All Things Being Equal?

Stan Kelly-Bootle, Author

New year,

ANOTHER

PERSPECTIVE

By the time these belles-lettres reach you, a brand new year will be upon us. "Another Year! Another Mighty Blow!" as Tennyson thundered. Or as Humphrey Lyttelton (q.g.)[1] might say, "The odious odometer of Time has clicked up another ratchette of entropic torture." Less fancifully, as well as trying hard not to write 2007 on our checks, many of us will take the opportunity to reflect on all the daft things we did last year and resolve not to do them no more.[2] Not to mention all the nice things we failed to do. I have in mind the times when I missed an essential semicolon, balanced by the occasions when inserting a spurious one was equally calamitous. Surely any half-decent computer language should know where my statements are meant to terminate, and then properly redistribute the punctuation provided? The smarter Lisps became good at DWIM (do what I mean), balancing those damned, spurious parentheses. But I digress, having planted a topic known to incite reader feedback.[3]

As one of our Anglican General Confessions humbly confesses, "We have left undone those things which we ought to have done. And we have done those things which we ought not to have done." One is never sure if all denominations still use the exact wording acquired from youthful, schooled repetition of Daily Morning service. That sonorous balance of sinful omission and commission has probably been diluted in some LaxLib parishes: "O Lord, considering my genes, nasty parents, and an inadequate educational system, I haven't done *that* badly." That's not quite as cynical as you might think. I find that what we used to confess as "There is no *good* in us" has been changed in some services as "There is no *health* in us."

PREVAILING SINS

The mal in malware is the same evil mal we find in the Order of the Garter: Honi soit qui mal y pense. The rather antiquated French motto means: Shame on you for thinking evil of it. Here we can take *it* as the unspecified *it*, the third-person inanimate pronoun, or, why not, giggle again at our abbreviation for information technology. What the original *it* was that could induce shame is best left to (nudge-nudge) speculations on the role of garters in dysfunctional royal families.

More germane to *Queue* readers is the ongoing evil of malware. In particular, I address the plight of my old friend, Professor George Ledin Jr. of SSU (Sonoma State University), where he has been teaching computer security for more than 30 years. At SSU, CS operates as part of its ES (engineering science) department, indicating a general bias toward the hands-on, lab-centered, hands-dirty, practical aspects of our fair trade. Ledin has recently attracted much flak (and some praise) for teaching a course on Malware (which I now elevate as an uppercase domain for study, even if the term *discipline* seems inappropriate).

Reversing the (in)famous template "X Considered Harmful,"[4] Ledin wrote an editorial entitled "Not Teaching Viruses and Worms Harmful" for *Communications of the ACM* (January 2005). Peter Neumann's essential Inside Risks Web site carries links to this and related columns.[5]

Ledin's plea for more openness in discussing the structure and internal workings of Malware did not arouse much ACM feedback. The press sniffed controversy, however, when Ledin introduced a specific course devoted to Malware in the spring 2007 semester, sending reporters and photographers to interview him and his students.[6] Students were shown how to write malicious code that evaded current antiviral products.

There seem to me to be two main, equally rational arguments tugging me in opposite directions. Pro-Ledin, his analogy with biology is persuasive. In med schools and labs, we study, teach, and even manufacture (gene manipulation, etc.) diverse life-threatening viruses in the worthy fight to understand nature's ever-evolving biohazards and develop ever-changing defenses. That knowledge and the germs it can produce are undoubtedly dangerous, nay apocalyptic, in the wrong hands. Yet, in spite of 100 percent security being unattainable, nobody suggests that teaching viral biology and manipulation is harmful. Given similar security, using computers isolated from outside networks, Ledin claims that leakage of viral code

Continued on page 54

Continued from page 56
can be controlled as effectively as for biological hazards.

