Invited paper from Symposium on Requirements Engineering for Information Security (SREIS), March 2001, Indianapolis, IN.

Security Requirements Specifications: How and What?

Extended Abstract

John Rushby
Computer Science Laboratory
SRI International
Menlo Park CA 94025 USA
Rushby@csl.sri.com

1 Introduction

Requirements engineering for information security poses two main challenges: eliciting what are the requirements for a particular system, and figuring out how to specify them in a way that is both perspicuous (to the problem owner) and useful (to the developer). In this short note, I look at some of the challenges in *how* to specify security requirements, and at *what* kind of requirements we may expect to encounter in the future.

2 How To Specify Security Requirements?

My prejudice is that specification of security requirements is best undertaken using formal methods. This is because formal methods allow requirements specifications and their refinements to be examined by mechanized calculation (e.g., symbolic execution, model checking, or theorem proving), and this is valuable because it allows *reviews* to be supplemented or replaced by *analyses*. I am using these terms in the sense in which they are employed in the guidelines for software on commercial aircraft [13, Section 6.3]: reviews are processes that depend on human judgment and consensus, while analyses are objective "mechanical" processes such as testing or calculation. Of course, certain questions do require human judgment, and some decisions require consensus, but many other issues are better addressed by analyses than by reviews: analyses are systematic, can be checked by others, and can even be automated. Especially when automated, analyses can be more reliable and thorough than reviews, and cheaper. And they liberate human time and talent for those issues that really do require judgment and consensus (see, e.g., [4]).

There are two main approaches to specifying requirements in a formal manner: one way is to describe a model system that has the characteristics required and then stipulate that an acceptable implementation must be a refinement, in some suitable sense, of that model. The other way is to specify the requirements as constraints that must be satisfied by an acceptable implementation. Security requirements have proven unusually, perhaps uniquely, troublesome to both approaches.

The classical Bell and La Padula formulation of military "multilevel" security is a model-based specification that exhibits the problems of this approach to security requirements specification [2]. First, one has to interpret the "subjects" and "objects" of the model in terms of the elements of the intended application. If some aspects of the system state or behavior are overlooked, then they may provide channels for undesired communication despite verification that the application complies with the model. Channels of this kind were present in the "Multics Interpretation" [3] that was intended to exemplify the utility of the approach [20]. Next, the model proved too restrictive for some aspects of the behavior of real systems, so "trusted subjects" were introduced and allowed to violate the restrictions of the model. The problem with this approach was that there was no overarching security specification to constrain the behavior of the trusted subjects, nor an effective way to calculate properties of the overall behavior of the system [15].

The Bell and La Padula model also illustrates the need for care in interpreting formal demonstrations of consistency in security requirements specifications: one of the most useful ways to examine a formal requirements specification is to check whether it is consistent with some alternative formulation, or entails some expected property. The "Basic Security Theorem" of Bell and La Padula was a demonstration of this kind that was advanced by some (though not by its authors) as evidence that their model captures the "essence" of security [9]. McLean refuted this claim by establishing an essentially identical theorem for a model that clearly violates any reasonable notion of security [11]. (The problem is that the Basic Security Theorem is a consequence of the underlying state machine model and not of the security requirements layered on top of it.)

Difficulties with the Bell and La Padula "model based" approach to security requirements specification were one of the motivations for the development of an alternative "property-based" approach. This was first seen in work that developed the basis for checking multilevel security by information flow analysis [5] and was later generalized as "noninterference" [7]. Noninterference works very nicely for sequential systems and for multilevel security properties, but proved difficult to generalize to distributed systems and to nonhierarchical policies. For distributed systems, the challenge is to find a natural formulation of security that is compositional: that is, one that has the property that if two secure systems are joined together in a suitable way, then the result is also a secure system. Straightforward extensions of noninterference are not compositional, and those that are compositional are not straightforward (early versions, such as "restrictiveness" [10] were very unintuitive) [6]. The problems with nonhierarchical policies are similar: there are many applications where we wish to specify that information may flow from A to B and from B to C, but not directly from A to C. These "intransitive" policies are surprisingly difficult to specify correctly—witness a sequence of papers each identifying flaws in its predecessor's attempt to capture the idea [8], [17], [12], [14].

