CERIAS Tech Report 2007-64

Happy Birthday, Dear Viruses

by Richard Ford and Eugene H. Spafford

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

attaches acetyl residues to proteins, thereby possibly modifying their properties (*12*). Other enzymes of this class are famous for their ability to change chromatin structure by acetylating histone proteins, but the physiological substrates of Eco1 are unknown. More strangely, its acetyltransferase activity is required for de novo establishment of cohesion in G_2 phase (*4*), but is not essential when cohesion is generated during S phase (*13*). To explain this conundrum, Ünal *et al*. speculate that acetyltransferase activity may only be required to reactivate Eco1 after DNA damage, but more work is needed to resolve the mystery surrounding the role of this enzymatic activity. Finally, it will be interesting to determine why yeast cells "strengthen" their preexisting cohesion on chromosomes in a genome-wide manner after DNA damage, and to understand how a single chromosome break triggers cohesion across the entire genome.

References

1. K. Nasmyth, C. H. Haering, *Annu. Rev. Biochem.* **74**, 595 (2005).
2. E. Watrin, J. M. Peters, *Exp. Cell Res.* **312**, 2687 (2006).
3. L. Ström *et al.*, *Science* **317**, 242 (2007).
4. E. Ünal, J. M. Heidinger-Pauli, D. Koshland, *Science* **317**, 245 (2007).
5. F. Uhlmann, K. Nasmyth, *Curr. Biol.* **8**, 1095 (1998).
6. C. H. Haering *et al.*, *Mol. Cell* **15**, 951 (2004).
7. L. Ström, H. B. Lindroos, K. Shirahige, C. Sjögren, *Mol. Cell* **16**, 1003 (2004).
8. L. Ström, C. Sjögren, *Cell Cycle.* **4**, 536 (2005).
9. E. Ünal *et al.*, *Mol. Cell* **16**, 991 (2004).
10. R. V. Skibbens, L. B. Corson, D. Koshland, P. Hieter, *Genes Dev.* **13**, 307 (1999).
11. A. Toth *et al.*, *Genes Dev.* **13**, 320 (1999).
12. D. Ivanov *et al.*, *Curr. Biol.* **12**, 323 (2002).
13. A. Brands, R. V. Skibbens, *Curr. Biol.* **15**, R50 (2005).

10.1126/science.1146072

COMPUTER SCIENCE

Happy Birthday, Dear Viruses

The first computer virus was created 25 years ago, but there is no end in sight to malicious software.

Richard Ford and Eugene H. Spafford

Birthdays and other anniversaries are often a time for celebration, as we reflect on milestones passed. In the world of computing, we have quite a few happy anniversaries: for example, the first computer (arguably, Babbage's design of 1822) and the first e-mail message sent (1965).

Some remembrances, however, are less positive, and 2007 marks the silver anniversary of a darker sort—the genesis of malicious computer viruses (*1*–*3*). In 1982, a virus written by a high-school student in Pittsburgh began appearing on Apple II systems. This virus—known as "Elk Cloner"—infected the operating system, copied itself to floppy discs, and displayed bad poetry. Primarily intended to be irritating, the virus came and went with little notice. Few people spent time worrying about the beastie, and almost nobody predicted that it was a harbinger of the current multibillion dollar antivirus industry.

CREDIT: PETER HOEY

From such humble beginnings, computer viruses—and, more broadly, "malware" programs—are now so ingrained in popular culture that they've become the butt of jokes in ads and talk shows. Although the malware problem grew slowly in the early 1980s, not much time passed before it really made the news. In 1988, the infamous "Morris Worm" spread worldwide, causing outages across the fledgling Internet. There was also the media storm surrounding the Michelangelo virus, which was set to trigger on 6 March 1992, threatening to destroy data on infected machines. Since then, SQL.Slammer, Code Red, Nimda, Concept, and Melissa all had their 15 minutes of fame and, in the process, collectively caused billions of dollars in damage.