Contra-Ledin, he underestimates the dangers of Malware by stressing that our current attacks are relatively minor annoyances with hints that the anti-Malware industry is overly fond of scaremongering for commercial gain. Indeed, Ledin reports that three antivirus companies have threatened not to hire any SSU graduates if he continues to teach virus construction. His students have certainly uncovered many weaknesses in the available anti-Malware products by creating new viruses that are undetected by current defenses, and by creating harmless code that triggers false positives. Yet, to be fair, new mutations are arriving daily in the real world outside SSU's closed box, and it always takes time, as Ledin acknowledges, for mutations and brand new attacks to be detected, analyzed, and, to use the technical jargon, for the bastids to be nailed.

With the devilish cunning of the malefactors, and the axiom that a code's *real intentions* are noncomputable (Discuss!), one must always expect some level of false positives, to be balanced against the greater risk of nondetection (false negatives?). The point here seems to be that certain headers, for example, are wrongly identified as viral *signatures* from purely statistical inference when the code body may itself be benign. One would need to let the SSU test viruses out of the box, as it were, to test the efficiency of the antiviral industry under normal conditions. I can side with Ledin here by suggesting that the antiviral companies could, in fact, benefit by recruiting those SSU graduates who have mastered the subtleties of Malware from Ledin's course.

The challenge is the ancient "need-to-know" problem facing many authorities in our wicked post-9/11 and -7/7 worlds. Whether Ledin's students or AVG's staff are taught virus-DIY, we cannot guarantee that the knowledge is used for niceness and not naughtiness (as Maxwell Smart used to say). In the history of bank-safe builders versus bank-lock breakers, the availability of detailed blueprints was carefully controlled. Then, as now, breaches of security often came from inside jobs, the weakest link in the chain being human greed. Quite apart from the minor, banal Malware annoyances cited by Ledin, there is a huge profitable side to modern hacking based on identity theft and other skullduggeries. Whom to trust when Malware tricks can command such tempting, outsourced rewards?

Then we must mention real cases of political and military intrusions and attacks on vital information infrastructure. It's not farfetched to rate such dangers on a par with bomb-belted martyrs. Most Western governments have passed laws banning or limiting the publication of instructions for the making of various weapons of terror. In some places these gags on what was traditionally subsumed under the Freedom of Scientific Data Exchange have been extended to limit books and sermons merely (merely?) inciting others to evil thoughts and violence. Policing these laws in a globally Webbed cosmos is far from easy, yet they can be defended without undue paranoia by those who have suffered. L'Affaire Ledin poses these intractable dichotomies: freedom limited in the fight to defend a greater freedom.

Less dramatically, though, the vast majority of computer users can never master nor care to master the dark

corners of Malware. They are forced to rely on the never-perfect defenses built by extremely bright programmers who, hopefully, are chosen for their incorruptibility and paid accordingly!

IS CODEWORD COMPUTABLE?

The London Times has added Codeword to its repertoire of crosswords and sudoku. I still tackle the latter daily but find Codeword a refreshing change. It is less boring than sudoku insofar as it is clearly less computable, relying on your knowledge of natural language aided (occasionally hindered) by some cryptographic know-how. The 13x13 grid looks like a crossword puzzle with black squares arranged symmetrically, leaving white rows and columns where you insert the usual down-across areas with interlocking letters to form English (in the *Times* version) words. But there are no traditional crossword clues. Rather, each white square bears a number in the range 1 to 26. What you are after is the hidden mapping of each number to a unique letter A to Z. A certain number of squares are helpfully prefilled with letters, so that usually you start by knowing, say, three of the number-letter

assignments. Below the 13x13 grid is 2x13 grid numbered 1 through 26. As you deduce the mappings for each number, you can write them in the bottom grid, and then, of course, build up the letters in the main 13x13 grid. Knowing the frequency of the English letters and letter-pairs is naturally useful, but cunning Codeword setters know them too and can plant surprises such as SYZYGY and other Scrabble favorites. The setters do guarantee that each letter A to Z is used at least once. Or, rather, they guarantee that the whole alphabet can be mapped, meaning that at most one letter may be unused, being then uniquely determined when you get the 25 that have been used.

