Accountability for Perfection

FRED B. SCHNEIDER
Associate Editor in Chief

Perfection is great if you can get it. But most of the time, we must live with less. Computing systems are as good an example as any: they aren't secure, yet we live with them.

Software Producers

We do know how to build computing systems that are more secure than those being fielded today. This prompts critics to suggest that software producers be held accountable for what they build. That suggestion cannot, however, be applied to systems, like the Internet, that evolve by accretion and, therefore, have no identifiable producer to hold accountable. But even ignoring such systems, implicit in proposals to hold some producer accountable is a presumption that we can somehow place the blame.

Centuries of bridge and building failures have fostered the development of forensic analyses for catastrophes involving mechanical and civil engineering artifacts. This is undoubtedly helped by the relative simplicity of such artifacts when compared with computing systems. But there are also other reasons that the "blame game" for engineers of physical systems isn't like that for engineers of computing systems:

- A computing system might fail because a component has failed. This could mean that the component's producer should be held accountable, or it could mean that the system integrator should be held accountable for deploying the component in an environment the producer never intended. In February 1991, a Patriot missile battery was deployed in an environment its designers never anticipated when it was run continuously for 100 hours rather than 14 hours; the accumulated clock error left the system ineffective as an antimissile defense, with 28 dead and 98 injured as a result. Unless software components are accompanied by adequate descriptions (functional specifications as well as assumptions about the deployment environment, such as what threats can be tolerated), we can't assign blame for system failures that can be traced to component failures.
- Alternatively, a computing system might fail even if no component fails but nevertheless there are unacceptable (and surprising) emergent behaviors. A long tradition of such surprises exists in bridge design, including the Tacoma Narrows Bridge in Washington State and the Millennium Bridge in London. Moreover, correct behavior for bridges is generally well understood and relatively simple to state, as compared with correct behavior for nontrivial software systems. And unlike bridges, software typically isn't delivered with a paper trail documenting what the system is supposed to do (and not supposed to do), why the design should work, and what assumptions are being made.

So, to hold software producers accountable, we need a mature discipline of forensics for computing systems and components. But getting there will require some radical changes in software development practices, since in addition to delivering systems, producers will need to deliver specifications and analyses—something that, today, is far beyond the state of the art.

Attackers

Accountability can also serve as a defense, thereby playing a second important role in system security. Rather then deploying defenses that prevent misbehavior, we ensure that each system action can be attributed to some responsible party in the "real" world. With this doctrine of accountability, unacceptable actions aren't prevented but simply attributed, which in turn brings repercussions for the perpetrator—trial, conviction, and penalties. Of course, suitable evidence must be available, and the accuracy of claims being made about accountability is crucial. But getting that right is likely much easier than obtaining perfection for an entire system, as required when defenses involve preventing misbehavior.

Implementing a doctrine of accountability implies an increased emphasis on audit mechanisms. Look at the number of pages in a typical computer security text-

Editorial Board Changes

IEEE Security & Privacy magazine is a volunteer effort, and we couldn't produce the high-quality material we publish in each issue without the talented people who donate their time and energy to the effort. We want to publicly thank some of our faithful editorial board members and department editors who have served the magazine well over the past several years but are now concluding their terms. Members of the editorial board at large are responsible for handling the refereeing process for contributed articles and often act as guest editors for our theme issues. Martin Abadi, Avi Rubin, and Giovanni Vigna have served these functions since the magazine's inception and contributed substantially to our success; we thank them for their years of service and wish them continued success in future ventures.

Department editors take on the added responsibility of producing or recruiting content for a department that runs in each issue, while also fulfilling regular editorial board member responsibilities from time to time. Stepping down from regular department editor duties are Roland Trope and Michael Power (co-editors of the Privacy Interests department), Martin Stytz (co-editor of the Digital Protection department and a previous incarnation of our Book Review department), and Shari Lawrence Pfleeger and Charles Pfleeger (current co-editors of the Book Review department). All of these people have served with distinction. Shari and Charles will remain on our editorial board, and Roland, Michael, and Marty will join our newly created Senior Advisory Board. We want to retain access to their accumulated experience and perspective, and fortunately for us, they've all agreed to serve on the SAB and consult with the editorial board from time to time on matters of importance to the magazine's operation.

We're also pleased to welcome some new volunteers who have recently agreed to join us. Jim Dempsey of the Center for Democracy and Technology and Terry Benzel of the University of Southern California's Information Sciences Institute are new members of the editorial board. Fred Cate of Indiana University and Ben Laurie of Bunker Secure Hosting have agreed to assume the reins of the Privacy Interests department. The Digital Protection department will be replaced by a new Security & Privacy Economics department, under the guidance of Michael Lesk of Rutgers University and Jeffrey MacKie-Mason of the University of Michigan. Patrick McDaniel of Pennsylvania State University will join Sean Smith as a new co-editor for the Secure Systems department. Vijay Varadharajan of Macquarie University has joined O. Sami Saydjari as co-editor of the On the Horizon department. Finally, Marc Donner has agreed to revive his popular BiblioTech department for us a few times a year.

We are indeed blessed with a wealth of new talent and energy. We'll do our best to make it serve your interests, but we can do it much better if you let us know what you like, and what you don't like, in the magazine. Please do take a moment to write us when you can.

—Carl E. Landwehr, Editor in Chief

book devoted to discussing authorization versus what is devoted to audit mechanisms, and it becomes clear that adopting the doctrine of accountability would have far-reaching effects in what we teach as well as how we design systems.

There is, in addition, a tension between accountability and anonymity, so a doctrine of accountability impinges on our societal values, our culture, and our laws. Moreover, accountability in networked systems isn't a property that can be enforced locally. When network traffic crosses international borders, accountability for originating a packet can be preserved only if all countries carrying that traffic cooperate. Some countries will see mandates for cooperation as mandates to cede autonomy, and they will resist. Various cultures resolve tension between anonymity and accountability in different ways, perhaps even selecting different trade-offs for their own traffic than for outsiders' traffic. In short, there's no universal agreement on mandates for accountability.

Beyond system and legal support for accountability, we will need analysis methods that can identify a perpetrator after an offense has occurred. Classical techniques for criminal investigations in the physical world—the fingerprint on the wine glass, the fiber sample from the rug, DNA matching—aren't much use on data packets. Bits are bits, and they don't travel with detritus that can help identify their source, intent, or trajectories. Thus, the relatively new field of computer forensics faces some tough challenges, especially when there's scant system support for accountability, as is the case today.

Accountability, then, could be a plausible alternative to perfection. And while perfection is clearly beyond our capabilities, accountability is not. It's therefore feasible to contemplate an exchange: accountability for perfection. But to support accountability, we must develop computer forensic methods for assigning blame when the transgressor is a system producer or when the transgressor is a system user.

Not coincidentally, this issue of *IEEE Security & Privacy* magazine is devoted to computer forensics. The issue is cosponsored by the IEEE Signal Processing Society and will be seen by all its members. Given the growing importance of computing forensics—both in producing software and in defending it—this issue is unlikely to be our last word. □