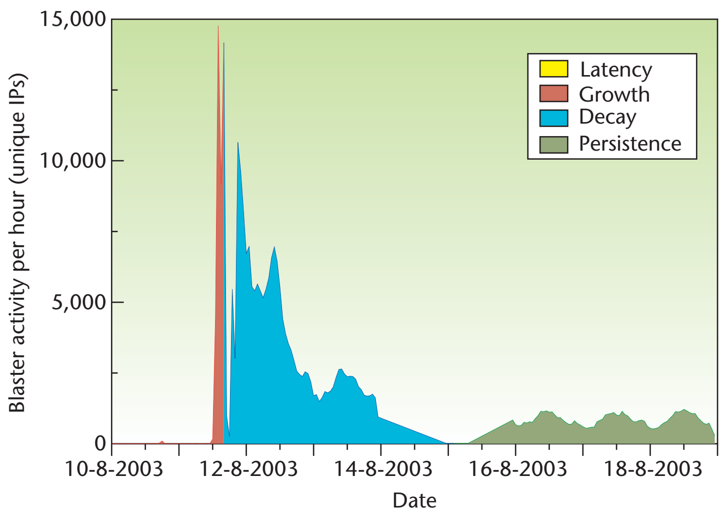


Figure 2. The Blaster worm life cycle. The four phases shown include the end of the latency phase, its growth phase, its decay phase, and the beginning of its persistence phase.

exploits. Our measurement infrastructure observed that after the LSD group released its initial advisory and Microsoft its advisory and patch, the number of unique hosts scanning TCP port 135 increased from between four and 10 unique sources per day to between 100 and 300. Additionally, we observed several higher activity spikes during this period, each correlating to the release of successive exploit tools from various underground individuals or groups.

In the growth phase, the worm strikes and begins to infect the vulnerable population. We often use epidemiological models from population growth and biology to describe this period, and their application to random scanning worms is well understood.[10–12] During the first few hours of its growth, the Blaster worm spread exponentially, with a doubling time of approximately nine minutes. However, as the worm's prevalence increased, this doubling time slowed, and our observations peaked at nearly 15,000 unique IP addresses scanning TCP port 135 in a single, one-hour period, with roughly 106,000 unique sources in a 24-hour period. Reverse Domain Name System (DNS) lookups for the active hosts during the peak hour of Blaster's activity showed a global distribution of hosts (see Table 1). Upon analyzing the second-level domain names, we see that the worm's spread affected several consumer broadband providers.

The growth phase is followed by a decay phase in which infected systems are patched or removed from the network, or organizations deploy policies to minimize the worm's impact. Within eight hours of the worm's initial outbreak, the number of unique hosts per hour scanning for TCP port 135 against 256 contiguous addresses began to diminish. Fitting this loss of worm activity to a simple exponential decay, we calculated a half-life of roughly 12 hours. This loss of activity continued for approximately four days.

Finally, most nondestructive worms enter a persistent phase in which a relatively small population of hosts remain infected. Following the decay phase, the observed Blaster activity reached a fairly consistent level. Figure 3 shows Blaster activity in late August 2003 compared to activity one year later. The initial growth in observations in August 2003 correlates with the Welchia worm's appearance on 18 August. The last two days represent the steady state seen for the next several months.

The Blaster worm a year later

The Blaster worm was released roughly two years ago, providing ample opportunity for individuals and organizations to clean up infected machines. We might thus expect the worm to decay quickly, with only a handful of hosts still infected in August 2004. In reality, a year later, the Blaster worm was not only still scanning the Internet, but the remaining infected population was larger than expected.

We performed observations for two months starting in August 2004 and discovered more than 200,000 unique IP addresses scanning our dark address monitor, with peaks of 4,100 unique addresses per day and 500 per hour. Although the effects of the Dynamic Host Configuration Protocol (DHCP) can lead to overcounting,[10] other effects, such as the Blaster worm's slow scanning rate and a host's short daily operational lifetime, might lead to undercounting when we consider only daily or hourly values. To evaluate the effects of DHCP overcounting, we analyzed the number of unique subnets. We did this by evaluating only the first 24 bits of an observed IP address, assuming that most DHCP-assigned addresses are from a small pool of IP addresses in the local subnet. This analysis showed roughly 90,000 unique subnets during the two-month period (August and September 2004), suggesting a significant number of unique infected hosts.

