15

Viruses Are Good for You

Julian Dibbell

What scares you most about getting that virus?

Is it the prospect of witnessing your system's gradual decay, one nagging symptom following another until one day the whole thing comes to a halt? Is it the self-recrimination, all the useless dwelling on how much easier things would have been if only you'd protected yourself, if only you'd been more careful about whom you associated with?

Or is it not, in fact, something deeper? Could it be that what scares you most about the virus is not any particular effect it might have, but simply its assertive, alien presence, its intrusive otherness? Inserting itself into a complicated choreography of subsystems all designed to serve your needs and carry out your will, the virus hews to its own agenda of survival and reproduction. Its oblivious self-interest violates the unity of purpose that defines your system as yours. The virus just isn't, well, you. Doesn't that scare you?

And does it really matter whether the virus in question is a biological or an electronic one? It should, of course. The analogy that gives computer viruses their name is apt enough to make comparing bioviruses and their digital analogs an interesting proposition, but it falls short in one key respect. Simply put, the only way to fully understand the phenomenon of autonomously reproducing computer programs is to take into account their one essential difference from organic life forms: they are products not of nature but of culture, brought forth not by the blind workings of a universe indifferent to our aims, but by the conscious effort of human beings like ourselves.

Why then, after a decade of coexistence with computer viruses, does our default response to them remain a mix of bafflement and dread? Can it be that we somehow refuse to recognize in them the traces of our fellow earthlings' shaping hands and minds? And if we could shake those hands and get acquainted with those minds, would their creations scare us any less?

These are not idle questions. Overcoming our fear of computer viruses may be the most important step we can take toward the future of information processing. Someday the Net will be the summation of the world's total computing resources. All computers will link up into a chaotic digital soup in which everything is connected—indirectly or directly—to everything else. This coming Net of distributed resources will be tremendously powerful, and tremendously hard to harness because of its decentralized nature. It will be an ecology of computing machines, and managing it will require an ecological approach.

Many of the most promising visions of how to coordinate the far-flung communication and computing cycles of this emerging platform converge on a controversial solution: the use of self-replicators that roam the Net. Free-ranging, self-replicating programs, autonomous Net agents, digital organisms—whatever they are called, there's an old fashioned word for them: computer viruses.

Today three very different groups of heretics are creating computer viruses. They have almost nothing to do with each other. There are scientists interested in the abstract behaviors of self-replicating codes, there are developers interested in harnessing the power of self-replicating programs, and there are unnamed renegades of the virus-writing underground.

Although they share no common experience, all these heretics respect a computer virus for its irrepressible mobility, for the self-centered autonomy it wrests from a computer environment, and for the surprising agility with which it explores opportunities and possibilities. In short, virus enthusiasts relate to the virus as a fascinating and powerful life form, whether for the fertile creation of yet more powerful digital devices, as an entity for study in itself, or, in the case of one renegade coder, for reckless individual expression.

Getting a Buzz from the Vx

One computer virus writer in his early 20s lives on unemployment checks in a white, working-class exurb of New York City. He tends to spend a fair amount of his leisure time at the local videogame arcade playing Mortal Kombat II, and would prefer that you didn't know his real name. But don't let the slacker resume fool you: the only credential this expert needs is the pseudonym he goes by in the computer underground: Hellraiser.

Hellraiser is the founding member of the world-renowned virus-writers' group Phalcon/Skism. He is also creator of 40Hex, an electronic zine whose lucid programming tips, hair-raising samples of ready-to-run viral code, and trash-talking scene reports have done more to inspire the creation of viruses in this country than just about anything since Robert Morris Jr.'s spectacularly malfunctional worm nearly brought down the Internet.

And as if all this weren't enough, Hellraiser also comes equipped with the one accessory no self-respecting expert in this cantankerous field can do without—his very own pet definition of computer viruses. Unlike most such definitions, Hellraiser's is neither very technical nor very polemical, and he doesn't go out of his way to make it known. "Sure," he'll say, with a casual shrug, as if tossing you the most obvious fact in the world: "Viruses are the electronic form of graffiti."

Which would probably seem obvious to you too, if you had Hellraiser's personal history. For once upon his teenage prime, Hellraiser was also a hands-on expert in the more traditional forms of graffiti perfected by New York City youth in the 1980s. Going by the handle of Skism, he roamed the city streets and train yards with a can of spray paint at the ready and a Bronx-bred crew of fellow "writers" at his side, searching out the sweet spots in the transit system that would give his tag maximum exposure—the subway cars that carried his identity over the rails, the truck trailers that hauled it up and down the avenues, and the overpasses that announced it to the flow of travelers circulating underneath.

In other words, by the time Hellraiser went off to college and developed a serious interest in computers, he was already quite cozy with the notion of infiltrating other people's technology to spread a little of himself as far and wide as possible. So when he discovered one day that his PC had come down with a nasty little digital infection, his first thought was not, as is often customary, to curse the "deviant hackers," "sociopaths," and "assholes" who had written the program, but to marvel at the possibilities this new infiltration technique had opened up. Street graffiti's ability to scatter tokens of one's identity across the landscape of an entire metropolis looked provincial in comparison. "With viruses," Hellraiser remembers thinking, "you could get your name around the world."

He was right. The program that had infected his own computer in late 1990, the so-called Jerusalem virus, had spread from Italy to Israel to North America before finally making its way into the pirated copy of Norton Utilities that brought it to Hellraiser's hard drive. And though Jerusalem's author remained uncredited, other programmers from nearly every corner of the globe were pulling off feats of long-distance self-aggrandizement that dwarfed anything within the reach of America's spray-paint commandos. A kid who called himself Den Zuk had launched a virus that was flashing his handle on computer screens all over Europe, the U.S., and South America. Early speculation placed its origin in Venezuela, but the virus was eventually tracked to its true source in Bandung, Indonesia, when a researcher in Iceland guessed that some enigmatic characters in the source code were in fact a ham-radio call sign; contact was made with the call sign's registered operator, who immediately copped to his authorship of the program.

