Does Security Trump Reliability?

James Bret Michael, *Naval Postgraduate School*

Phillip A. Laplante, *Pennsylvania State University*

Jeffery Payne, *Coveros*

Paul E. Black and Jeffrey M. Voas, *National Institute of Standards and Technology*

A conference panel discussed security and reliability and which of these concerns outweighs the other. Although the panel didn't draw a definitive conclusion, it did open the question for further consideration.

Does security trump reliability, or vice versa? In 2013, the Seventh International Conference on Software Security and Reliability (SERE) in Gaithersburg, Maryland, hosted a panel discussion with representation from industry, academia, and government to answer this question. Neither the panelists nor anyone in the audience took a firm stance. Rather, the discussion meandered from the original question to two conclusions: it depends, and both are equal.

IT DEPENDS

Arguably, the debate over which "ility" trumps the other really hinges on the context in which the system, application, or service operates. For example, processors embedded in a weapon system, along with the software running on those processors, are expected to operate failure-free for a long time period during both peace and combat time, as well as under nominal and degraded modes of operation.

Although security will always be important—consider emissions security and access control in the weapon system example—the reliability of the key requirements-enabling mission effectiveness for the weapon system's operational profiles will lead to some lowering of expectations for security and acceptance of security risk at the expense of reliability. For example, the tradeoff could be to accept less-than-optimal levels of tamperproofing and shielding from jamming.

Similar tradeoffs apply in civilian applications. Online banking users expect their accounts to be protected from unauthorized outside access. It's possible, however, that security functionality can lead to reliability problems from the consumer's perspective when the security measures prevent access to his or her account. Having to enter a PIN multiple times or requiring secondary authentication, for example, can confuse users and lead to aborted transactions, preventing users from completing their banking activity successfully. This is an example of a tradeoff that involves obtaining usable security. As Butler Lampson pointed out, "Security gets in the way of other things you want" (*Comm. ACM*, vol. 52 no. 11, pp. 25-27).

Continuing the online banking example, we could argue that security

should be given more weight. Confidentiality, integrity, and availability are of paramount importance for financial systems. Juxtaposed with the weapon system example, the online banking example makes it obvious that requirements prioritization isn't a matter of all or nothing but of how to prioritize to each grouping of requirements, by function and nonfunction type as well as within each function.

Consider that every software application has a different risk profile based on its function, the operations it performs, the types of data it uses, and even the other applications with which it interacts. Because of this, the *consequence of failure* drives the balance between reliability and security. For example, for a medical device, if low reliability results in loss of life, it's difficult to argue that security is more important. Similarly, if a security breach results in the leakage of proprietary information about the design of a new product, it's easy to argue that system reliability is more important if it could prevent an attacker from succeeding in stealing intellectual property.

Our analysis also depends on how reliability and security are defined: if reliability is the probability of failure-free operation for a specific time period in a specific environment, then we might consider security as a reliability superset because it concerns assurance outside of time and environmental constraints. Conversely, if reliability is simply higher general software quality, then we would probably consider security as a reliability subset.

Finally, optimal security-versus-reliability prioritization depends on market requirements—market regulations, standards, policies, procedures, or even the whims of individual customers can often dictate what assurance is required for a successful market outcome.

A RECENT EXAMPLE: NYTIMES.COM OUTAGES

In August 2013, there were two separate incidents in which the nytimes.com website was unavailable to subscribers of the online news service (J. Vijayan,"New York Times Site Outage Caused by Attack on Domain Registrar, Company Says," *ComputerWorld*, 27 Aug., 2013; www.computerworld.com/s/article/9241952/_i_New_York_Times_i_site_outage_caused_by_attack_on_domain_registrar_company_says). According to *The New York Times*, the unavailability of the site on 14 August was the

---

The consequence of failure drives the balance between reliability and security.

---

result of an upgrade to a Web content distribution system. This was a reliability issue because the system experienced failures almost immediately after the installation of the upgrade. However, the upgrade could also have introduced new security flaws that hackers could exploit.

