DOI:10.1145/1897816.1897832

George Ledin, Jr.

Inside Risks
The Growing Harm of Not Teaching Malware

Revisiting the need to educate professionals to defend against malware in its various guises.

At the risk of sounding a byte alarmist, may I call to your attention the extreme threat to our world posed by cyberwar, cyberterrorism, and cybercrime? Cyberattacks are already numerous and intricate, and the unquestionable trend is up. To grasp the likelihood of these threats, consider the similarities between physical and virtual violence. Before attacking the U.S. on Sept. 11, 2001, terrorists rehearsed their assaults on a smaller scale at the World Trace Center and in several more distant venues.

Since that infamous date, paralleling physical attacks, cyberstrikes of increasing severity have been carried out against many targets. A few small nations have been temporarily shut down. These attacks are proofs of concept waiting to be scaled up. I hope cybersecurity is on governments' front burners. We ought not wait to react until a devastating cyber-onslaught is unleashed upon us.

Six years ago I wrote a *Communications* Inside Risks column urging that viruses, worms, and other malware be taught ("Not Teaching Viruses and Worms Is Harmful," Jan. 2005, p. 144). The goal of that column was to involve future generations of computer professionals in the expanding global malware problem and persuade them to help curb it. Six years later, malware is still not being taught. And the problem is now much worse.

Malware Evolution

During the first decade of the 21st century the malware problem has evolved in two significant ways. Gone are the lethal but simplistic payloads, produced by improvised, amateur scripts. Gone also are the idiots savants who cut-and-pasted such scripts. Carders, script kiddies, spammers, identity thieves, and other low-level miscreants will probably and deplorably never be completely gone. Gangs of much better trained programmers have largely replaced the individual crooks and nuisance makers. These gangs ply their trade for or in behalf of political syndicates, organized crime cartels, and government-sanctioned but unacknowledged dark ops. Some nation-states covertly train and support them.

What began as gross mischief evolved into criminal activity. Rather than erasing a hard disk drive, why not steal the data stored on it? Or encrypt the drive and extort a ransom for decrypting it? Or hijack the users' computers? Today's malware is a killer app: obfuscated, often; clumsy, never. A medley of viruses, worms, trojans, and rootkits, it is clever, enigmatic—a sly hybrid. Its bureaucratic components (such as installers and updaters) are examples of automated elegance.

Today's malware is a killer app: obfuscated, often; clumsy, never.

Identity theft, botnetting, and many other forms of trespass and larceny continue. Coupled with negligence by institutions that are supposed to safeguard our privacy, the picture is bleak. Malware launchers seem to be always ahead. And their products are no longer stupid capers but skillful software packages. These are valuable lessons that are not being understood by us, the victims.

Malware perpetrators have clearly mastered these lessons. Trading local pranks for global villainy, the perps are readying their next steps on the international political stage, where cyberspace is a potential war zone in-the-making. Inadequately capable of defending ourselves from being burgled, we are easy targets for evil geniuses plotting fresh hostilities.

We cannot protect ourselves from what we do not know. We must not remain stuck in a weak, purely reactive, defensive mode. New malware should no longer be an unexpected, unpleasant surprise. And we must be embarrassed when anti-malware products cause more problems than they solve. As human beings, we have a duty to make our world a better place. As computer professionals, we must do our fair share to stanch malware and prevent cyberwar.