Equally far-ranging was the journey of the Joshi virus, which spread from India to parts of Africa and on to the rest of the world, popping up every January 5th to command computer users to type "Happy Birthday Joshi" if they wanted control of their systems back.

What impressed Hellraiser as much as the vast geographic distances covered by viruses, however, was their long range over time. After all, a painted graffiti tag would only last as long as it took to fade away or be painted over, but viruses, it seemed, might replicate forever in the wild. Indeed, the Jerusalem virus had been doing so for three years before Hellraiser encountered it, and four years later it remains one of the world's most commonly reported viruses. Likewise, Den Zuk is still reproducing on computers worldwide six years after it first left the island of Java; Joshi continues for the fifth year in a row to extort international birthday wishes. Dozens of other viruses from the U.S., Canada, Eastern Europe, Taiwan, Australia, Turkey, Malta, and other far-flung locales thrive globally. (This despite the fact that the antivirus industry spends tens of millions of dollars a year to eradicate them.) Bearing encoded bits of their authors' souls—clever jokes, crude graphics, friendly greetings, and, of course, occasionally, malicious intentions (though in fact the majority of viruses found in the wild are designed to do no damage)—viruses roam the earth in apparent perpetuity.

For Hellraiser, steeped as he was in graffiti culture's imperative to "get the name across," there was only one possible response to this new technology of self-projection: he had to get in on the action. But how? Virus writing wasn't exactly a standard subject in computer-science courses, and even the computer underground—with its loose-knit network of bulletin boards and e-zines proffering instruction in the illicit arts of hacking and phone phreaking—wasn't the most dependable source of virus lore. Occasionally, a hack and phreak board might offer a small collection of cryptic viral source code for brave souls to experiment with, but as far as Hellraiser knew, the only system exclusively devoted to viruses at the time was a place called the Virus Exchange, operating out of what was then the world's epicenter of virus production: post-Communist Bulgaria, where the Cold War's endgame had left a lot of overtrained programmers with time on their hands and anarchy on their minds.

Lacking the money or the phreaking skills to dial in to the Virus Exchange, Hellraiser made do with what he did have: a live specimen of the Jerusalem virus, replicating furiously inside his desktop system and poised to trash every program file he tried to run on any upcoming Friday the 13th. Carefully, Hellraiser extracted all copies of the virus from the computer and holed up in his dorm room to examine its workings. He studied it for weeks, and then finally, tentatively, he produced a virus of his own. It was a shameless hack really, essentially just the Jerusalem code with the tag line "SKISM-1" inserted in place of a few of the original characters. But after infecting as many computers as he could and subsequently finding his creation enshrined in antivirus literature as the "Skism-1" virus, Hellraiser swelled with a pride he would later recall with some amusement: "Shit, I thought I was the man back then."

Hooked on that buzz, he dove deeper into his studies, aiming for proficiency in DOS assembly language, the formidably austere low-level programming dialect in which Jerusalem was writ-

ten (like the vast majority of computer viruses then and now). He quickly acquired the ability to produce viruses he could truly say were his, and along with this ability, he picked up the beginnings of a rep among New York-area denizens of the underground. Gradually, through the hack/phreak (h/p) bulletin-board scene, he made contact with other isolated virus writers—subculture orphans compared with the h/p crowd and its Legions of Doom, MODs, Chaos Clubs, and other constantly forming and re-forming groups and factions.

Hellraiser started wondering why he shouldn't put together a group of his own. Soon enough, the retired graffiti bomber was again running with a crew, formally known as Smart Kids Into Sick Methods (Skism for short) and dedicated to sharpening the virus-writing skills of both its members and the virophilic public at large.

And it was to serve more or less those lofty ends that Skism's electronic house journal 40Hex was born. Named for the assembly-language function by which viruses copy themselves, the publication hit the boards of the Vx underground with an infectiousness all its own. (Vx, short for virus exchange, denotes all boards devoted, like their Bulgarian namesake, to virus discussion and traffic in viral source code.) Its unapologetic bad attitude was a brash wake-up call to the still-embryonic virus-writers' community. "This is a down and dirty zine [which] gives examples on writing viruses and . . . contains code that can be compiled to viruses," wrote Hellraiser in the introductory file of 40Hex's March 1991 première. "If you are an antivirus pussy, who is just scared that your hard disk will get erased so you have a psychological problem with viruses, erase these files. This aint for you."

The warning scared off no one, of course, least of all the alleged pussies of the antivirus industry, who took to scouring every new issue for a peek inside the mind of the enemy, getting up close and personal at last with the phantoms they'd been battling for years. Not that the life of the virus hunter was a lonely one. In fact, the antivirus community was already in many ways a more advanced subculture than that of the virus writers, complete with local color and a mystique all its own: the industry pioneer and media darling John MacAffee was famed for his giddy morning-after overestimation by a factor of 10 of the Internet worm's damage; then there were those Bulgarians, the notorious and proud Dark Avenger—who signed, and even dedicated, his viruses—and his driven nemesis, Vesselin Bontchev. Endlessly revising and debating the burgeoning taxonomy of virus species, nervously policing the boundary between the great unwashed and those trustworthy enough to handle "live" specimens, the world of antivirus research offered its initiates a thrill somewhere between the delightful romance of butterfly collecting and the grim camaraderie of working for the National Security Agency.