The most talked-about risks from today's malware have a distinctly financial flavor. If the viruses and worms of the past decade were the online equivalent of graffiti artists, malware is now like criminals who wish to steal your wallet and forge your checks. This has led to much quieter attacks, because too much visibility would cut down on profits. Instead of displaying a message or erasing your hard drive, modern malware is more insidious, turning your machine into a relay for spam, a staging ground to attack other systems, or a spy capturing your bank account and credit card information—or all three.

Spyware, phishing, rootkits, and bots — the cutting-edge malware of today—are truly nasty, and considerable effort has been invested in their creation. It has become a significant criminal enterprise and supports a thriving underground economy.

Surely the scientific community has simply been too preoccupied to deal with this challenge and a good solution is available. Sadly, even after decades, it appears that no end is in sight. This stems partly from a subtle computational twist: Building a computer program that can tell with absolute certainty whether any other program contains a virus is equivalent to a famous computer science conundrum called the "halting problem." It has no solution in the general case and has no approximate solution for our current computing environments without also generating too many false results (*4*).

Popular opinion holds that malware is a Microsoft-only problem. Macs, for example, don't seem to suffer from malware as much as Windows, so perhaps everyone should switch to Macs. Linux and Unix, too, are often touted as obvious solutions. However, no system is fully immune.

Windows has suffered for a variety of reasons; Microsoft must take some responsibility for the problem. They made some architectural decisions that, in retrospect, left Microsoft products more vulnerable. Backwards compatibility meant that later releases kept old weaknesses. And Microsoft products have undergone intense scrutiny: It is the obvious software to attack as it is the dominant player in the desktop

---

R. Ford is at the College of Engineering and Computer Sciences, Florida Institute of Technology, Melbourne, FL 32901–6975, USA. E-mail: rford@se.fit.edu E. H. Spafford is at the Center for Education and Research in Information Assurance and Security, Purdue University, West Lafayette, IN 47907–2086, USA. E-mail: spaf@ cerias.purdue.edu

market. Macs and Unix systems would undoubtedly be more frequently attacked if they were dominant, although their underlying architecture and maturity might result in less (but not zero) success for attackers. It is definitely not the case that malware is a problem only because of consumers' choice of operating system (the "platform"). The truth is much more complicated and far more worrisome.

Diversity of platform is a double-edged solution in that it solves some problems neatly but creates new ones. For example, if we want some machines always running, diversity makes it very difficult for one attack to wipe out all available computers—some machines are always immune. The flipside is that diversity may actually increase the "attack surface": Although some machines are safe and secure, diversity may increase the chances that other machines are vulnerable to some other attack. Diversity is a boon for survivability but a potential risk in terms of network penetration.

There is one basic fact in security: The more functionality, the more opportunities a developer has to make a mistake. The simple truth is that modern computers are anything but simple—their increasing complexity is driven by consumers' thirst for functionality. Furthermore, computers are almost ubiquitous: For most people, the cell phones in their pockets are as much computers as are their laptops. Virulent cell-to-cell malware is not far off; researchers have already seen some limited "proof of concept" efforts. Personal digital assistants, music players, "smart" appliances, and more are all increasingly making use of available connectivity. Consumers and producers alike need to understand that more functionality means more risk. Unfortunately, no change is likely in the near term, and vendors will continue to add poorly thought-out code to their products.

Despite the best efforts of researchers, malware is not going to vanish any time soon. Computers are extremely difficult to secure, and humans are often the weakest link. For example, in one hoax users were encouraged to delete a particular file from their computers. Many users did exactly that and carefully followed the instructions to forward the warning message to all their friends. The file they deleted was critical to the system; the "virus" was executing in their minds. There is no obvious "fix" for human nature—that has not changed in many hundreds of years. Because of this, it seems likely that in another 25 years time, we will all be lifting our glasses to (or because of) malware once again.