The activity in both the 2003 and 2004 observations displays a circadian pattern, with peak activity occurring near midday in the eastern US (see Figure 3). Closer inspection shows that, on average, Monday sees the most traffic and Saturday the least. Additionally, the peak activity typically occurs between 10:00 and 15:00 EST, which roughly correspond to working hours in the eastern US and suggests that the infected systems are power cycled every day as workers and home users turn their computers on and off.

The fact that many old worms persist is not a new observation,[13] but the huge number of unique hosts still infected with Blaster is very surprising. Interestingly, Microsoft released a Blaster removal tool on 6 January 2004 (www.microsoft.com/presspass/press/2004/feb04/02-24GIAISpr.asp) that was supposed to remove the worm executable and install the patch needed to prevent reinfection. Although millions of users downloaded this tool, Blaster observations for January 2004 changed very little, suggesting that it had little effect on the infected population at large. Simply put, these persistent infections are due to people who either don't know or don't care enough that their machines are infected.

One way to gain insight into the Blaster's persistence is to compare the infected population observed during the outbreak with the infected population one year later. Looking first at reverse DNS data, we find that the two populations have similar demographics (see Table 1). A comparison of the number of US-based infections during the outbreak with its persistent population shows them to both to be roughly 55 percent. Overall, the top-level domain distribution across countries changed very little between outbreak and persistence. The number of infected systems in different countries appears to have remained relatively stable. However, the actual addresses have changed dramatically: 99.5 percent of persistent addresses aren't found in the outbreak population and vice versa.

We also compared the identified Blaster populations from August 2003 and August 2004, and tried to account for DHCP effects. It's possible that several infected hosts simply moved within the same LAN due to DHCP or other administrative decisions, so again we compared only the first 24 bits of the host IP addresses. We found that 73 percent of the persistent addresses aren't found in the outbreak population and 85 percent of outbreak addresses aren't found in the persistent population. Thus, a very significant movement exists between subnets. As a final test, we performed the same analysis comparing only the first 16 bits of the addresses and masking out the last 16. The results showed that 37 percent of persistent addresses aren't found in the outbreak population and 21 percent of outbreak addresses aren't found in the persistent population. Hence, a substantial churn still exists in the infected networks between the outbreak and infected populations.

Our observations indicate that the Blaster worm isn't going away anytime soon. Recent tools targeted at eradicating it appear to have had little effect on the global population. Additionally, when we analyze the persistent population, we see that the infection appears to have successfully transitioned to new hosts as the original systems are cleaned or shut off, suggesting that the Blaster worm, and other similar worms, will remain significant Internet threats for many years after their initial release. □

Table 1. Top-level domain (TLD) analysis of unique source IPs on 11 August 2003 and one year later.*

TLD	AUGUST 2003 (%)	AUGUST 2004 (%)
net	39.3	34.2
com	15.4	19.6
jp	3.8	7.7
fr	3.1	2.1
ca	1.8	1.3
de	1.7	5.3
br	1.5	2.0
it	1.2	3.5
au	1.2	0.0
edu	1.1	0.1

*Roughly 40 percent of the addresses had reverse Domain Name System entries.

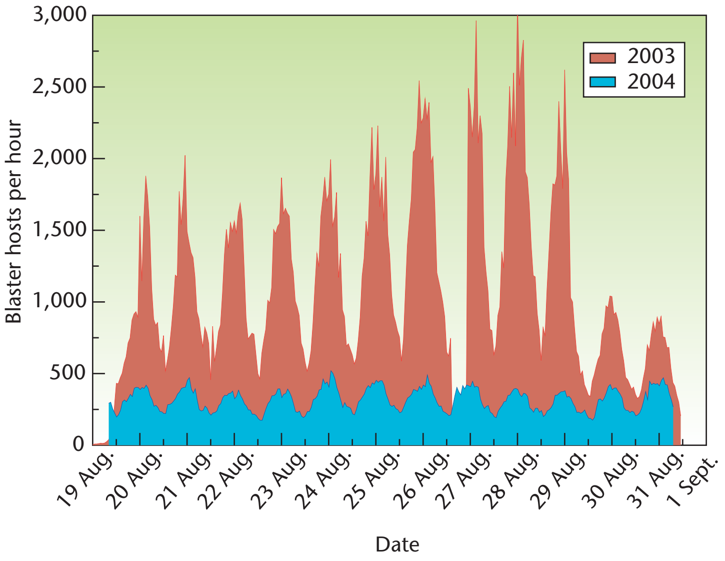


Figure 3. Unique Blaster hosts per hour. We measured this for late August 2003 and for the same period in 2004. Although the magnitude of the persistent population is less, the pattern that emerges in late August 2003 is still evident in August 2004.

