DOI:10.1145/3000606

Roger R. Schell

Privacy and Security
Cyber Defense Triad for Where Security Matters

Dramatically more trustworthy cyber security is a choice.

In the early days of computers, security was easily provided by physical isolation of machines dedicated to security domains. Today's systems need high-assurance controlled sharing of resources, code, and data across domains in order to build practical systems. Current approaches to cyber security are more focused on saving money or developing elegant technical solutions than on working and protecting lives and property. They largely lack the scientific or engineering rigor needed for a trustworthy system to defend the security of networked computers in three dimensions at the same time: mandatory access control (MAC) policy, protection against subversion, and verifiability—what I call a defense triad.

Fifty years ago the U.S. military recognized subversion[a] as the most serious threat to security. Solutions such as cleared developers and technical development processes were neither scalable nor sustainable for advancing computer technology and growing threats. In a 1972 workshop, I proposed "a compact security 'kernel' of the operating system and supporting hardware—such that an antagonist could provide the remainder of the system without compromising the protection provided." I concluded: "We are confident that from the standpoint of technology there is a good chance for secure shared systems in the next few years. However, from a practical standpoint the security problem will remain as long as manufacturers remain committed to current system architectures, produced without a firm requirement for security. As long as there is support for ad hoc fixes and security packages for these inadequate designs, and as long as the illusory results of penetration teams are accepted as a demonstration of computer system security, proper security will not be a reality."[8]

The security problem will remain as long as manufacturers remain committed to current system architectures, produced without a firm requirement for security.

Current Approaches Aren't Working

Our confidence in "security kernel" technology was well founded, but I never expected decades later to find the same denial of proper security so widespread. Although *Forbes* reports spending on information security reached $75 billion for 2015, our adversaries are still greatly outpacing us. With that large financial incentive for vested interests, resources are mostly devoted to doing more of what we knew didn't work then, and still doesn't.

a As characterized by Anderson, et al.,[2] "*System subversion* involves the hiding of a software or hardware artifice in the system that creates a 'backdoor' known only to the attacker."

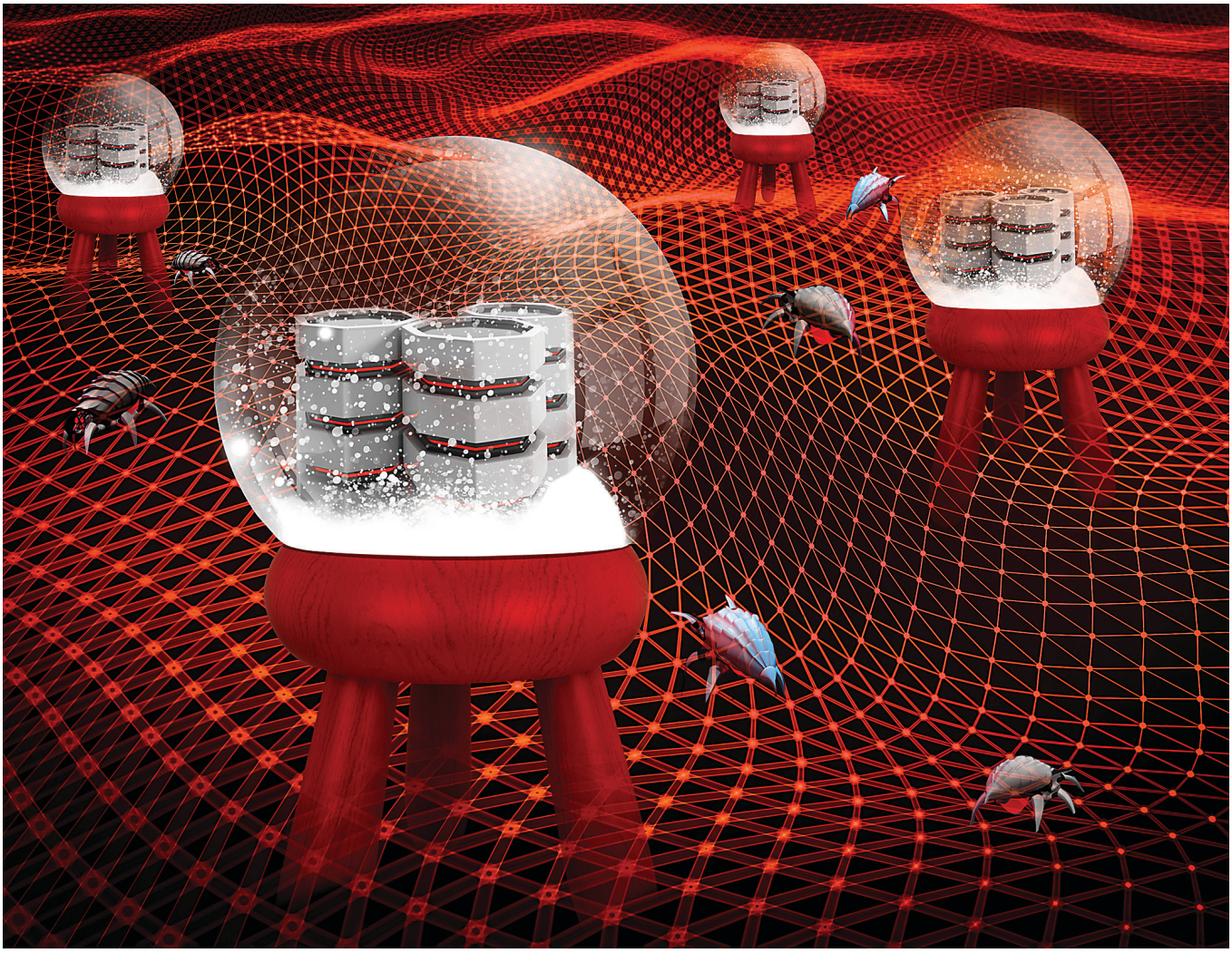
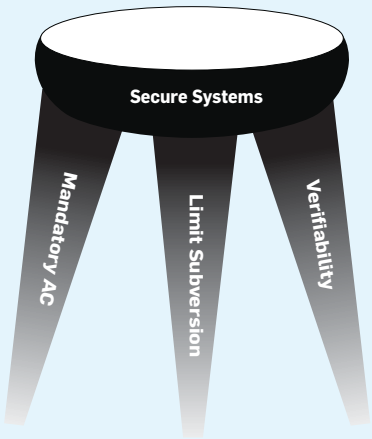


IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

Why does cyber security seem so difficult? Today's emphasis on surveillance and monitoring tries to discover that an adversary has found and exploited a vulnerability to penetrate security and cause damage—or worse, subverted the security mechanism itself. Then that hole is patched. But science tells us trying to make a system secure in this way is effectively non-computable. Even after fixing known flaws, uncountable flaws remain. Recently, Steven Lipner, formerly of Microsoft, wrote a *Communications* Privacy and Security column advocating technical "secure development processes."[6] But, similar to surveillance, "as new classes of vulnerabilities ... are discovered, the process must be updated."

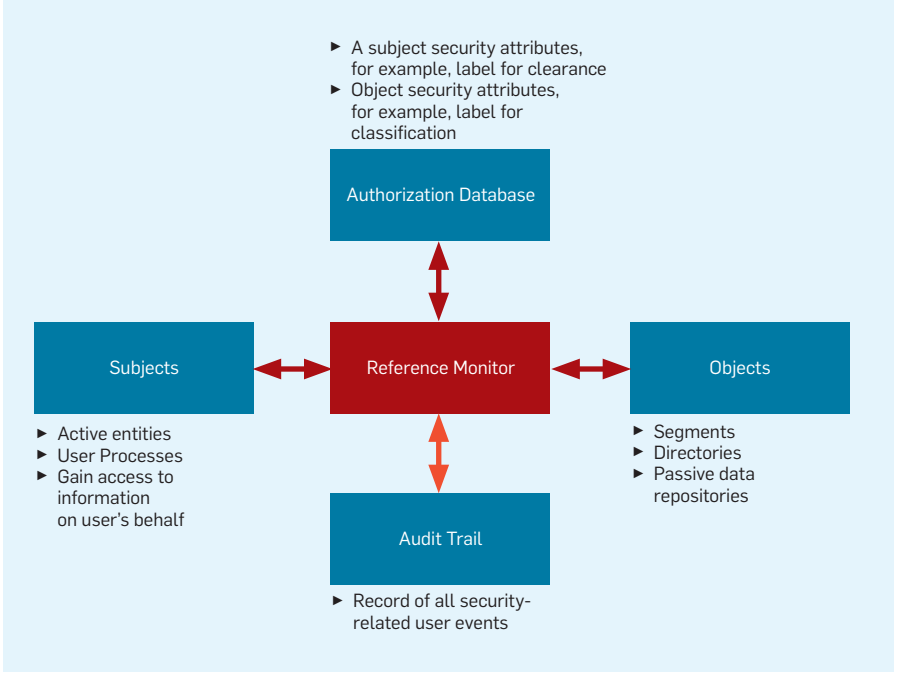
This paradigm has for decades been known as "penetrate and patch." The defender needs to find and patch most (if not all) of the holes, while the adversary only needs to find and exploit one remaining hole. Even worse, a witted adversary has numerous opportunities to subvert or sabotage a computer's protection software itself to introduce insidious new flaws. This is an example of "malware," a preferred attack for many of the most serious breaches. An IBM executive a few years ago described the penetrate-and-patch cycle as "an arms race we cannot win."[5]

Figure 1. Cyber security defense triad.