In comparison, virus writing—while obviously not without its kicks—lacked community. But in the months and years following 40Hex's début, that began to change. The previously inchoate and virtually invisible virus-writing underground at last coalesced and shifted into high gear. Various groups proliferated and crossbred: Skism merged with another New York posse called Phalcon to form the Phalcon/Skism supergroup, while the pan-European TridenT team and the Canadian-Australian-Swiss-Taiwanese-multinational NuKE crew quickly rose to challenge Phalcon/Skism's prestige and programming skills. Zines multiplied, too: NuKE's Info Journal and West Coast virus writer Urnst Kouch's Crypt Newsletter challenged 40Hex's hegemony, as did the number of so-called Vx bulletin boards that rocketed from a handful worldwide to rough estimates of as many as 200 at present.

Amid all the rapid growth it helped set in motion, 40Hex has kept pace. After the first four raucous issues, Hellraiser handed over the editorial reigns to Phalcon's designated archivist, Garbage Heap, who has steadily increased the circulation of the zine while slowly steering it toward something suspiciously like respectability. Available now in a crisp, desktop-published paper edition as well as good old-fashioned e-text, today's 40Hex still brims with the gnarliest of viral code and remains a feisty defender of the right to create and publish viruses. But it frowns on anyone

who looses viruses into the wild and is more likely to solicit guest editorials from antivirus types than to hurl obscenities at them.

The young hellion who founded the zine would probably not approve—that is, if the same young hellion were still around to say anything about it. But he isn't. Not really. Hellraiser has undergone some changes of his own lately. Once quite cavalier about releasing viruses that intentionally deleted files or otherwise "fucked people's shit up" (after all, what better way to make your tag linger on in their memory?), he eventually decided that creating destructive programs just gave virus writing a bad name and resolved thenceforth to produce viruses with more or less benign payloads only. And then one day, not too long ago and without much fanfare, he simply called it quits. Partly, he was starting to chafe at the limited range of programming challenges involved in virus creation, he says, but more to the point, his evolving young world view had somehow gotten infected by a creeping respect for the right of others to control what goes into their own digital back yards. Destructive payload or no destructive payload, Hellraiser reached the conclusion that it was just plain "wrong to 'pollute' other people's systems with viral garbage."

Which isn't to say he's gone over to the ranks of his old antivirus nemeses. Hardly. He's still too tight with all his Phalcon/Skism homeboys for that. Even if he weren't, he's been a virus writer for too long to feel comfortable with the easy demonizations that are the stock in trade of antivirus rhetoric. For the rest of us, of course, it's easy enough to accept the standard caricature of the underground virus writer as a low-grade sociopath. After all, what else but antisocial perversity could lead someone to produce a mechanism we encounter principally as contamination in the digital environment, as noise on the line?

Yet Hellraiser's career path—from graffiti writing to virus writing and beyond—demands a more complicated understanding of the virus phenomenon. It asks us to recognize that viruses, like graffiti, are just as much signal as noise—that they are in fact an irreducible confusion of the two. As Hellraiser came to recognize, the noisiness of viruses is built in—they are by definition information that subverts control. But as the subculture Hellraiser helped build will always remember, every virus turned out into the computer wilds—like every tag sprayed onto the hard urban landscape—is also a carrier for the purest and strongest signal a human being can send. "Remember my name," the virus says, which—after all—is another way of saying: "I'm alive."

This is about as far as most discussions of virus writing get: ignorant kids thrashing about in codes, creating horribly simple but efficient digital bombs. And even if you take a very generous view that the underground virus writers are inadvertently creating new forms of life, the discussion of beneficial viruses would have to stop here if it weren't for folks like Dr. Mark A. Ludwig.

The Mutator in the Desert

Mark Ludwig lives in a desert, and compared to Hellraiser's background, seems to hail from an entirely different planet. But Ludwig, too, is chasing the elusive nature of computer viruses.

A married man with three young children, Ludwig lives in Tucson, Arizona, where barrens of sand and sun and saguaro cactus shimmer not too far beyond the sump-cooled confines of his home. But the desert where he wanders is someplace else entirely: it's the lonely intellectual wilderness reserved for those who practice science on the fringe, outside the cozy realms of institutional affiliation, professional consensus, or methodological decorum.

He doesn't have to be there. With his PhD in physics from the University of Arizona (and his prior course work at Cal Tech and MIT), Ludwig could easily return to the fold of respectable researchers if he chose. All he'd have to do is let go of his somewhat obsessive scholarly pursuit of the wild computer virus, and pick a slightly more conventional object of study. Or maybe just pursue his present subject with a little more sober attention to devising antivirus countermeasures and a lot less gleeful fascination with viruses in and of themselves. Or maybe just tone down the florid

libertarian rhetoric and sweeping philosophical claims in which he tends to couch his otherwise gruelingly meticulous analyses of viral performance and technique.

Really, it wouldn't take much.

But Ludwig isn't likely to do any of these things, because he actually seems to prefer the hardships of the fringe to the rewards of a life on the techno-scientific inside.

He didn't always. "Once I was a scientist of scientists," writes Ludwig in the introduction to his latest self-published treatise, *Computer Viruses, Artificial Life, and Evolution*. "Born in the age of Sputnik, and raised in the home of a chemist, I was enthralled with science as a child. If I wasn't dissolving pennies in acid, I was winding an electromagnet, or playing with a power transistor, or doing a cryogenics experiment—like freezing ants—with liquid propane." Eager to work his way into the company of "the great men of science" and join their noble quest for objective Truth (he'd read about it in textbooks), Ludwig rushed through his undergraduate work at MIT in two years, then plunged into his graduate course of studies with equal enthusiasm. By the time he got his doctorate, however, he'd seen enough of the political infighting and blind prejudice that structure the real work of contemporary scientific investigation to sour the romance permanently. Disillusioned, he dropped out of the hard-sci grind and into a job working with computers, a field that at least provided some of the wide-open pioneering spirit that the textbook histories of science had promised, even if it moved him further from pure science's intimacy with the mysteries of nature.

