by Bryan Kocher, acm President

president's letter

A Hygiene Lesson

On November 2nd, 1988, an electronic epidemic was started that infected many of the UNIX computers attached to Internet. There are two interesting aspects to this epidemic. One is that the attacking "virus" was non-destructive; it did not destroy files or processes in progress. The other is that the alleged perpetrator of the epidemic is the son of the chief scientist at the National Computer Security Center of the National Security Agency (NSA). I believe that after many years of fruitless admonitions by the NSA, a way has finally been found to focus serious attention on systems security, i.e., hygiene.

The germ causing this epidemic was quite different from the "viruses" previously encountered in the PC world. The PC "viruses" have two common traits. First, to serve their purpose they must be malicious. PC "viruses" are electronic pranks. The originator of the prank wants the victims to know they've been had. Many PC users, however, aren't too sophisticated. They might not notice that their machine is running slow, or the disk is always full, etc., so the prankster does something that anyone would recognize as abnormal. The "virus" erases all their files! Just to make sure that the prank is noticed, the "virus" usually puts a message on the screen explaining what just happened. Second, the victim must do things to help the "virus" spread. The victim gets the virus by downloading software not certified to be safe from an electronic bulletin board or by exchange with another victim. The parallels between contracting a PC "virus" and a sexually transmitted disease are painfully obvious.

The November UNIX epidemic was different from the PC "viruses." It did not damage any of the hundreds of machines infected. It did nothing to announce its presence. Obviously, the perpetrator assumed that the infected systems' owners would realize they had been pranked. More importantly, the prankster apparently wanted the Internet community to realize that truly dangerous infections would not announce their presence. Most of the Internet systems are part of professionally managed systems installations. A PC-like "virus" could only destroy data created since the last system backup. At most installations, that means one day's to one week's work could be lost. That is not a big loss compared to the PC users who almost never make backups and would lose everything. A destructive prank wouldn't be catastrophic on Internet.

Internet, however, does contain lots of data that the government would like to label as "unclassified but sensitive." A really destructive "virus" would spread itself slowly and quietly throughout Internet, collecting and collating data from the entire network until worthwhile intelligence materials were developed. This would be an automated version of the "Wily Hacker" exposed in the May issue of this magazine. In fact, **there is no assurance that such an electronic "mole" is not already in place.**

Potential invaders of UNIX networks must be heartened to note how easily and frequently security can be breached through Internet. The Lawrence Berkeley Laboratory was vulnerable to the Wily Hacker until mid-1987, yet the Lab's organizational cousin, Lawrence Livermore, a nuclear weapons facility, admitted to ten invasions in one recent week. It seems that the Wily Hacker episode has not convinced many people to strengthen their security sufficiently to preclude successful viral attacks.

The UNIX epidemic is like any other epidemic disease. It won't go away until the conditions that allow it to flourish are changed to prevent further infection. Cholera is a classic example of epidemic disease. First identified in Calcutta, India in 1817, it reached Britain in 1829 and killed over 22,000 people within two years. Hundreds of thousands died over the next 30 years. Once germ theory was understood and the contamination of drinking water by sewage shown to be the cause of cholera, the epidemic could be controlled. The city of London constructed 1,300 miles of sewers (built by hand with 318 million bricks) to carry 420 million gallons of effluent per day out to sea. Public health laws were passed requiring that drinking water be piped from certified safe sources. Other public health legislation has been added over the years and Britain has become safe from most epidemic diseases.

Just as in human society, hygiene is critical to preventing the spread of disease in computer systems. Preventing disease requires setting and maintaining high standards of sanitation throughout society, from simple personal precautions (like wash-

(continued on p. 6)

Davis quotes literature about Petri Nets that was published between 1962 and 1979. Let me recommend to him and the readers of *Communications* the bibliography [1]. It contains 2634 references to books and articles about Petri Nets, 1972 of which were published after 1979.

The Petri Net Newsletter [2] contains a list of 26 Petri Net tools which have been (or are being) programmed by various universities and companies throughout Europe. Some of these tools have been used specifically for the requirements specification of large systems and are commercially available (e.g. a tool offered by my company PSI).

Some of the tools not only allow to check Petri Nets automatically (for syntactic correctness and completeness) but enable the user to analyze nets (or rather: the systems modeled by nets) for properties like liveness ("will the system run indefinitely or may it get stuck?"), boundedness ("is the number of system states finite or infinite?"), reachability ("are certain desirable/undesirable system states reachable/unreachable from a certain initial state?"), T-invariants ("does the system have cyclic behaviors?") and S-invariants ("is the number of certain movable parts of the system constant or are there sources and/or drains which may change the number of such parts?") etc. The grade 0 ("poor") assigned by Davis to Petri Nets with respect to the criterion "Automatic Checking" should be reconsidered after an evaluation of the theoretical and practical tools for the checking and analyzing of Petri Nets.

Most Petri Net tools (e.g. the one offered by PSI) can execute nets much like programs. Thus every system model in the form of such a Petri Net is a prototype of the system and can be executed and tested. The grade 0 assigned to Petri Nets with respect to the criterion "Prototype Generation" should therefore be reconsidered also.

A more subtle point concerns the notions of time, clocks, clockpulses etc. Net theory tries to capture axiomatically the general properties of those things that can be concurrent to each other (they are called "conditions" and "events" in net theory) much like logic tries to capture the general properties of those things that can be true or false (they are called formulas, propositions etc.). In this endeavor concurrency is considered 1) to be a phenomenon of the physical world (and not only as a phenomenon inside certain abstract or concrete machines) and 2) to be more fundamental than the phenomenon of time. To explain the semantics of Petri Nets with "clockpulses" is therefore rather misleading. Purging any references to time, clocks etc. from a sketch of the semantics of nets normally makes it shorter, easier to understand and true to the deeper aims and goals of net theory.

I agree with Davis's personal judgment that Petri Nets deserve a medium grade concerning their "understandability to computer-naive personnel", e.g. the grade 4 (on a scale from 0 to 10). But it has to be added that non-naive computer personnel (e.g. computer scientists) typically have more problems with understanding the ideas underlying net theory. It is harder to unlearn basic concepts (like e.g. global time and global states) than it is to learn new ones (like causal dependency and partially ordered local states). On the other hand: to grasp a new basic concept and to apply it in practice is a challenging and exciting activity.

REFERENCES

[1] S. Drees, D. Gomm, H. Plünecke, W. Reisig, and R. Walter. "Bibliography of Petri Nets 1988". Arbeitspapiere der GMD Nr. 315.

[2] Frits Feldbrugge. "A list of 26 Petri Net Tools". Petri Net Newsletter Nr. 22, October 1985.

Copies of [1] and [2] may be requested by writing to: Petri Net Newsletter, GMD-F1, Post Box 1240, D-5205 St. Augustin, Fed. Rep. of Germany.

Ulrich Grude
PSI
Kurfürstendamm 67
1000 Berlin 15
Fed. Rep. of Germany

President's Letter *(continued from p. 3)*

ing your hands or not letting anyone know your password), to large investments (like water and sewage treatment plants or reliably tested and certified secure systems).

Standalone systems, like hermits, almost never get sick. They never come in contact with germs that they haven't already beaten. However, if we are to become a networked society, we must treat computer diseases as a real threat to that society. We must heed the public health warnings from NSA, practice personal systems hygiene, adhere to sanitary standards, and support the development of secure systems to keep the germs out. Electronic epidemics should be like cholera epidemics—something you only read about in history books.