- Subversion is tool for choice for witted adversary
- Only label-based MAC policy can enforce secure information flow
- Security kernel (reference monitor) is only known verifiable protection technology

Figure 2. Reference monitor.



Cyber Defense Triad for Secure Systems

All three defense triad components are critical for defense of both confidentiality and integrity of information—whether the sensitive information is personally identifiable information, financial transactions (for example, credit cards), industrial control systems in the critical infrastructure, or something else that matters. Although not sufficient for perfect security, all three are practically necessary. These dimensions can be thought of as three strong "legs of a stool," as illustrated in Figure 1.

Security for cyber systems built without a trustworthy operating system (OS) is simply a scientific impossibility. NIST has emphasized, "security dependencies in a system will form a partial ordering ... The partial ordering provides the basis for trustworthiness reasoning."[3] Proven scientific principles of the "Reference Monitor" model enable engineering a verifiably secure OS on which we can build secure cyber systems.

For a Reference Monitor implementation to work, it must ensure three fundamental properties. First, it must validate enforcement of the security policy for every reference to information. Second, it must be tamper-proof, that is, it cannot be subverted. Lastly, it must be verifiable, so we have high assurance it always works correctly. These three fundamental properties are directly reflected in the cyber defense triad.

As illustrated in Figure 2, a Reference Monitor controls access by subjects to information in objects. A security kernel is a proven way to implement a reference monitor in a computer. Whenever a user (or program acting on behalf of a user) attempts to access information in the computer system, the Reference Monitor checks the user's clearance against a label indicating the sensitivity of that class of data. Only authorized users are granted access.

Applying the Cyber Defense Triad

The flawed foundation of current systems is evident in the unending stream of OS security patches that are today considered part of best practices. But we can choose a better alternative. At least a half-dozen security kernel-based operating systems have been produced that ran for years (even decades) in the face of nation-state adversaries without a single reported security patch.[7] These successes were not unexpected. As a 1983 article put it, "the security kernel approach provides controls that are effective against most internal attacks—including some that many designers never consider."[1] That is a fundamentally different result than penetrate and patch.

These systems did not survive long after the end of the Cold War, and much of the "institutional memory" is now lost. But fortunately, some security kernel products were maintained and this original equipment manufacturer (OEM) technology is still commercially available today. And many commodity processors (for example, those that implement the Intel IA32 architecture) still include the hardware segmentation and protection rings essential to efficient security kernels. High assurance of no security patches is truly a paradigm shift. What alternative approach comes close?

Mark Heckman of the University of San Diego and I recently published a paper focused on techniques for applying those Reference Monitor properties, leveraging the fact that "associated systematic security engineering and evaluation methodology was codified as an engineering standard in the Trusted Computer System Evaluation Criteria (TCSEC)"[4] created by NSA. However, the TCSEC didn't include administration and acquisition mandates to actually use this knowledge to create a market in the face of entrenched vested interests. I refer interested readers to our paper for more details on the triad components summarized here.

▸ **Mitigating software subversion.** Several cyber security professionals have concluded that subversion "is the attack of choice for the professional attacker."[2] The primary means for software subversion are Trojan horses and trap doors (commonly called malware). Under the seven well-defined security classes in the TCSEC, only Class A1 systems substantially deal with the problems of subversion.

▸ **Mandatory access control (MAC) policy.** The reference monitor is fundamentally about access control. All access control policies fall into two classes: Discretionary Access Control (DAC) and MAC. Only a label-based MAC policy can, with high assurance, enforce secure information flow. Even in the face of Trojan horses and other forms of malicious software, MAC policies can protect against unauthorized modification of information (integrity), as well as unauthorized disclosure (confidentiality).

Lipner asserts that this reference monitor approach is "not able to cope with systems large enough to be useful."[6] Heckman and I respond that "this quite widely-spread assertion has been repeatedly disproven by counter-examples from both real systems and research prototypes."[4] The paper gives numerous examples of how, by leveraging MAC, complex integrated systems can be composed from logically distinct hardware and software components that may have various degrees of security assurance or no assurance at all.

▸ **Verifiability.** The Reference Monitor implementation defined as a security kernel is the only proven technology for reliably achieving verifiable protection. It does not depend on unproven elegant technical solutions, such as open source for "source code inspection" or "gratuitous formal methods."[2] Security kernels have been shown to be effective for systematic, repeatable, systems-oriented security evaluation of large, distributed, complex systems.

Lipner in his paper[6] asks a critical, but largely unanswered, question: How can customers have any assurance that they are getting a secure system? His answer is limited to development process improvements that don't address fundamentally what it means for a system to be "secure." Heckman, by contrast, details how the Reference Monitor approach, with its strong definition of "secure system," can answer precisely that question.[4]

What Should We Do Then?