But not long after that, around 1988, he started picking up reports of contagious programs running loose among the machines he now made his living from, and the course of his life changed yet again. For Ludwig, viruses came bearing the same mind-expanding message-in-a-bottle they would not much later be bringing to Hellraiser. Except that Ludwig decoded the message a little differently. Where Hellraiser heard the signal "I'm alive" coming from the virus's creator, Ludwig understood the message as coming directly from the virus itself. Viruses behaved like living things: self-reproducing and autonomous. Might we not understand life a little better, he wondered, if we could create something similar, and study it, and try to understand it? The mysteries of nature, in other words, now loomed closer than ever—right there on the wide-open technological frontier to which he'd fled from the wreckage of his scientific aspirations—and Ludwig couldn't resist the temptation to go questing after them once more.

His initial attempts to acquire specimens to observe were frustrating. Today's teeming ecology of one-stop Vx trading posts didn't exist. When Ludwig approached the antivirus community for access to its shared research collections, he found himself shut out: then as now, the A-V crowd refused to release captured virus code to anyone outside a trusted inner circle. So, true to his style, Ludwig decided to go it alone. He set up a BBS, announced a bounty of US$25 for every virus uploaded, and sat back while the code rolled in. After building up a representative cross section of the wild virus population, he set about examining his haul, and within a few months his research bore its first fruit: *The Little Black Book of Viruses*, a technical primer on the essentials of virus writing, complete with scrupulously annotated source code for four virus programs of his own creation.

The Little Black Book made something of a name for Ludwig, but it wasn't an especially pretty one. Though the tutorial viruses were pointedly nondestructive and came surrounded by warnings against their misuse and instructions on how to keep them from getting loose, the book was roundly condemned as an incitement to digital vandalism. In the three years of steady sales since *The Little Black Book*'s original publication in 1991, various mainstream computer magazines have summarily dropped Ludwig's advertisements for the book as inappropriate subject matter for their audiences. And when the book was recently released in France (as *Naissance d'un Virus*, or *Birth of a Virus*), its publishers there were immediately slapped with a legal injunction against distributing it with the infectious source code intact.

But Ludwig has remained undaunted in the face of the world's virophobia. If anything, its vehemence has only sharpened his determination to share the wealth of his knowledge. "People think of viruses as an invasion from Mars," he says, "and that hurts research into these things. My aim is to change people's attitudes, to cut down some of the fear."

To that end he has established an annual international virus-writing competition, flying cheerfully in the face of the "swarming hordes of antivirus developers." (One year's contest rewarded the smallest functional DOS virus submitted.) Ludwig also publishes a newsletter now, *Computer Virus Developments Quarterly*, in which he mingles detailed technical discussion of viral code with rants against the tyrannical tendencies of American government, the moral bankruptcy of contemporary Western culture, and (last but not least) the evils of repressing detailed technical discussion of viral code. Occasionally he even gets a sign that the general public is starting to come around to his pro-knowledge agenda: after five months of wrangling its way through the French courts, for instance, the suit against *Naissance d'un Virus* was finally thrown out by a tribunal arguing, as Ludwig proudly reports, that "trying this case was like putting Galileo on trial again."

Yet amid all of Ludwig's busy agitation in defense of viruses, what ever became of the intellectual mysteries that first drew his attention to them? His pleasure at being compared to Galileo, the archetype of the politically incorrect scientist, certainly suggests that he never lost his sense of scientific mission. But the proof of Ludwig's abiding interest in viruses as tools of natural philosophy lies in his sequel to *The Little Black Book*: the aforementioned *Computer Viruses, Artificial Life, and Evolution*. Published late in 1993, the book is a dense and daunting 373 pages' worth of charts, differential equations, and tightly reasoned arguments in support of Ludwig's intuition that self-reproducing computer code bears deep lessons about the workings of life.

As the title's nod to the fashionable new scientific discipline of artificial life makes plain, however, Ludwig is clearly aware that other researchers, backed by the imprimatur of Official Science, have been building on the very same intuition for some time now. The first two volumes of the Santa Fe Institute's *Proceedings on Artificial Life,* published in 1989 and 1992, devote several papers to the idea of computer viruses as synthetic life. But taking the idea further, Ludwig argues that computer viruses, unlike such other forms of artificial life as cellular automata, mobots, or genetic programming, are the only form of artificial life not biased by the hope of their creators. Because computer viruses must exist in an environment (DOS in particular) that was designed without any thought of the digital organisms that might come to inhabit it, they are free from any accusation that the environment's "physics" were written to support the emergence of their lifelike behavior. Or to put it more bluntly, feral viral ecologies (versus the controlled experiments in university labs) represent the only known simulation of life that does not implicitly (and quite unscientifically) build God into the system.

Having carefully constructed this ambitious claim, Ludwig proceeds to test drive it straight into the heart of biology's most vexing questions: How did life get here in the first place? How did the staggering diversity of life forms that exists today come to be? He sics viruses on the theory of evolution itself, in other words, sending them in to illuminate with their logical simplicity the still murky depths of Darwin's grand hypothesis. It's a bold move, but a puzzling one at first glance. Although the viruses found in the wild may exhibit a wide range of lifelike features, they've never been known, after all, to evolve.

Or have they? Not too long after the first virus was written, the first antivirus program was written as a countermeasure. Once anti-virus software was introduced into the cybernetic ecology, viruses and the programs that stalk them have been driving each other to increasing levels of sophistication. This is nothing less than the common coevolutionary arms race that arises between predators and prey in organic ecosystems.

