USENIX notes

USENIX MEMBER BENEFITS

Members of the USENIX Association receive the following benefits:

FREE SUBSCRIPTION to *;login:*, the Association's magazine, published six times a year, featuring technical articles, system administration articles, tips and techniques, practical columns on such topics as security, Perl, Java, and operating systems, book reviews, and summaries of sessions at USENIX conferences.

ACCESS TO ;LOGIN: online from October 1997 to this month: www.usenix.org/publications/login/.

ACCESS TO PAPERS from USENIX conferences online: www.usenix.org/publications/ library/proceedings/

THE RIGHT TO VOTE on matters affecting the Association, its bylaws, and election of its directors and officers.

DISCOUNTS on registration fees for all USENIX conferences.

DISCOUNTS on the purchase of proceedings and CD-ROMs from USENIX conferences.

SPECIAL DISCOUNTS on a variety of products, books, software, and periodicals. For details, see www.usenix.org/membership /specialdisc.html.

FOR MORE INFORMATION regarding membership or benefits, please see www.usenix.org/membership/ or contact office@usenix.org. Phone: 510-528-8649

USENIX BOARD OF DIRECTORS

Communicate directly with the USENIX Board of Directors by writing to board@usenix.org.

PRESIDENT
Michael B. Jones,
mike@usenix.org

VICE PRESIDENT
Clem Cole,
clem@usenix.org

SECRETARY
Alva Couch,
alva@usenix.org

TREASURER
Theodore Ts'o,
ted@usenix.org

DIRECTORS
Matt Blaze,
matt@usenix.org

Rémy Evard,
remy@usenix.org

Niels Provos,
niels@usenix.org

Margo Seltzer,
margo@usenix.org

CREATING THE EVT WORKSHOP

DAN WALLACH
Rice University,
EVT '06 co-chair

We only just completed the second annual USENIX/ACCURATE Electronic Voting Technology Workshop alongside the USENIX Security Symposium, and it's already time to start writing the history of the workshop. My, how time flies.

EVT began when two good ideas collided. Alva Couch, working with the USENIX Board of Directors, wanted USENIX to have a voting workshop, which he originally titled "Verifiable Electronic Voting." This idea found its way to Avi Rubin, who had been working on the issue and was a former USENIX board member. Now, Avi, myself, and a number of other researchers had found out that we were about to be awarded a big grant from the NSF to study electronic voting security. Of course, it was still secret and we couldn't tell anybody until the NSF made the big announcement. So we stalled, knowing full well that one of the things we promised to the NSF was to set up a public workshop on electronic voting. Finally, a few weeks later, NSF made the announcement and we got the workshop rolling.

For readers who have never worked with the USENIX staff, it's hard to appreciate how wonderful it can be to organize a USENIX conference. Ellie Young and her staff really do all the dirty work. As the first chair of EVT, my only responsibilities were managing the program committee and ultimately delivering a list of accepted papers. I didn't have to worry about renting space. I didn't have to worry about catering or registration fees or any of that stuff. (Have I mentioned that the USENIX staff rock?)

So, out went the call for papers, in came the submissions, and off to the races went the reviewers on a compressed timetable. We decided to do away with paper proceedings so we could go cheaper and faster (ah, the delicious irony). In the end, attendees got CD-ROMs and everybody can find the papers online. This year, we had the second annual EVT, and with the luxury of advance publicity, we got many more submissions than we could possibly accept, leading some people to say that we already need *two* voting workshops. And so now I'm working on creating yet another workshop. Details to be announced (and history to be written).

If you have an idea for a new workshop USENIX might be interested in sponsoring, please contact workshopproposals@usenix.org.

2008 USENIX NOMINATING COMMITTEE

ELLIE YOUNG

The biennial elections of the USENIX Board of Directors will be held in early 2008. The USENIX Board has appointed Mike Jones to serve as chair of the Nominating Committee. The composition of this committee and instructions on how to nominate individuals are sent to USENIX members electronically and published on the USENIX Web site.

LETTERS TO THE EDITOR

To the editor:

I enjoyed the article, "Some Lesser-Known Laws of Computer Science" (August 2007), and in that spirit wanted to share my three laws of system design (a term that, for me, includes both hardware and software). I have always been guided by them, but only wrote them down in 1998 when I found that I was repeating them too often to junior programmers who lost sight of why they were doing what they were doing in the first place. (I had thought they were too obvious to require stating, but I guess I was wrong.)

So, without further ado, here are Chessin's three laws of system design (with apologies to the estate of Isaac Asimov):

- First Law: Make life easy for the ultimate end user of our systems.
- Second Law: Make life easy for the ISV, except where to do so conflicts with the First Law.
- Third Law: Make life easy for ourselves, except where to do so conflicts with the First or Second Laws.

