fragmentation and recycling of necrotic and apoptotic tissues, whereas Ly-6C^{low} monocytes arrive at the scene later to promote revascularization and collagen deposition. Recruitment of Ly-6C^{high} monocytes to damaged myocardium is dramatically diminished in CCR2-deficient mice. Swirski *et al*. used the mouse myocardial infarction model to further characterize Ly-6C^{high} monocyte recruitment and identified the subcapsular red pulp of the spleen as a major source of recruited monocytes. Interestingly, angiotensin II, a circulating peptide that regulates vascular tone and blood pressure, promotes CCR2-independent emigration of splenic Ly-6C^{high} monocytes into the circulation.

Corticosteroid administration and vigorous physical exertion both result in abrupt increases in the number of circulating white blood cells, including monocytes. It has been assumed in these circumstances that white blood cells are released from endothelial surfaces. The finding by Swirski *et al*. that an increase in the circulating concentration of angiotensin II after myocardial infarction induces the dimerization of the angiotensin receptor on Ly-6C^{high} monocytes reveals a novel mechanism to boost circulating white blood cells in times of stress.

The findings by Swirski *et al*. raise questions about whether other types of stress or injury draw upon the spleen's reserve of monocytes as well. In the meantime, although the study does not make the spleen any less dispensable for mammalian survival, it does make this easily dismissed immune system organ seem a bit more purposeful and deserving of recognition.

References

1. F. K. Swirski *et al*., *Science* **325**, 612 (2009).
2. F. R. Sabin, C. A. Doan, *J. Exp. Med.* **46**, 627 (1927).
3. C. Auffray, M. H. Sieweke, F. Geissmann, *Annu. Rev. Immunol.* **27**, 669 (2009).
4. N. V. Serbina, T. Jia, T. M. Hohl, E. G. Pamer, *Annu. Rev. Immunol.* **26**, 421 (2008).
5. H. W. Ziegler-Heitbrock, B. Passlick, D. Flieger, *Hybridoma* **7**, 521 (1988).
6. F. K. Swirski *et al*., *J. Clin. Invest.* **117**, 195 (2007).
7. L. Boring, J. Gosling, M. Cleary, I. F. Charo, *Nature* **394**, 894 (1998).
8. L. Gu *et al*., *Mol. Cell* **2**, 275 (1998).
9. F. Tacke *et al*., *J. Clin. Invest.* **117**, 185 (2007).
10. N. V. Serbina, E. G. Pamer, *Nat. Immunol.* **7**, 311 (2006).
11. C. L. Tsou *et al*., *J. Clin. Invest.* **117**, 902 (2007).
12. C. Varol *et al*., *J. Exp. Med.* **204**, 171 (2007).
13. M. Nahrendorf *et al*., *J. Exp. Med.* **204**, 3037 (2007).

10.1126/science.1178329

COMPUTER SCIENCE

Is Your Computer Secure?

Security must be built in to software from the outset rather than added on later.

Frederick R. Chang

Cybersecurity is a pressing global issue today and increasingly affects all of us. According to a recent estimate, each U.S. adult had a 66% chance of experiencing at least one data exposure in 2008 and a 30% chance of experiencing multiple such exposures (*1*). A key motivator for cyber attackers is financial gain; a study of the underground economy observed advertisements for over $276 million in total "goods" (such as stolen credit card information) during a recent 1-year period (*2*). The annual cost to companies due to intellectual property theft and repair after data breaches has been estimated at over $1 trillion globally (*3*). Yet, in a study of consumer's personal computers, only 37% had up-to-date anti-malicious software (malware) protection; of those, 23% had active malware infections (*4*). What are the key cybersecurity dangers today, and how can they be addressed?

Computers can be infected merely by surfing the Web. By attacking a single Web site, attackers can potentially infect the computers of all visitors to that site. Using a technique known as SQL (Structured Query Language) injection, an attacker can insert malicious code into the database associated with the Web site. If the victim's browser is vulnerable, that malicious content is transmitted to the victim's computer. Using another technique, cross-site scripting, the software toolkit known as "Mpack" recently caused considerable damage (*5*). Users visiting legitimate Web sites were invisibly redirected to a server that downloaded malicious software onto the user's machine. Various types of malware can be downloaded to the victim's machine in this way, including keyloggers (which steal account and password information), rootkits (which hide the presence and activity of malicious software), and software enlisting the computer in a botnet.