The nytimes.com site was unavailable again on 27 August, this time due to alleged hacking activity by the Syrian Electronic Army (SEA). In this case, the SEA didn't directly attack the website. Instead, the news media reported that the SEA used a phishing technique to obtain the login credentials of an employee of the Australian-based domain name registrar Melbourne IT. Melbourne IT is the third-party managed Domain Name Server (DNS) provider for *The New York Times*. The SEA redirected Web browser queries to the nytimes.com site to alternative sites, thus denying access.

In the hacking incident, there didn't appear to be a reliability issue. There was no reported design flaw:

the DNS software, the apps being used to access the nytimes.com website, and the website itself apparently were operating failure-free. This was a security issue: the hackers were able to take advantage of the chain of trust between *The New York Times* and its DNS registrar.

During both instances in August, as well as a period in January when the site was unavailable due to hacking activity originating in China, *The New York Times* experienced losses of revenue from online advertisers and likely the goodwill of some of its content subscribers. It really doesn't matter how the unavailability arose: reliability and security are equal in such cases.

How to assure both reliability and security for distributed systems, heterogeneous networks, and systems of systems is still a perplexing engineering problem. For instance, the security and reliability requirements for individual systems can differ from those requirements needed when these same systems must interoperate as a system of systems. For example, the reliability of firewalls in each of several systems might be acceptable, but when those systems are hooked up such that data moves among the separate enclaves via bring-your-own-wireless-device to the workplace, it's challenging to define the perimeters that need defending by the firewalls because the boundaries can be porous and constantly changing.

SECURITY AND RELIABILITY AS EQUALS IN TRUSTED SYSTEMS

Engineers of trusted systems need to set the requirements prioritization correctly, whether engineering for security, reliability, functionality, or otherwise. They also must fill in any gaps for missing, incorrectly specified, inconsistent, and otherwise deficient requirements.

In the development and maintenance of trusted systems, the question isn't what trumps what, but what overall level of assurance can we ascribe to a system in terms of reliability, security, and other concerns? Reliability and security are considered equal concerns; however, it might not be possible to attain the desired assurance level for each for a particular system.

Perfect reliability or security can't be assured, because our techniques, methods, tools, frameworks, and so on continue to evolve, and resources—such as people, funding, and time—are limited.

Perhaps instead of the original question—security or reliability—a better question is how can we do a better job of bringing together reliability and security engineers to build trusted systems?

In a recent editorial, ACM President Vint Cerf pointed out that hackers and defenders tend to look only at software bugs and are primarily concerned with whether the bugs are exploitable vulnerabilities ("Freedom and the Social Contract," *Comm. ACM*, vol. 56, no. 9, 2013, p. 7). Reliability engineers look at system, software, and hardware failures and faults rather than just at software bugs. Reliability engineers can therefore help defenders and ethical hackers identify much broader categories of failures and faults and fix their root causes rather than chasing bugs and exploits. The latter point was made by Felix Lindner seven years ago, but has yet to be realized in a significant way ("Software Security Is Software Reliability," *Comm. ACM*, vol. 49, no. 6, 2006, pp. 57-61).

So, what we're really trying to achieve here is to answer the question how do we engineer all systems to have adequate levels of both security and reliability? The key is adequate—not over—engineering or assuring of a system, application, or service with security or reliability, as well as recognizing the fact that the use of anything, human-made or naturally occurring, isn't risk-free. ∎

James Bret Michael is a professor in the Naval Postgraduate School's Computer Science and Electrical and Computer Engineering departments. Contact him at bmichael@nps.edu.

Phillip A. Laplante is a professor of software engineering at the Pennsylvania State University. Contact him at plaplante@gv.psu.edu.

Jeffery Payne is chief executive officer of Coveros, a computer security consultancy based in Fairfax, Virginia. Contact him at jeff.payne@coveros.com.

Paul E. Black is a computer scientist with the National Institute of Standards and Technology. Contact him at paul.black@nist.gov.

Jeffrey M. Voas, Security column editor, is a computer scientist with the National Institute of Standards and Technology. Contact him at j.voas@ieee.org.

cn **Selected CS articles and columns are available for free at** http://ComputingNow.computer.org.