Step one in this quasi-Darwinian dance took place when security-minded programmers developed what has since become the standard defense against viruses for most PC owners—scanning software that looks for telltale code fragments of known viruses (often some scrap of graffiti-esque text) and alerts the user when it finds any. In time, virus hackers responded by wrapping their programs in a blanket of encryption impenetrable to scanners. But since the built-in subroutines that decrypt the programs for execution cannot themselves be enciphered, antivirus programmers simply retooled their scanners to look for the decryption code. Later, in step two, the legendary Bulgarian writer Dark Avenger came up with a clever innovation known as a mutating, or poly-

morphic, virus. A mutating virus randomly reorganizes its decryption algorithm every time it replicates to outsmart the policing of the scanner. In step three, antivirus engineers devised "heuristic" scanners, built to sniff out all but an insignificant percentage of a virus' mutants through educated pattern recognition.

Surveying the fossil record of this game, Ludwig found himself pondering a logical next move: what if someone were now to develop a strain of polymorphs with a genetic memory, so that rather than completely reshuffling their structure with every generation, the few mutants that escape discovery by heuristics could pass their undetectable code on to their offspring?

The prospect of virus populations able to autonomously build up immunity to any scanning techniques thrown at them thoroughly depressed antivirus programmers. To Ludwig, however, the possibility proved too intriguing to wait around for some random underground hacker to realize it, and he resolved to do the job himself. The result: Ludwig's "Darwinian Genetic Mutation Engine," a programming utility that turns any normal DOS virus into a souped-up, genetically evolving polymorph, complete with an option for sexual gene-swapping between individuals that come into contact in the wild. Curious hackers can find the Darwinian Genetic Mutation Engine's complete source code in the pages of *Computer Viruses, Artificial Life, and Evolution*, along with detailed experimental results demonstrating the ability of Darwinian Genetic Mutation Engine-enhanced viruses to run rings around existing scanners. But the program's deeper significance, of course, lies in its potential to transform viruses' heretofore hacker-driven pseudo-evolution into something very like the real thing: a finely tuned interaction of variety and natural selection that allows the environment itself to shape the internal code of the organisms dwelling in it.

The Darwinian Genetic Mutation Engine is all Ludwig needs, in other words, to prove viruses capable of meaningful evolution, and incidentally, test Darwin's theory. And it's no surprise perhaps, given Ludwig's hard-earned distrust of anything smacking of intellectual orthodoxy, that he has found that Darwin's venerable theory fails the test. Running his beloved viruses through assorted experimental hoops and mazes, Ludwig followed them to the conclusion that Darwinian evolutionary mechanisms alone are just not mathematically fertile enough to have created and shaped life as we know it. This is a well-worn scientific heresy, of course, but it's not without its small but respectable following within the ivory walls Ludwig so proudly dismisses.

To be fair, though, Ludwig is not asking to be ranked among his boyhood heroes—those scientific greats whose unique insights clear broad new vistas of understanding in a single bound. All he wants from the rest of the world is a modicum of respect for the wild computer virus as a legitimate subject of scientific investigation. Or at least acknowledgment that this enduringly lifelike wonder could be useful if we but understood it, rather than the casting of it as the ultimate technological taboo.

Ludwig managed a remarkable intellectual shift. He elevated the computer virus from the digital equivalent of a can of spray paint to an object capable of perhaps infinite variations and almost lifelike behavior. He transformed a tool of vandals into a field of scientific study by emphasizing a computer virus' biological affinity. But by the time Ludwig began publishing, the computer virus was already well on its way from the fringes of science to the seat of honor at research symposiums.

Booting Up the Cambrian Explosion

"I'll be out at my place in the jungle over the weekend," said the message, posted in May 1994 from an obscure Internet site in Central America, "so I'll be out of e-mail contact till Monday."

And just like that, University of Delaware ecologist Tom Ray (now visiting scholar at the Advanced Telecommunications Research Institute International in Kyoto, Japan) disappeared once more into the rain forests of Costa Rica, leaving behind the clean conveniences of the digital world

for an organic riot of plant and animal life. As promised, though, he would be back. Ray's passion for the unkempt splendor of the jungle has remained unabated after nearly two decades of intermittent research there, but in the last few years, it's the digital world that has claimed his closest attentions. Since late 1989, Ray has done his most important fieldwork seated in front of a computer, observing the busy fruits of an activity that has come to define his career: he breeds viruses.

Or to put it more precisely, he breeds worms, since that's the stickler's term for software that is both self-reproducing and able to execute its code independent of any host program. Ray, convinced that his programs are as good as alive, calls them simply "organisms," or "creatures." Whatever they are, though, he's been breeding quite a lot of them. He's been breeding them with the full support of his university employers, with the financial backing of major corporations, and with the steadily growing curiosity and respect of fellow researchers in the fields of both biology and computer science. And if all goes according to plan, he will keep on breeding them until he has achieved a goal far more adventurous than anything yet attempted by other virus programmers—infusing the vast unused spaces of the global computer networks with a roiling digital ecology as complex, as fascinating, and ultimately as beneficial to humankind as the rain forests that he has long sought to protect and understand.

In short, by infecting the Net with self-replicating code, Ray aims to turn it into a jungle.

He didn't start out so ambitious. In the beginning there was just a lone drive of a Toshiba laptop to populate, one tiny digital germ to do it with, and a hunch Ray had been kicking around for a decade or so to spur him on. The hunch was that experiments with self-replicating programs (Ray had first heard about them as a Harvard undergrad in the late '70s) might add some theoretical rigor to eco-science's essentially anecdotal attempts at explaining the abstract processes that gave rise to the complex interspecies relationships he had observed in the field. "I was frustrated," he would later tell a group of colleagues, "because I didn't want to study the products of evolution—vines and ants and butterflies. I wanted to study evolution itself."