Dealing with Malware

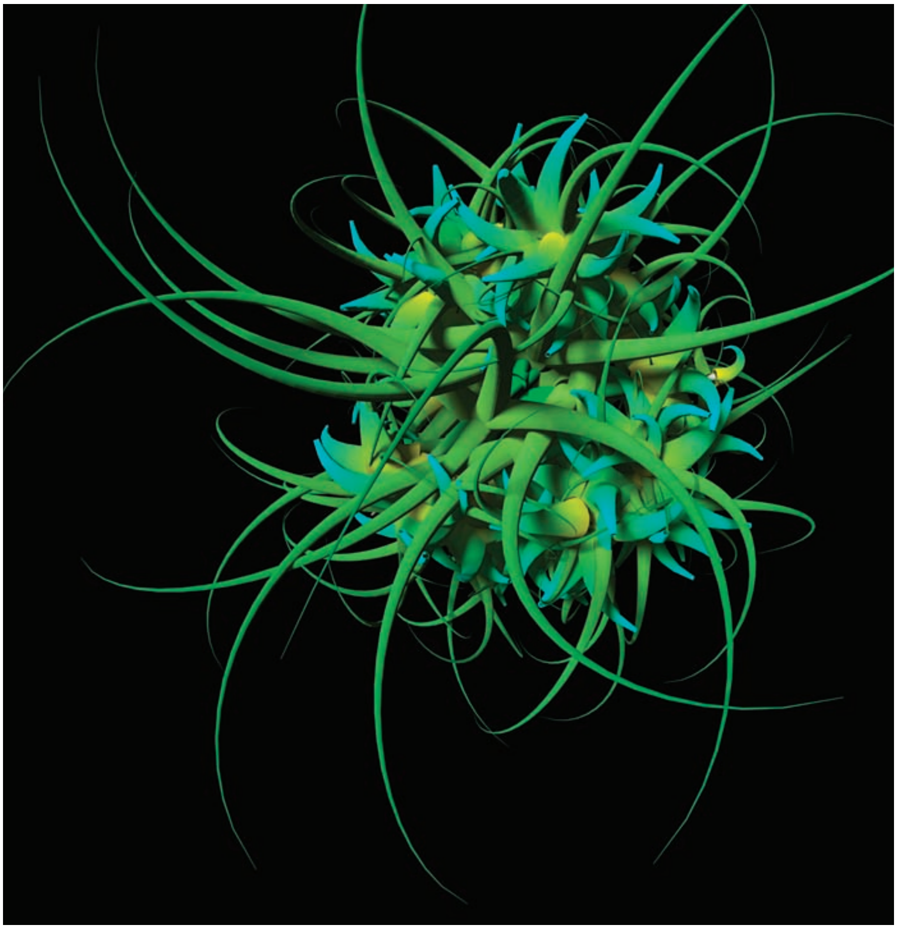
The malware problem must be dealt with on many fronts, proactively. Ideally, we should anticipate and be prepared for new malware. On the research front, funding agencies should follow DARPA's example. If synthetic genomics—the fabrication of new genetic material—merits $50 million in grants per year, so should exploration of new, novel, innovative malware.

University classrooms and laboratories should serve as locations for spreading malware literacy. Understanding is achieved only by doing. The most effective way to comprehend something is to program it. We cannot afford to continue conferring degrees to computer majors who have never seen the source code of viruses, worms, trojans, or rootkits, never reversed any malware binaries, and never programmed their own malware.

Standard undergraduate computer science curricula offer courses on many disparate topics, such as artificial intelligence and database systems. Students graduating with a degree in computer science are expected to have a solid acquaintance with various subjects that may not be their chosen specialty. Some graduates will dig deeper and become adept at these topics, but the mere fact that these topics are routinely taught to all undergraduate majors is in itself beneficial, because future computer professionals should not be completely ignorant in fields outside their areas of concentration.

Teaching malware will not turn our students into specialists. Malware literacy is not malware expertise. However, unlike artificial intelligence or databases, unfortunately malware is not a standard undergraduate course or even a regular part of an elective computer security course. (Syllabi of computer security courses may pay lip service to diverse issues, including malware, but such courses are overwhelmingly concerned with cryptography.) This means we are matriculating computer scientists whose knowledge of malware is roughly on a par with that of the general population of amateur computer users.

Six years and many articles, interviews, and blogs later, the question, "Should we teach malware?" still evokes apprehension, trepidation, even dread. The answer, of course, is, "Yes, we should." Indeed, we must! It would be irresponsible not to have a single course dedicated exclusively to malware, or a course that studies vulnerabilities in general and malware in particular, or some other combination, so that students completing the course will gain a deeper understanding of malware.



Visualization derived from disassembled code of MyDoom worm.

VISUALIZATION BY ALEX DRAGULESCU

The apprehension, trepidation, and dread will not go away easily. Spreading viruses, worms, Trojans, and rootkits is dirty business. Programming them may feel like doing something forbidden. Over the past six years, I've heard many concerns about the ethics of teaching malware. Taboos are difficult to dispel. For example, the prohibition of dissecting cadavers held back medicine for centuries. How else could aspiring physicians and surgeons learn anatomy? Today, life science majors are not necessarily bacteriologists, parasitologists, or virologists, but all enjoy the benefit of a standard curriculum that offers exposure to microbiology theory and its laboratory practice. This is not the case with computer science majors, whose curricula omit theory and programming of malware. Sadder yet, undergraduates learn sorting, database, and other theories, and carry out their corresponding programming assignments, but do not take a similarly rigorous course on malware.