Acknowledgments

This work was supported by the US Department of Homeland Security (DHS) under contract number NBCHC040146, the Advanced Research and Development Activity (ARDA) under contract number NBCHC030104, and by a corporate gift from Intel Corporation. We thank all the IMS participants for their help and suggestions. We also thank Dug Song, Robert Stone, and G. Robert Malan of Arbor Networks and Larry Blunk, Bert Rossi, and Manish Karir at Merit Network for their assistance and support. We thank Johannes Ullrich from the SANS Internet Storm Center for sharing DShield data for comparison. Finally, we thank the anonymous reviewers for their corrections and helpful comments.

Worm-monitoring techniques

In contrast to monitoring unused address space—the technique we used to monitoring the Blaster worm—other approaches to monitoring worms and other global Internet threats monitor production networks with live hosts. In monitoring used networks, systems can choose to watch traffic directly via fiber taps or span ports, or watch data abstractions, such as flow records, device alerts, or logs. Security devices are an important source of these abstractions; alerts and logs from host-based antivirus programs, intrusion-detection systems, and firewalls can all help effectively characterize worms. To achieve the scale required to monitor broadly scoped threats, systems that use this direct monitoring approach often aggregate data from tens of thousands of individual devices into a global view (see http://analyzer.securityfocus.com).

Our observations of Blaster worm activity differ significantly from Blaster observations performed using other methods. For example, the Microsoft *remove* tool saw a completely different view of Blaster activity. Microsoft reported that this tool helped clean 8 million infected computers in January 2004 (www.microsoft.com/presspass/press/2004/feb04/02-24GIAISpr.asp). The maximum number of unique hosts per day we measured was 106,000. To understand these differences, we compared the observations from our network monitor to data collected from DShield (www.dshield.org). The DShield approach to network monitoring aggregates information from tens of thousands of individual security devices throughout the network. During Blaster's onset, DShield saw a maximum number of 208,000 unique IP addresses scanning TCP port 135. Prior to the initial Blaster worm attack, DShield saw roughly 7,000 unique IP address scanning this port. Although these two methods saw roughly the same order of infections, these numbers differ significantly from the values Microsoft reported.

To explore where these differences might have come from, we examined the distribution of the source IP addresses from both our unused network monitor and the DShield data. By examining a BGP routing table from 11 August 2003, just before the worm's onset, we were able to calculate the number of network blocks in each region of the IPv4 space and determined which networks had infected hosts visible to each technique. Figure A shows the result of this analysis. The horizontal axis in the figure ranges from 0 to 255, showing each of the 256 possible first octets in IPv4 space. The vertical axis shows the proportion of Blaster observations made using each method for each advertised address block. Although the unused address monitoring and firewall logs show similar visibility in the networking space, neither technique saw infection attempts from nearly 95 percent of the routed networks. Whether the lack of observations from these blocks is the result of some policy deployed at these networks, an artifact of worm propagation, or the distribution of the vulnerable hosts is unknown. Understanding the differences between these two perspectives of the vulnerable population remains an interesting research problem.

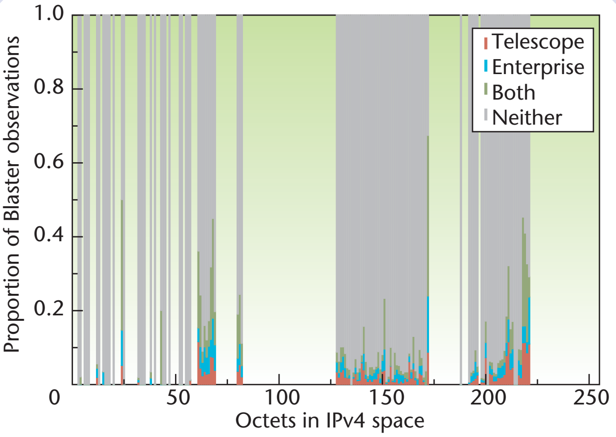


Figure A. The percentage of routed networks with Blaster activity recorded by two different network-monitoring techniques.

