J Comput Virol (2009) 5:7–8
DOI 10.1007/s11416-008-0114-0

FOREWORD

Foreword

Jürgen Kraus

"Wer in der Zukunft lesen will,
muss in der Vergangenheit blättern"
"Celui qui veut lire dans,
le futur doit feuilleter dans le passé".
André Malraux

Let us take a stroll down memory lane; some twenty-eight years to be exact.

In 1980-the year in which Ronald Reagan had just been elected 40th President of the United States—we find the Cold War still very much ongoing and Java synonymous with an island in the Indian Ocean. Adolescents had not yet had their spatial acuity tested by Rubic's Cube, and you could be fairly certain that chess skills were measured solely against other carbon-based bipeds.

In 1980, most computers were behemoths, so-called mainframes, whose gargantuan dimensions filled entire halls. However, one could discern the first stirrings of change, with home computers like the Atari 400 or the Sinclair ZX80 appearing. What we call our present day Internet was slowly developing from the US DoD's ARPANET project. The poster child of the personal computer era, the IBM PC, was to be launched a mere year later.

In 1980, the University of Dortmund was also about to graduate me in Computer Science; in those days a field still in its infancy.

My studies comprised in roughly equal parts theoretical and practical subjects; though I had always felt especially drawn to theoretical vantage points. You may ask why: On the one hand, the bromide of theory revealing the essence of things by virtue of material detachment rang true to me, on the other hand, I did minor in mathematics.

Though exams were tedious, I needed to hand in a thesis to graduate; this pre-supposed my finding a topic first. Here is where my penchant for theory paid off handsomely: I had been auditing the course "Recursive Functions" in the Winter semester 1978/1979, taught by Prof. Volker Claus, one of Germany's outstanding computer science pioneers. I approached Prof Claus in my search for a thesis topic. His suggestions revolved around computability and recursive functions, which helped crystallize the following question: Is it possible to construct self-reproducing programs, and can we analyze their algorithmic complexity and the function classes these programs realize?

The impetus for this particular line of investigation lay in Prof Claus' visionary ideas: "If one conceives of a computer system (in a feat of futuristic abstraction) as an artificial world, we may search for life-like processes therein. Computer programs may be cast in the role of simple living things as long as they evince defining traits such as mutation, reproduction and metabolism; roughly analogous to their biological counterparts."

In the mind of an euphoric student, a thesis on self-reproducing programs represented no less than the quest for life in a primordial computer soup consisting of ones and zeros. It also obviate the need to fly to Mars to find little green men—and happily, to substitute astronaut food for donuts and to subject oneself to space radiation. After all, the computer system might be harboring creatures just waiting to be discovered by yours truly.

Hence, I adopted the topic. I did this in full knowledge that the existence of self-reproducing programs (also in higher level languages) followed by necessity from Kleene's recursion theorem, but that, in practice, actual construction using

This foreword has been translated from the German by the editors Daniel Bilar and Eric Filiol (corresponding editor: efiliol@wanadoo.fr).

J. Kraus (✉)
Hagen, Germany

said theorem would have to fail. I had to find a way, my own way—that was the challenge.

To cut to the chase, the thesis' goals were met in that self-reproducing programs in higher level programming languages were constructed and their properties investigated. It turned out that self-reproducing properties could be formulated with relative few lines of code and simple algorithmic (loop1) complexity. Of particular importance was the realization that self-reproducing programs could be enhanced by arbitrary—possibly harmful—functions while maintaining their self-reproductive capabilities. Put another way, in addition to their harmful functionality, malicious programs could have self-reproduction added to their repertoire, thereby significantly increasing reach and noxiousness.

Compared to biological organisms, however, self-reproducing programs lack a metabolism. As such, they can at most be classified as viruses. It seems that little green men are yet more likely to be found on Mars or in some other galaxy.

Thus, already in 1980, self-reproducing programs with noxious capabilities described the notion of a "computer virus", albeit unwittingly and without appreciating the notoriety the subject would achieve in years to come.

After receiving my Master's from the University of Dortmund and starting my corporate career end of 1980, I could therefore not have anticipated running into my old thesis "On Self-Reproducing Programs" over and over again. Quite a few articles on computer viruses referenced my work. As far as I know, the first in-depth discussion appeared in Ralf Burger 's "Das Grosse Computerbuch" (Data Becker 1987). He found it remarkable that on the one hand, it predated Fred Cohen's "Computer Viruses—Theory and Experiments" (University of Southern California 1983) by roughly four years. On the other hand, the treatise had remained virtually unknown to the general public. Though not missing or lost, it was only accessible via the University (today Technical University) of Dortmund.

This relative obscurity has come to an end, in part thanks to Daniel Bilar, a promising young scientist at the University of New Orleans. He had heard about the thesis in the course of his investigating metamorphic computer viruses. He queried for and obtained—after some determined efforts—a copy from Dortmund. He then enthusiastically announced to the Dortmund Department of Computer Science his intentions to make an English translation available: His colleague, fellow translator and editor-in-chief Eric Filiol would publish it in the leading specialized academic computer virology journal, the "Journal in Computer Virology" (Springer Paris). With pleasure, I agreed to this endeavor. To have the treatise "On Self-Reproducing Programs" republished, after almost thirty years, is after all a very special honor.

Hagen, November 2008 Jürgen Kraus