As an example, Codeword 117 offered a seven-square sequence that read 22 | 25 | 7=W | 9=D | 2=E | 26 | 1 |. We immediately fill in any 7, 9, or 2 squares with the letters W, D, and E. Next we see if ??WDE?? matches any legal English word. Not quite a Unix regular expression since each of the ?s must match a different letter! I chose POWDERS, but later had to backtrack when the assignment 1=S proved impossible elsewhere. It turned out that 1=Y, yielding POWDERY.

I showed the problem to Bob Toxen (of Linux Security fame) who was passing through London (en route Atlanta-Tel Aviv to be unnecessarily precise). He agreed that given a dictionary of allowed words, a solving algorithm certainly exists, but the best time-space-conserving strategy needs considerable care. As with newspaper sudokus, we assume that at least one solution exists: the one devised by the setter. Whether it is unique is a separate, vexing question. Reader feedback predictably solicited.

Finally, to answer the rhetoric in my title. All things are provably never equal. You throw them all, each wrapped in anonymous leftover gift paper, into a bag labeled Universal Class. Then you have to face that bag-thing itself that surely belongs in the bag? Happy New Year! Q

REFERENCES

1. I introduce q.g. (*quod google*) to supplement, if not replace, the archaic q.v. (*quod vide*). Recall that *vide*, pronounced veeday, is the second person singular imperative of *videre* (to see). So *quod vide* is really a bossy cross-reference command to "look it up, or woe betides!" In the same way, mandating that *googlere* (to google) is a regular second conjugation verb, *quod google* is to be pronounced kwad googlay, and to be ignored at your peril. In this instance, your search for Lyttelton, or Humph to his many fans, will be well rewarded.
2. Misinformed prescriptionist pop-grammarians continue to attack the double negative as ignorant and barbaric. They wrongly assume that natural language must always follow the logic of Boolean algebra, where not-true means false and not-not-true means true. In fact, piling on the negatives is an idiomatic survival of earlier standards, to be taken as simple emphasis and reinforcement as in the famous triple-negative of Chaucer: *He nevere yet no vileynye ne sayde*. One is reminded here of the French *ne-pas* and *ne-jamais* constructions, blessed by custom and l'academie.
3. An ancient, shameless ploy of lonely columnists. My own surefire triggers over the years include any mention of GOTO or APL. More recently, sudoku has proved a hot button. When A. L. (Bert) Lloyd edited *Picture Post*, he could always drum up letters from blistering Berkshire Brigadiers by planting debates on whether dogs should be allowed to attend church.
4. Edsger Dijkstra's 1968 letter titled "Go To Statement Considered Harmful" is usually taken as the original seed for this template. To which Donald Knuth replied, effectively saying, "Well, not really harmful, in fact damned useful if you take structured care" (Structured Programming with Go To Statements, *ACM Computing Surveys*, 1974). The Bible seems to support Dijkstra: Go to, let us go down and confound their language (Jehovah at Babel, Genesis 11:7).
5. Inside Risks; http://www.csl.sri.com/users/neumann/insiderisks05.html.
6. News summary by Jean Wasp, SSU's media relations coordinator; http://www.sonoma.edu/pubs/newsrelease/archives/001090.html.

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

STAN KELLY-BOOTLE, born in Liverpool, England, read pure mathematics at Cambridge in the 1950s before tackling the impurities of computer science on the pioneering EDSAC I. His many books include *The Devil's DP Dictionary* (McGraw-Hill, 1981), *Understanding Unix* (Sybex, 1994), and the recent e-book *Computer Language—The Stan Kelly-Bootle Reader*. *Software Development Magazine* has named him as the first recipient of the new annual Stan Kelly-Bootle Eclectech Award for his "lifetime achievements in technology and letters." Neither Nobel nor Turing achieved such prized eponymous recognition. Under his nom-de-folk, Stan Kelly, he has enjoyed a parallel career as a singer and songwriter. He can be reached at curmudgeon@acmqueue.com.