Botnets are responsible for attacks including spam, phishing, distributed denial of service, data harvesting, click fraud, and password cracking. A bot is a computer that has been infected such that it can be remotely controlled; a botnet is a large network of bots. Up to 25% of the world's network-connected computers may be part of a botnet (*6*). In 2008, the botnet Srizbi sent out an estimated 60 billion spam messages per day—about 50% of the world's total (*7*). At the end of 2008, Srizbi's impact was much reduced when a suspect Web hosting company was cut off from the Internet. The botnet made a comeback, but in early 2009 a software patch was released that could remove the Srizbi software from client computers.

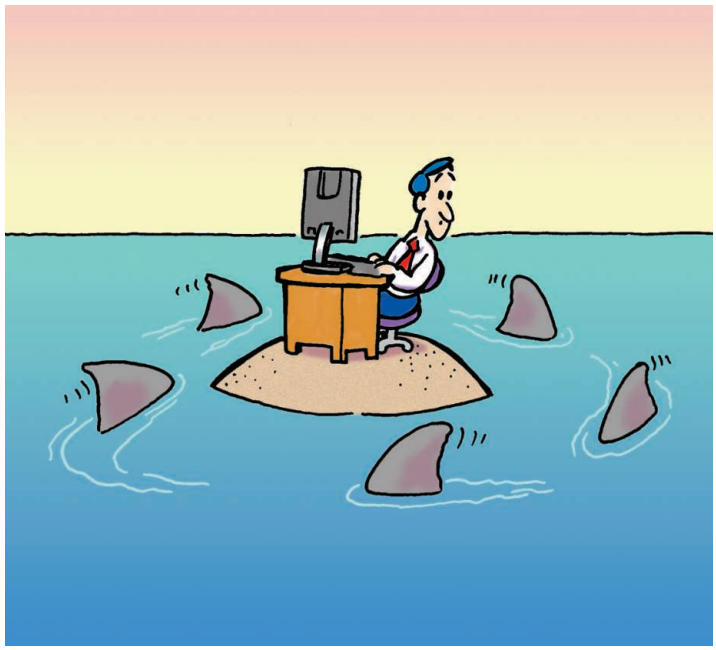
Another recent botnet, Storm, was estimated to have infected 1 million to 5 million computers; each infected computer sent out an average of 1200 spam messages an hour (*8*), generating healthy revenue for the botnet owners (*9*). Storm successfully evaded antivirus protection, had a decentralized control structure that made it difficult to shut down, and had a built-in self-defense mechanism (it launched denial of service attacks against researchers trying to access and study it). Storm also made sophisticated use of social engineering techniques: It was highly effective at inducing people to take action (such as to download and execute files), thereby infecting their computers with malware.

Social engineering here refers to manipulating a computer user to take an action with undesired consequences, such as downloading a file containing malware, clicking on a link that takes them to a fraudulent Web site, or divulging confidential information. Many users are easily manipulated in this way. In a study (*10*), university computer science students were sent an e-mail explaining that the system password database had been compromised and that they should reply with their password so that they could be validated in the database. No such database compromise had in fact occurred. The students were advised that they should never reveal their passwords to anybody, yet 41% of the students sent their passwords. Most were suspicious, changing their passwords in the 2 weeks after the study, but very few reported the incident (*10*).

In another study, a credit union hired a security firm to perform a social engineering "penetration test" on itself. When

Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712, USA. E-mail: chang@cs.utexas.edu

a group of the credit union's employees found "free" USB drives near their place of work, they brought them into the workplace and plugged them into their computers. The drives contained malicious software and infected the users' computers (*11*). Malware authors have also used phony security software or fraudulent electronic greeting cards. As attackers target the "human layer" with social engineering, it will be increasingly important to ensure that users are educated as to current cybersecurity risks and dangers.



A key problem is that too much software today is insecure. Reports of software vulnerabilities are increasing rapidly (*12*), and attackers can easily exploit these vulnerabilities to cause harm. Even when a specific software vulnerability is reported, it can take months for an appropriate patch to be released and even longer for that patch to be installed to repair the vulnerable system. Antivirus software, firewalls, and intrusion detection systems are important and useful technologies, but they are symptomatic of the fact that security is often an afterthought.

The traditional goals of software quality and reliability must be expanded to include the fact that software must operate dependably in the face of malicious attack. As a result, increasingly, security must be built in to software from the very beginning (*13*). This is not to say that all software needs to be made nearly "hacker proof," but rather that much more software needs to be made more "hacker resistant." It is critical to know how secure the system needs to be.