References

1. D. Ferbrache, *A Pathology of Computer Viruses* (Springer, Berlin, 1991).
2. E. H. Spafford, in *The Encyclopedia of Software Engineering,* J. Marciniak, Ed. (Wiley, New York, 1994).
3. P. Szor, *The Art of Computer Virus Research* (Addison-Wesley, Boston, MA, 2005).
4. F. B. Cohen, *A Short Course on Computer Viruses* (Wiley, New York, 1994).

10.1126/science.1140909

GEOCHEMISTRY

Strange Water in the Solar System

Analysis of a primitive meteorite offers clues about the environment in which the solar system formed.

Edward D. Young

Cosmochemists use isotope ratios to understand the stellar environment in which our solar system formed. The most pronounced and mysterious of these ratios involve the three stable isotopes of oxygen, ^{16}O, ^{17}O, and ^{18}O. Normally, ^{17}O and ^{18}O separate partially from the more abundant ^{16}O according to their relative mass differences. Variations in the $^{17}O/^{16}O$ ratio are thus about half those of $^{18}O/^{16}O$. But many rocky materials in the solar system violate this expectation, exhibiting variations in isotope ratios that are independent of mass. This is most apparent in chondrite meteorites, which are remnants of primitive rocks accreted during the earliest stages of solar system formation.

This anomalous distribution of oxygen isotopes produces a distinctive line with slope equal to 1 on a plot of $\delta^{17}O$ versus $\delta^{18}O$ (*1*) rather than a slope of ~ ½ typical of oxygen reservoirs on Earth (see the figure). The cause of this "^{16}O anomaly" has been a mystery for three decades (*2*). Water, it seems, was a key player in the origin of the ^{16}O anomaly, and on page 231 of this issue, Sakamoto *et al*. (*3*) report evidence for the original isotopic composition of water in the early solar system. From this discovery come insights into the origin of the ^{16}O isotope anomaly and clues to the nature of the stellar nursery that gave birth to the Sun (*4*).

Many mechanisms have been proposed for producing the ^{16}O anomaly in the solar system. Perhaps we have simply inherited the isotope abundances as they evolved in our Galaxy (*5*). Or possibly the isotope ratios stem from chemically induced mass-independent fractionation, analogous to what happens during ozone production in Earth's atmosphere (*6, 7*).

Researchers have recently looked to light-induced destruction of CO as the cause. About half of the total oxygen in a protoplanetary disk like the one that produced our solar system resides in CO. Another third exists in the form of H_2O with the remainder as oxides of other elements (*8, 9*). Carbon monoxide absorbs ultraviolet (UV) light emanating from stars and is dissociated to C and O. In regions of the right gas density, UV absorption cleaves $C^{16}O$, $C^{17}O$, and $C^{18}O$ molecules in proportions inverse to their relative abundances, a process referred to as "self-shielding."

Because $C^{16}O$ is the most abundant of these isotope varieties, the oxygen liberated by CO photodissociation is ^{17}O and ^{18}O rich and ^{16}O poor. Clayton (*10*) suggested that CO self-shielding at the inner annulus of the solar protoplanetary disk might be the cause of the slope = 1 line on the $\delta^{17}O$ versus $\delta^{18}O$ plot (see the figure). Yurimoto and Kuramoto (*11*) suggested that CO photodissociation and self-shielding in the molecular cloud precursor to the solar system could have caused the ^{16}O anomaly. Lyons and Young (*12*) suggested that CO photodissociation at the surfaces of the protoplanetary disk might have been the cause (see the figure).

A key prediction of the CO self-shielding models is that O liberated by CO photodestruction reacted with H to form ^{16}O-poor H_2O (*11–14*). We know that water in the early solar system was depleted in ^{16}O relative to rocks, but the extreme depletions predicted by the CO self-shielding models were not observed. Estimates of the original oxygen isotope ratios of solar system water relied on inferences from the measured oxygen isotope ratios of "secondary minerals"

The author is in the Department of Earth and Space Sciences and the Institute of Geophysics and Planetary Physics, University of California, Los Angeles, Los Angeles, CA 90095, USA. E-mail: eyoung@ess.ucla.edu