—*Steve Chessin*
(steve.chessin@sun.com)

To the editor:

Just read the August 2007 issue, with your "Musings." It seems that *;login:* (and many others) is publishing numerous articles of which the gist is basically "beware."

So, how does a suitably paranoid individual who still wants to get some computing done actually get it done? What are the defensive measures, the procedures, the "safe" software, etc?

I've been doing some on-again off-again research on this topic for a while, and there doesn't seem to be a comprehensive answer. Linux Live CDs are a defense against a large set of attacks, but data continuity (saving you work for tomorrow) is a problem. There are still lots of hazards at the physical and logical layers.

Is there a small set of resources out there that could help the individual computing at home?

—*John Lloyd*
(jal@mdacorporation.com)

John:

The most dangerous activities for home users today are browsing the Web and reading email. I'm not kidding.

The best countermeasure for Web browsing attacks is to segregate your critical use of the Web by visiting key sites (banking, etrading) only from a browser running in a VM. I do this. I even use a Live CD image (Linux). No history, and that is a bother but also a security measure.

As for email, what I do—run a very primitive mail client that cannot execute anything—doesn't work for most people. The problem with email clients is that they will automatically invoke HTML-rendering libraries, and this makes email into a real vulnerability. What other suitably paranoid friends do is use Mac OS X to read their email. I just got back from USENIX Security '07, and noted many people doing this, asked Bill Cheswick just how safe he thought this was, etc. When Leopard comes out, with some real security upgrades, I may do the same thing. But not yet.

—*Rik Farrow*

To the editor:

Reading your August '07 editorial brought a few things to mind that might be of interest to you. First, on labeling URLs as harmful, I recently stumbled across the Netcraft toolbar, http://toolbar.netcraft.com/. The idea is to use the sort of information that Netcraft monitors to rank a Web site's trustworthiness. It also lets them publish Web site

rankings based on those using the toolbar. See, e.g., http://toolbar.netcraft.com/stats/topsites?c=IE#65597.

I also recently saw an article about how ineffective these methods of labeling URLs as suspicious are: http://www.rsa.com/rsalabs/cryptobytes/CryptoBytes-Winter07.pdf.

That issue of *CryptoBytes* also contains an interesting article on how easy it can be to guess mother's maiden names.

Later in the article, you commented on how many users one's system has. You mention Apache as a one-user system. Actually, there are two choices here. One is to run Apache (and all the CGI programs) as one user. The other is to use suEXEC to run CGI as different user, making Apache a multi-user system. There are advantages to both systems, depending on your model.

A long time ago, I realised that having lots of "small" users is a good idea. I was trying to track an offensive email sent, and I discovered that it originated from the "nobody" user on a UNIX system. There were a bundle of small services that ran as nobody, and having them all use the same UID meant that typical UNIX tools didn't help track the problem (i.e., files, process accounting, and syslog messages all remember UIDs/usernames). Since then, I try to run each service (and indeed, each piece of each service) as a different UID. That way, if some "nobody" misbehaves, I have a quick way to find the one that's the problem.

Finally, your comment on security and multiprocessing reminded me of a paper by Robert Watson presented at the USENIX WOOT '07 conference, "Exploiting Concurrency Vulnerabilities in System Call Wrappers," viewable at http://www.usenix.org/events/woot07/tech/. It explains how security systems must be careful not to check data that might be modified after the check but before use. Watson demonstrated that this error can lead to exploits of a number of systems.

—David Malone (dwmalone@maths.tcd.ie)

David:

The point I was making is that most of our systems are, essentially, single-user systems, especially when you consider that most systems are desktops. And even servers, by default, run as single users, with suEXEC being the exception, not the rule.

—Rik Farrow

To the editor:

While I disagree with your review of *Myths of Innovation* (Berkun), in the August '07 issue (I gave the book a negative one elsewhere), that's not the point of this note.

What I'd like to object to is the assigning of the "eureka" moment to Euripides, a Greek tragedian of the 5th century B.C. (?480–?406 B.C.), and depriving Archimedes of Syracuse (~?287–?212 B.C.). As Archimedes has been called "the greatest mathematician of his age," and he did a vast amount of engineering work (see E.T. Bell's *Men of Mathematics* [1937, 4th ed., 1976]), I'm not sure the story, which can be found in Plutarch (~46–127 A.D.), is indeed a myth.

—Peter H. Salus, Ph.D. (peter@netpedant.com)

Peter:

Good point: it's Archimedes who should have been credited with the eureka moment—certainly not a Greek playwright.

However, I did enjoy the book. Having worked at many startups, as well as having listened to lots of people who had great ideas that they expected would naturally be accepted and built upon *for the idea's worth alone*, I thought the book was quite important for those people to read.

—Rik Farrow