In this, Ray's attraction to self-reproducing programs differed little from that of Mark Ludwig (who in fact was not unfamiliar with Ray's work by the time he set out to write his magnum opus on computer viruses and evolution). Unlike Ludwig, however, Ray felt neither philosophically obliged nor ethically disposed to work with viruses able to thrive in already existing computer environments. Not that he never considered the option. In fact, his initial plan was to set mutating machine-language organisms loose in a single computer and watch their evolution as they competed against one another for direct access to the computer's core memory, a strategy that might have evolved viruses superbly adapted to any system based on the same instruction set as the original petri chip. But Ray soon scrapped this idea—the risk of accidentally releasing his specimens into the wild seemed too great. Instead, he decided, he would evolve his organisms inside a virtual computer, modeled inside a real one in much the same way some operating systems today can model working emulations of other OSes, allowing DOS programs (for instance) to run in Macintosh environments. The difference, in Ray's scheme, was that his simulated system would be the only environment of its kind; thus, any program that escaped into other computers would find itself a fish out of water, unable to function anywhere but in its birthplace.

While the security benefits of this approach were obvious, its contribution to the scientific effectiveness of the experiment was even more significant: now that Ray was working with an imaginary computer, he was free to shape the system's design to create an environment more hospitable to life. And there was one key change to be made in that regard, for as Ray had come to recognize (and Ludwig would later set down in hard math), today's digital environments simply weren't built with mutant programs in mind. Typical operating systems might let a program randomly move some of its algorithms around with impunity (as the polymorphic viruses do), but at the fine-grained level of individual bit-flipping most closely analogous to genetic variation, even a single chance alteration almost always results in a system-crashing bug. Nature's tolerance of random code revisions

is much greater, and if Ray wanted a more "natural" computer, then one way to get there would be to give it an instruction set in which nearly any sequence of bits would make some kind of sense to the system's virtual CPU.

So he gave it that instruction. He also equipped his phantom computer with a death function, a "Reaper," which would terminate any individual program sooner or later—but would always get to the oldest or most error-prone programs first. Thus primed to carry out the requisite natural selections, Ray's digital ecosphere was nearly complete. He called it *Tierra* (Spanish for "earth") and started preparing the final touch: an inhabitant. Later dubbed "the Ancestor," it was the first worm Tom Ray ever created—an 80-byte-long self-replicating machine written in Tierra's quirky assembly language—and as it happens, it was also the last. Once loosed into the Tierra environment installed on Ray's laptop, the creature's offspring quickly spread to the new world's every corner, within minutes displaying the evolutionary transformations that would "write" Ray's organisms from then on.

A 79-byte variation appeared, rapidly displacing its slightly clunkier predecessors, then smaller descendants followed—a 45-byter, a 51, eventually even a 22—entering a taxonomy that would grow to accommodate hundreds of subspecies as Ray played with Tierra in the months and years to follow. The swift and drastic size reductions of those first runs startled Ray, but even more remarkable were the survival strategies these variants encoded. The 45- and 51-byte creatures, it turned out, were not worms but bona fide parasitic viruses, achieving their leanness by borrowing reproductive code from larger programs when they needed to copy themselves. In turn, host programs acquired an immunity from parasites by failing to register their location in the virtual computer's memory, thus foiling the parasites' attempts to find them.

To the casual student of computer viruses, it's interesting to observe that despite the wide-open and neutral terrain into which the first Tierrans were placed, they swiftly and spontaneously adopted the same techniques built into wild viruses to ensure survival in an environment thick with hostile users and their software: parasitism and stealth. But to the serious scholars of biology who soon began to take note of Ray's work, such developments were more than just interesting. Out of the barest simulation of environmental forces, some of life's more sophisticated interrelationships were emerging entirely unbidden, and while the Mark Ludwigs of the world might object that Ray's initial fine-tuning of Tierran "physics" tainted the experiment, Ray was more than satisfied with its scientific implications. Here, in the unexpectedly colorful diversity bred from a single simple program, was a compelling model of evolution's creative power.

"In my wildest dreams, that was what I wanted," Ray later told author Steven Levy. "I didn't write the Ancestor with the idea that it was going to produce all this."

As much as this bustling ecology-in-a-box thrilled and surprised Ray, however, it soon began to dawn on him that the Ancestor had produced something even more unexpected: high-quality software. Almost all of the Ancestor's progeny displayed some improvement in the efficiency of their code, but in a few cases, evolution seemed to have attained a level of tight-wound optimization difficult for even the most wizardly of human software engineers to achieve, and Ray couldn't help wondering if there was a way to yoke this inhuman skill to the development of practical applications.

It wasn't an unheard-of notion. As long ago as the early '60s, for instance, cutting-edge programmers had begun experimenting with what they called "genetic algorithms"—pools of software subroutines repeatedly multiplied, mutated, and weeded according to how well they performed a given task.

Two decades later, in the same ground-breaking work that established the ability of digital viruses to penetrate nearly any system defenses, computer scientist Fred Cohen also proved that viruses are potentially useful as all-purpose computing devices. As Cohen later put it, "anything a Turing machine can compute, a virus can evolve." Since then, Cohen has tested the proposition

that viruses can create useful code in a number of applications. One notable experiment of his is a network-maintenance ecosystem in which survival of the most needed cleanup tasks ensures maximum efficiency—in which, for instance, self-replicating programs designed to delete unwanted files randomly mutate their file-chasing strategies, with those strategies least wasteful of system resources being spared the Reaper's blade.