There is a good reason why these property-based formulations of security requirements have proved difficult: security is not a property! Technically a "property" is a predicate on (or, equivalently, a subset of) the traces of a system.[1] "Liveness" and "safety" properties are important classifications, and it is known that any property can be expressed as the conjunction of a safety and a liveness property [1]. The difficulty with noninterference-like specifications of security is that these are not predicates on traces (thus, they are not properties) they are predicates on *sets* of traces (i.e., they are higher order). This is so because we cannot tell whether a particular trace represents a run in which information flows from A, say, to B, without knowing if there is another trace that looks the same to B, but in which A is absent. I suspect that it is this "nonproperty" character that makes certain formulations of security so tricky to work with. I suspect also that these difficulties are inherent, and not mere artifacts of the technical methods used to specify noninterference.

As evidence for my suspicion, I note that many security requirements are naturally expressed in "counterfactual" terms. Counterfactual statements concern what might have been: "if you hadn't been driving so fast, you would not have crashed." In computer security we use statements like "information flows from A to B if B sees different behavior when A is taken away" and "this certificate authorizes me to do this action because X would not have signed it otherwise." Counterfactuals have long been studied by philosophers, logicians, psychologists and linguists (see the "Counterfactual Research News" web page at `http://www.sfu.ca/counterfactual/`) and it is fair to say that their analysis still poses problems. One issue is that when we consider what might have been we cannot simply subtract out the event we are interested in because that might produce an inconsistent or impossible world (e.g., a world in which you were not born but in which your children are still present) so we have to make some other adjustments—and how are we then to be sure that it is not those adjustments that are the cause of the consequences we wish to investigate. In computer systems we can avoid some of these philosophical difficulties by considering *all possible* worlds in which the event

[1] Several variations on the precise definition are possible, but a trace is essentially a time-ordered sequence giving the history of values passed over the communications channels of the system: an *event* (c, v) corresponds to the value v being sent over channel c; a trace is then a sequence of such events recorded in their order of occurrence (simultaneous events are recorded in some arbitrary order).

of interest did not occur, but we will then be faced with technical difficulties similar to those of noninterference.

Of course, the only restrictions on behaviors that we can actually enforce in a computer system must be properties of some kind (because the enforcement mechanisms must make decisions on the basis of the part of the current run or trace seen so far); I have argued that they are safety properties [16] and Schneider gives a sharper characterization [19]. A subtle nonproperty like noninterference is enforced by some property (e.g., that corresponding to Bell and La Padula) that can be shown to be at least as strong. Since well-behaved properties are compositional, it might seem that we could avoid a lot of difficulty by eschewing the high-falutin nonproperty formulations of security requirements and focusing on the down-to-earth "shadows" that they cast as properties. There are two arguments against this: one philosophical and one practical. The philosophical one is that nonproperties are often closer to the real requirements: we should get these straight and prove that the properties we actually enforce are indeed their stronger "shadows." The practical one concerns analysis of implementations—or "refinements" as they are called in formal contexts. Because security requirements mostly concern what must *not* happen, they are often not preserved under standard notions of (functional) refinement—which demand only that a refinement or implementation does "at least as much" as its specification. There is no restriction on doing "more" and this can allow violation of security requirements (e.g., it does no good to protect files if the implementation allows access to the raw disk). This means that we cannot verify a design with respect to a property "shadow" of some nonproperty security requirement and then simply verify its implementation as a refinement of the design: we need either a "security-preserving" notion of refinement, or we need to directly verify the implementation against its security requirements. I believe that both of these choices are likely to require reference to the fundamental nonproperty requirement.