References

1. CERT Coordination Center, "CERT Advisory CA-2003-20 W32/Blaster worm," Aug. 2003; www.cert.org/advisories/CA-2003-20.html.
2. E. Cooke, Z.M. Mao, and F. Jahanian, *Worm Hotspots: Explaining Non-uniformity in Worm Targeting Behavior*, tech. report CSE-TR-503-04, Dept. Electrical Eng. and Computer Science, Univ. of Michigan, 2004.
3. F. Castaneda, E.C. Sezer, and J. Xuy, "Worm vs. Worm: Preliminary Study of an Active Counter-Attack Mechanism," *Proc. 2004 ACM Workshop Rapid Malcode* (WORM 04), ACM Press, 2004, pp. 83–93.
4. A.L. Foster, "Colleges Brace for the Next Worm," *Chronicle of Higher Education*, vol. 50, no. 28, 2004, p. A29.
5. K. Peterson, "Blaster Hacker Receives 18-Month Sentence," *Seattle Times*, Business and Technology section, 29 Jan. 2005.
6. P. Roberts, "Suspected Blaster-F Author Nabbed," *PC World*, 3 Sept. 2003; www.pcworld.com/news/article/0,aid,112308,00.asp.
7. D. Moore, G.M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *Proc. 10th Usenix Security Symp.*, Usenix Assoc., 2001, pp. 9–22.
8. M. Bailey et al., "The Internet Motion Sensor: A Distributed Global Scoped Internet Threat Monitoring System," *Proc. Network and Distributed System Security Symp.* (NDSS 05), Internet Soc., 2005.
9. D. Song, R. Malan, and R. Stone, "A Snapshot of Global Internet Worm Activity," *Proc. 1st Conf. Computer Security Incident Handling and Response*, 2002; www.4law.co.

il/Le288.htm.

10. C. Shannon, D. Moore, and J. Brown, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," *Proc. Internet Measurement Workshop* (IMW), ACM Press, 2002, pp. 273–284.
11. S. Staniford, V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time," *Proc. 11th Usenix Security Symp.*, Usenix Assoc., 2002, pp. 149–167.
12. C. Changchun et al., "Monitoring and Early Warning for Internet Worms," *Proc. 10th ACM Conf. Computer and Comm. Security*, ACM Press, 2003, pp. 190–199.
13. D. Song, R. Malan, and R. Stone, *A Snapshot of Global Internet Worm Activity*, tech. report, Arbor Networks, June 2002.

Michael Bailey *is a graduate student and program manager at the University of Michigan, and a former director of engineering at Arbor Networks. His research interests include the security and availability of complex distributed systems. Bailey has a BS in computer science from the University of Illinois at Urbana-Champaign and an MS in computer science from DePaul University. Contact him at mibailey@umich.edu.*

Evan Cooke *is a PhD candidate at the University of Michigan and a lead researcher on the Internet Motion Sensor (IMS) project. His research interests include network security, large-scale Internet measurement, and distributed systems. Cooke has BS degrees in electrical engineering, computer science, and psychology from the University of Wisconsin and an MS in computer science from the University of Michigan. Contact him at emcooke@umich.edu.*

Farnam Jahanian *is a professor of electrical engineering and computer science at the University of Michigan and cofounder of Arbor Networks. His research interests include distributed computing, network security, and network architectures. Jahanian has an MS and a PhD in computer science from the University of Texas at Austin. Contact him at farnam@umich.edu.*

David Watson *is a postdoctoral research fellow at the University of Michigan. His research interests include network routing protocols and network infrastructure security Watson has a BS in computer science from Carnegie Mellon University, and an MSE and PhD in computer science from the University of Michigan. Contact him at dwatson@umich.edu.*

Jose Nazario *is a software and security engineer at Arbor Networks. His research interests include worm detection techniques, distributed denial-of-service activity, and large-scale Internet measurements. Nazario has a BA in biology from Luther College and a PhD in biochemistry from Case Western Reserve University. He is the author of* Defense and Detection Strategies Against Internet Worms *(Artech House, 2003). Contact him at jose@arbor.net.*