But the benefits realized in these experiments were limited, as Ray saw it, by their dependence on artificial rather than natural selection—that is, the software was allowed to evolve only in the direction of a particular function chosen by the programmer. In Tierra, on the other hand, organisms evolved according to criteria that they themselves created collectively, constrained only by the "natural" imperative to reward the thriftiest use of existing resources. Tierra gave evolution a free hand, in other words, and Ray felt certain that the creativity thus unleashed had the potential to tackle software-writing challenges far beyond the reach of human programmers. In particular, the difficulties involved in writing the most productive code for the parallel-processing machines that will take us into of the next century of computing seem to cry out for an evolutionary approach. "We will probably never be able to write such software, as it is way too complex," Ray observes. "Yet we know that evolution can handle that kind of problem."

The reason we know that, of course, is that we—and all other multicellular organisms—are wetware embodiments of frightfully complex parallel processes. But that fact posed a new challenge for Ray. Despite the great variety of digital forms Tierra had generated, it remained an ecology of one-celled organisms, none much larger or much more complicated than the 80-byte Ancestor. In fairness it should be pointed out that the terrestrial biosphere spent its first 3 billion years or so in a similar state before finally exploding into multicellular diversity at the dawn of the Cambrian era (a mere 600 million years ago). Yet if Tierra was ever to prove its full value as a software-writing machine—or indeed as a scientific model of evolution—sooner or later it would have to cough up a Cambrian explosion of its own. And since the key to this burst of complexity seemed to Ray to lie in challenging his evolving creatures with more intricate problems than the simple bit-copying tasks they'd grappled with thus far, he decided that the explosion wouldn't happen nearly soon enough if Tierra remained stuck inside conventional computers, and he began looking into the possibility of installing Tierra on a parallel-processing system.

But then one day in early 1994, Ray had a minor epiphany: "I realized that the global network is just a loosely connected parallel computer, and much larger and more powerful than anything that will ever exist as a single machine."

And thus was born Ray's plan to colonize the Net. He wrote it up soon thereafter in a document plain-spokenly entitled "A Proposal To Create a Network-Wide Biodiversity Reserve for Digital Organisms" (see *Wired* 2.08, page 33), the text of which outlines a vast collective enterprise devoted to hastening the arrival of the digital Cambrian. Ray envisions a Tierran subnetwork spread across thousands of volunteer Net nodes, each of them running the environment as a low-priority background process sustained only by unused (and otherwise wasted) CPU cycles. He is confident that once his "one-celled" simple self-replicating organisms encounter the immensity, the topological intricacy, and the fluid instability of the Net, they will quickly rise to the occasion and evolve into tightly coordinated multicellular conglomerates, thus setting off the dreamed-of Big Bang of complex digi-biotic diversity.

Ray foresees digital naturalists like "modern day tropical biologists exploring our organic jungles. However, occasionally these digital biologists will spot an interesting information process for which they see an application. At this point, some individuals will be captured and brought into laboratories for closer study, and farms for breeding." Harvested, domesticated and then neutered of their self-replicating properties, these prize specimens of code could then be translated from Tierran language into standard programming languages and set to work at any number of tasks. Ray suspects some form of intelligent network agents would be the likeliest first applications to be

culled, but he prefers to emphasize that the most useful products of the digital jungle would be as difficult to predict as rice, pigs, penicillin, and silkworms might have been for an observer of the pre-Cambrian ooze of early carbon-based life.

There's a whiff of science fiction rising from all this, of course, but Ray is hardly indulging in idle speculation. Already a team of computer scientists has gathered under his supervision to work full-time on hammering out the technical details of the plan. He's accustomed by now to dealing with his listeners' occasional anxieties about the prospect of Tierran viral-like pests infiltrating the workaday network environment. "I explain why the things can't escape," he says, "and that quiets the nervous people, but some of them continue to look nervous."

But when the time comes to put their systems where their mouths are, how many site administrators will do so? Not enough, fears Danny Hillis, founder and chief scientist of Thinking Machines Corporation, the former manufacturer of massively parallel computers that had been supporting Ray's work. For all the tricky engineering involved in running Tierra on a Netwide scale, Hillis believes, the greatest challenge facing Ray "turns out to be more of a political issue than a technical issue. People are not necessarily going to want to give up their processing cycles for this"—even if those cycles will otherwise rot on the vine—simply because of a deep-seated reluctance to cede so much as a fragment of administrative control over system resources to a program whose internal processes serve no immediate ends but their own.

But even if computer users ultimately reject the deliberate presence of a global wilderness reserve for computer viruses woven neatly into the fabric of the Net, they may yet fail to keep the computer landscape from turning to jungle. After all, the same personal and subcultural imperatives that drove Hellraiser's career will continue to inspire underground virus writers. And the digital terrain continues to get more interesting. If the Darwinian innovations introduced by Mark Ludwig are any indication of coming trends in viral technique, then it's not inconceivable that a vital ecology might someday flourish in the midst of our daily routines, unplanned, uncontained, ill-comprehended, and irrepressible. It's an unnerving prospect. Yet it wouldn't have to be—not if we prepared for it by actively cultivating a digital biodiversity of the sort Tom Ray proposes. This is a niche that will be filled, whether we fill it deliberately or not.

"We're just going to have to live with them," artificial life researcher Chris Langton says of computer viruses. Our global web of digital systems, he predicts, is fast unfolding towards a degree of complexity rich enough to support a staggering diversity of autonomously evolving programs.

Viruses in a Suit and Tie

But the future of beneficial viruses is not only in the hands of eccentrics such as Hellraiser, Ludwig, or Ray. The good folks at General Magic corporation are eager to put viral code on a firmer and decidedly more lucrative footing. Not that they like to hear it said that they have anything to do with viruses, mind you.