Six years ago, when I proposed that not teaching malware was harmful, I was worried that new malware would attain greater sophistication, become much more complex, and that its force and impact would be felt more widely than those of its predecessors. Well, guess what? It has!

The reason we cannot solve the malware problem is simple: We don't have a theory of malware. There are textbooks on sorting and searching, on database methods, on computer graphics. These textbooks present algorithms and source code listings. The many different techniques of sorting, for example, are analyzed and their implementations are examined thoroughly. Students are encouraged to explore new approaches to sorting, to improve on what is known, to push the limits of performance. Whereas such explorations are standard practice in areas such as sorting, they do not exist for malware. Malware was absent from nearly all undergraduate curricula six years ago and it is still absent, for essentially the same technical and ideological reasons.

Technical and Ideological Requirements

On the technical side, teaching malware requires knowing viruses, worms, Trojans, and rootkits, which obligates teachers to have read their source code, which in turn requires them to have the ability to reverse the binaries, and the facility to launch, run, and infect machines on an isolated subnet. Having read a sufficiently large, representative sampling of historic malware source code then leads to formulating various generalizations to build a theory of malware that can be tested by writing derivative malware, new in a shallow sense but not necessarily innovative. These experiences then should culminate in inventing never-before-tried malware to foresee trends in cyberspace.

On the ideological side, arguments range from "moral purity" to "allocation of responsibility." These arguments are fueled by fear of the unknown, especially when the unknown is potentially toxic. Having one's reputation ruined by being labeled irresponsible, negligent, reckless, or incompetent is a strong disincentive. It is difficult to imagine computer scientists losing their professional standing or community esteem by demonstrating new multi-core implementations of Batcher's sort, especially if it beat all current sorting techniques; but it is not difficult to conjure the poisonous politics of unveiling new malware that would escape detection by all current commercial anti-malware products. Raising the stakes with powerful sorting algorithms is a laudable, honorable endeavor; casting a spell with powerful new malware is considered undignified per se.

The reason we cannot solve the malware problem is simple: We don't have a theory of malware.

Detecting and arresting malware and its launchers won't be easy unless we ramp up on all fronts, especially education.

That malware should be taught to computer science majors runs into a frequent and bothersome accusation—that we will be granting diplomas to hordes of malicious hackers, aiding and abetting greater misbehavior than is being suffered already. Physicians, surgeons, nurses, pharmacists, and other health professionals have the know-how with which to inflict pain, torture, and death. Every profession may have its "black sheep," but it is obvious that society benefits by having an absolute majority of responsible and caring professionals.

Conclusion

I began this column by calling your attention to the forthcoming triple trouble of cyberwar, cyberterrorism, and cybercrime. The last of the three—cybercrime—is abundantly in our midst already. The other two menaces are works in progress. All three typically deploy via malware. (Human gullibility is, tragically, a contributing factor.) The preferred way thus far has been to exploit overlay networks or saturation-bomb regions of the Internet to build a broad-based infrastructure of illegally tenanted user machines and servers—a large botnet, responsive to peer-to-peer and command and control communications. Such a botnet's unwitting foot soldiers—your and my machines—are powerful weapons in cyberspace, capable of mounting targeted distributed denial-of-service attacks against individual users, institutions, corporations, and governments. Botnets built by worms can remain silent and undergo quiet maintenance and upkeep between bursts of activity. Botnet battles—territorial disputes and turf fights—are vicious confrontations for supremacy, worth billions of dollars and euros. For nation-states, the cyber-arms-race is on: those with the strongest malware will emerge as super-cyber-powers. None of these near-future developments can be wished away. And we continue to harm ourselves by not teaching malware.

May we let thousands of talented young minds lie fallow until our ignorant denial of the problem can no longer be condoned? How much malware damage should we tolerate? Until universal infection is the status quo? How are we to respond to massive but very likely covert malware pandemics? Would our response be capable of restoring and maintaining stability? More importantly, would we be able to verify the effectiveness of such a response?

Detecting and arresting malware and its launchers won't be easy unless we ramp up on all fronts, especially education. Millions of educated professionals are our best defense. Classrooms can be constructive idea generators. Let's not wait another six years for important ideas, such as malware prevention and preemptive interdiction, to be realized. C

George Ledin, Jr. (george.ledin@sonoma.edu) is a professor of computer science at Sonoma State University and a visiting fellow at SRI International.