It can be expected to take 10–15 years and tens of millions of dollars to build and evaluate a high-assurance security kernel. However, once completed, a general-purpose security kernel is highly reusable for delivering a new secure system in a couple of years. It is economical to use the same kernel in architectures for a wide variety of systems, and the TCSEC's Ratings Maintenance Phase (RAMP) allows the kernel to be re-verified using the latest technology, without the same investment as the original evaluation. Heckman summarizes several real-world examples where, "This is demonstrated by OEM deployments of highly secure systems and products, ranging from enterprise 'cloud technology' to general-purpose database management systems (DBMS) to secure authenticated Internet communications, by applying commercially available security kernel technology."[4] Heckman additionally describes completed research prototypes in the past few years for things like source-code compatible secure Linux and a standards-compliant highly secure Network File Service (NFS).

A first necessary step is to identify where high-assurance security matters for a system. As just one example, several U.S. government leaders have expressed concern that we face an existential cyber security threat to industrial control systems (ICS) in the critical infrastructure, such as the power grid. Use of an integrity MAC security kernel can within a couple of years make our critical infrastructure dramatically more trustworthy. The U.S. government has a unique opportunity to change the cyber security game and should aggressively engage ICS manufacturers by sponsoring prototypes and providing a market using proven commercial security kernel OEM technology. Otherwise, costs may soon be measured in lives instead of bits or dollar signs. C

References

1. Ames Jr, S.R., Gasser, M., and Schell, R.R. Security kernel design and implementation: An introduction. *Computer 16*, 7 (1983), 14–22.
2. Anderson, E.A., Irvine, C.E., and Schell, R.R. Subversion as a threat in information warfare. *J. Inf. Warfare 3* (2004), 51–64.
3. Clark, P., Irvine, C. and Nguyen, T. Design Principles for Security. NIST Special Publication 800-160, September 2016, pp. 207-221; http://csrc.nist.gov/publications/drafts/800-160/sp800_160_final-draft.pdf
4. Heckman, M.R. and Schell, R.R. Using proven reference monitor patterns for security evaluation. *Information 7*, 2 (Apr. 2016); http://dx.doi.org/10.3390/info7020023
5. Higgins, K.J. IBM: The security business 'has no future.' *Information Week Dark Reading*, (4/10/2008); http://www.darkreading.com/ibm-the-security-business-has-no-future/d/d-id/1129423
6. Lipner, S.B. Security assurance. *Commun. ACM 58*, 11 (Nov. 2015), 24–26.
7. Schell, R.R. A University Education Cyber Security Paradigm Shift. Presented at the National Initiative for Cybersecurity Education (NICE), (San Diego, CA, Nov. 2015); https://www.fbcinc.com/e/nice/ncec/presentations/2015/Schell.pdf
8. Schell, R.R., Downey, P.J. and Popek, G.J. Preliminary Notes on the Design of Secure Military Computer Systems. ESD, Air Force Systems Command, Hanscom AFB, MA. [MCI-73-1], Jan 1973; http://csrc.nist.gov/publications/history/sche73.pdf

Roger R. Schell (schellr@ieee.org) is president of Aesec Corporation, and is currently a Distinguished Fellow at the University of San Diego Center for Cyber Security Engineering and Technology. Previously he was a Professor of Engineering Practice at University of Southern California.

The author wishes to thank Michael J. Culver, Mark R. Heckman, and Edwards E. Reed for their valuable feedback on an early draft of this Viewpoint.

Calendar of Events

November 2–4
VRST '16: 22th ACM Symposium on Virtual Reality Software and Technology
Garching bei München, Germany,
Co-Sponsored: ACM/SIG,
Contact: Gudrun J. Klinker,
Email: klinker@in.tum.de

November 6–9
ISS '16: Interactive Surfaces and Spaces Surfaces
Niagara Falls, ON, Canada,
Sponsored: ACM/SIG,
Contact: Mark Hancock,
Email: mshancock@gmail.com

November 6–9
SIGUCCS '16: ACM SIGUCCS Annual Conference
Denver, CO,
Sponsored: ACM/SIG,
Contact: Laurie J. Fox,
Email: fox@geneseo.edu

November 7–10
ICCAD '16: IEEE/ACM International Conference on Computer-Aided Design
Austin, TX,
Co-Sponsored: Other Societies,
Contact: Frank Liu,
Email: frankliu@us.ibm.com

November 12–16
ICMI '16: International Conference on Multimodal Interaction
Tokyo, Japan,
Sponsored: ACM/SIG,
Contact: Yukiko Nakano,
Email: y.nakano@st.seikei.ac.jp

November 13–16
GROUP '16: 2016 ACM Conference on Supporting Groupwork
Sanibel Island, FL,
Sponsored: ACM/SIG,
Contact: Stephan Lukosch,
Email: S.G.Lukosch@tudelft.nl