General Magic manufactures a hand-held communication device that relies on a nifty new network-streamlining program language called Telescript. Announced earlier this year with the very visible backing of such info-dollar heavyweights as AT&T, Apple, Sony, and Matsushita, Telescript proposes to do good things. Its intelligent agents, General Magic co-founder Bill Atkinson promises, will soon be flitting about cyberspace on your behalf, visiting remote commercial sites to buy, sell, and trade information for you, and generally behaving themselves with all the decorum you'd expect from a personal digital valet.

Still, despite rather severe restrictions on the agents' ability to replicate, it's hard to deny certain broad similarities between intelligent agents and the offerings of your typical Vx board. Both wild viruses and Telescript agents routinely copy themselves from one computer to another. Both viruses and Telescript agents can run themselves on the computers they travel to, and, for those

same reasons, raise differing degrees of concern about their security. "A virus never does anything good for you, it only does things to you," says hacker legend Bill Atkinson, nervously reaching for a fine semantic distinction between computer wildlife and Telescript's semi-autonomous "intelligent agent" programs.

More intriguing, though, are Telescript's close similarities with Tom Ray's digital diversity reserve and the experiments of Fred Cohen. Cohen, now happily self-exiled from academia and in business for himself as a computer-security guru, is experimenting with a distributed database in which self-reproducing query agents scurry throughout a network, much like the Telescript scheme. And like the sprawling biosphere of global Tierra, Telescript's bustling marketplace depends on a broad base of local interpreter programs installed wherever its agents go to do their business. This has two significant implications. For one thing, the fact that the mobile organisms of both Telescript and Tierra interact only with their interpreters, incapable of functioning in their absence or of bypassing them to directly affect the host environment, obviates many of the security concerns surrounding their autonomy. (Telescript, additionally, makes use of a battery of cryptographically secured restrictions to ensure that its agents don't subvert control of the host machine, either by accident or by malicious design).

And for another thing, the fact that all the interpreters speak the same programming language regardless of the underlying operating system and hardware means that, as the base of interpreters approaches omnipresence on the world's computer networks, the Net approaches the condition of a single, vast, and unmappable supercomputer, with each wandering digital organism a process in one worldwide parallel computation.

Taken together, these two features represent something of a watershed in the history of computing. It has long been observed, rather wistfully, that in principle the world's computers sum up to one gigantic parallel processor, and that the crushing bulk of that metacomputer's CPU cycles goes to waste, unused. Only now, however, with the advent of protocols like Telescript and Tierra, do we have the means to deploy such processes that treat the Net as one machine, safely and sensibly. This, then, is the real significance of these endeavors.

The Dark Side of Benefits

Trying to imagine the marvels that pour forth once you've successfully tapped a computer as elaborate as the Net is as futile as trying to map the future of a society, or of a life—or of life itself.

Of course, trying to foresee the risks that could emerge from that same computer is an equally hopeless task. But as it happens, we are bound to face those risks whether or not we seek to harness the full power of the Net, since the teeming and inevitable population of uncaged digital organisms will in any case plow forward with its own relentless exploration of the Net's capabilities. All we would miss by failing to orchestrate a more manageable viral exploration of our own, therefore, would be the potential benefits—including quite possibly some antidotes to the worst depredations visited on us by the viruses of the wild. And including also, perhaps, something even more precious. For if there is any purpose legible at all in the millennia of human history, it is in the unflagging persistence with which we add to the complexity of the universe. So, if we were to shrink from the chance to actively participate in transforming the Net into the single most complex information entity since the emergence of the human brain, would we not then be shirking a duty of almost cosmic proportions?

It could happen. It's hard to say which is really the more characteristically human trait—our drive toward complexity or our sometimes irrational fear of it. In the matter of computer viruses, fear could well gain the upper hand. It has already shown itself, after all, in our human tendency to overly reduce the multifaceted motivations of the virus writer to a caricature of hooliganism. Likewise it seems to lurk behind the urge to deny that viruses can be anything but lethally dangerous.

But we'd better think long and hard before we let it stand between us and the epic opportunities that globally distributed viral programming presents us with. Because in the end, the meaning of our long-term coexistence with computer viruses may prove difficult to distinguish from the meaning of our own existence.

Bibliography

Cohen, Fred, *It's Alive! The New Breed of Living Computer Programs* (Wiley & Sons, 1994).
Ludwig, Mark, *The Little Black Book of Viruses* (Tucson: American Eagle Publications, 1991).
———. *Computer Viruses, Artificial Life, and Evolution* (Tucson: American Eagle Publications, 1993).
Tom Ray: The Tierra home page can be found at http://www.his.atr.jp/~ray/tierra/.

Published in 2006 by
Routledge
Taylor & Francis Group
270 Madison Avenue
New York, NY 10016

Published in Great Britain by
Routledge
Taylor & Francis Group
2 Park Square
Milton Park, Abingdon
Oxon OX14 4RN

Routledge is an imprint of Taylor & Francis Group

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 0-415-94223-3 (Hardcover) 0-415-94224-1 (Softcover)
International Standard Book Number-13: 978-0-415-94223-2 (Hardcover) 978-0-415-94224-9 (Softcover)
Library of Congress Card Number 2005014486

Library of Congress Cataloging-in-Publication Data

New media, old media : a history and theory reader / edited by Wendy Hui Kyong Chun, Thomas W. Keenan.
p. cm.
Includes bibliographical references and index.
ISBN 0-415-94223-3 (hardback : alk. paper) -- ISBN 0-415-94224-1 (pbk. : alk. paper)
1. Mass media--History. I. Chun, Wendy Hui Kyong, 1969- II. Keenan, Thomas, 1959-

P90.N52 2005
302.23'09--dc22 2005014486

informa
Taylor & Francis Group
is the Academic Division of Informa plc.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the Routledge Web site at
http://www.routledge-ny.com