If security is to be built into the software, then the software must be free of known bugs that can be exploited to compromise security. There are lists of common programming errors that can help the programmer, as well as powerful "bug finding" tools that can automate the identification of serious bugs (*14*). Security must be a part of the design of the software, and rigorous mathematical analysis techniques should be used whenever possible to ensure that the designs are adequate. The goal of security leads to different questions for the software designers. Can the software withstand a malicious attack? Are there multiple layers of defense to make it more difficult for attackers to succeed? For example, access to a sensitive file might require proper permissions, a strong password and a decryption key. What happens if the software is compromised? Does the system degrade gracefully— that is, does it continue to operate, albeit at a possibly reduced level? Finally, security should be integrated into the entire software development life cycle (requirements, design, coding, testing, deployment and operations) (*13*).

Building security in is not a new problem. Fortunately, important technical advances (*15*) over the past 25 years have improved the ability of developers to build more fundamentally secure systems. Better decision methods such as BDDs (binary decision diagrams) and SAT (propositional satisfiability) along with advances in model checking and theorem proving have greatly improved the ability to reason about complex system abstractions. Combined with greater computational power, these sorts of advances, among others, have led to tangible improvements in the ability to produce more fundamentally sound hardware and software systems. Numerous modern tools have been developed based on these advances and are increasingly being implemented, with positive results. Indeed, the UK-based software company Praxis High Integrity Systems uses rigorous mathematical methods to reduce software defects and improve the quality of their software. This company offers a software warranty to their customers—a rather rare occurrence in the software industry (*16*).

Technology advances alone will not solve all the problems. Indeed, critical insights into the limitations of technical solutions are being gained at the intersection of information security and the social sciences. In economics (*17*), a key issue relates to incentives: How people are motivated to behave regarding system security may be more important than how the system itself behaves. In psychology (*18*), a key question is why social engineering techniques continue to be so successful. As technical measures improve the security of systems, the end-user will increasingly become the weakest link.

As the attractive economics of cloud computing trigger yet another wave of valuable services over the Web, software security will become even more urgent. Perfectly secure software is impossible, but that does not mean that we should not try.

References and Notes

1. D. E. Geer, D. G. Conway, *IEEE Secur. Privacy* **6**, 87 (2008).
2. *Symantec Report on the Underground Economy* (Symantec Corporation, Cupertino, CA, November 2008).
3. www.mcafee.com/us/about/press/corporate/2009/20090129_063500_j.html
4. *Malware Infections in Protected Systems* (Panda Security, Glendale, CA, 2007).
5. BBC News, "Hackers target 'legitimate' sites," 20 June 2007; news.bbc.co.uk/2/hi/technology/6221306.stm
6. T. Weber, "Criminals 'may overwhelm the web,' " BBC News, 25 January 2007, news.bbc.co.uk/2/hi/business/6298641.stm.
7. J. Stewart, *Top Spam Botnets Exposed* (SecureWorks, April 2008); www.secureworks.com/research/threats/topbotnets
8. P. Porras *et al.*, *A Multi-Perspective Analysis of the Storm (Peacomm) Worm, CSL Technical Note* (SRI International, Menlo Park, CA, October 2007).
9. C. Kanich *et al.*, *Spamalytics: An Empirical Analysis of Spam Marketing Conversion* (CCS'08, 27 to 31 October 2008, Alexandria, VA), see www.sigsac.org/ccs/CCS2008/program.html.
10. T. Greening, *SIGSAC Review* **14**, 9 (1996).
11. DarkReading, "Social engineering, the USB way," 7 June 2006; www.darkreading.com/security/perimeter/showArticle.jhtml?articleID=208803634.
12. *IBM Internet Security Systems X-Force 2008 Trend & Risk Report* (IBM Global Technology Services, Somers, NY, January 2009).
13. G. McGraw, *IEEE Secur.w Privacy* **2**, 80 (2004).
14. B. Chess, J. West, *Secure Programming with Static Analysis* (Addison-Wesley, Boston, MA, 2007).
15. R. E. Bryant, *IEEE Trans. Comput.* **c-35**, 677 (1986).
16. C. Jones, P. O'Hearn, J. Woodcock, *IEEE Compu.* **39**, 93 (2006).
17. R. Anderson, T. Moore, *Science* **314**, 610 (2006).
18. B. Schneier, *The Psychology of Security*, January 2008; www.schneier.com/essay-155.html.

19. I thank J Moore, M. Dahlin, W. Hunt, P. Acosta, and two anonymous reviewers for comments.

10.1126/science.1168661

CREDIT: JOE SUTLIFF/WWW.CDAD.COM/JOE

Is Your Computer Secure?
Frederick R. Chang (July 30, 2009)
Science **325** (5940), 550-551. [doi: 10.1126/science.1168661]

Editor's Summary

Science (print ISSN 0036-8075; online ISSN 1095-9203) is published weekly, except the last week in December, by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005.