3 What to Specify?

Multilevel security and the related notion of information flow are considered somewhat passé today—though still important to those who manage sensitive information, and also relevant (under the name "partitioning") to those concerned with fault-containment in safety-critical systems [18]. Recent interest has focussed on the needs of e-commerce, including interesting problems such as copyright protection and contract signing, but what of the future? The collapse of many dot.coms has led some to conclude that e-business and the Internet are also passé, but I share the opinion (which I first heard expressed by Stuart Card of Xerox PARC) that the Internet has been seriously *under*hyped, and that it will continue to shape new requirements for security. My opinion is influenced by projections for the imminent arrival of "ubiquitous computing," which will evolve from a combination of universal communications using radio technology such as Bluetooth and 802.11 for the last few yards, microcell broadband packet radio such as Ricochet (or clunky G3 cellphone technology) for the last mile, and essentially free broadband wireline for long distance, connecting small, cheap, Internet-enabled devices: anything that costs more than 5 dollars will have its own IP address.

The consequence of this ubiquity will be that many interactions that were not previously mediated by computers will become so, and many items that we currently think of as products will become more like services. Here is a trivial scenario: you throw your shirt in the washing machine with a comment that it has a stain. Your shirt button is a computer and knows where you have been recently (it talks with the computers of the places you visit), so it is able to tell the washing machine where you had dinner. The washing machine contacts the restaurant and examines your order. It determines that the stain is probably tomato sauce. It then contacts its manufacturer who downloads the best program and detergent choice for that stain on that shirt (the button knows what type of shirt it is on). This technology may result in slightly cleaner shirts, but it will certainly result also in the recording and mining of much information that previously went unnoticed: each of the computers involved in the scenario will send you a selection of advertisements for restaurants, tomato sauce, detergents, and shirts, and will sell news of your interest in these items to every other computer on the planet.

This scenario may be a little exaggerated, but it illustrates what I think will become dominant security issues in the future: concern for *privacy* and for the *quality* and *integrity* of information that is associated with individuals and their transactions. Most of us are already familiar with supermarket “loyalty” cards that allow the company to record our every purchase and with browser cookies that track our online habits, but how about wireless technologies that can tell where you at any time? Privacy has many facets, but one of them is surely the ability to separate different aspects of our lives (should your employer know how you spend your spare time?) and this is clearly threatened by ubiquitous surveillance. Public concern may then generate a market for selective anonymity, and for anonymity with accountability—concepts whose precise requirements seem quite challenging to capture. Some of the issues in attempting, simultaneously, to satisfy requirements for privacy, accountability, auditability, integrity, accuracy, and so on are vividly illustrated by recent concerns about voting in the USA—see Rebecca Mercuri’s web page at `http://www.notablesoftware.com/evote.html`.

Because most of the information gathered about us is obtained without our knowledge or cooperation, there are few checks on its accuracy or integrity. Anyone who has had a look at their credit report will know how false a picture is created by sloppy, unverified, and incomplete recording. Inaccurate credit reports are a nuisance; how much more serious will be the consequences when *everything* about us is recorded in a similarly sloppy manner? One might hope that truly important information, such as medical history, would be gathered and treated with more care—and in some jurisdictions it may be (e.g., the medical smart card carried by French citizens), but good solutions for decentralized societies seem an interesting challenge. As personal information is recorded more completely and more accurately, so the stakes are raised for traditional security requirements such as authentication: “identity theft” is becoming increasingly common, and is devastating to its victims. Certainly, legal and regulatory action is needed to address these issues, but I believe the information security community should take a lead in articulating requirements and developing mechanisms that place as much control as possible in the hands of the individual citizen. The genuine benefits that could be provided when accurate and complete personal information is available (e.g., automatic detection of multiple drug prescriptions having potentially harmful interactions) will be feasible only if citizens have enough confidence to allow its collection and integration.

Articulating the requirements and developing the mechanisms that allow individuals a measure of control over information collected about them is an urgent, but “reactive,” activity in response to trends that are already underway. A dual, more “proactive” endeavor is to identify opportunities where early focus on security requirements and mechanisms could enable development of new services or improved delivery of services to the benefit of all. To my mind these include almost all activities that involve authentication. Why should I have to show up in person at the consulate (or employ a service to do so) with my physical passport to get a visa? Surely, all that is needed is mutual authentication, consultation with a few databases, and issue of the appropriate authorization—all of which can be done electronically. Australia has exactly such a system (see `http://www.immi.gov.au/eta/eta.htm`) but it depends on information being recorded in Australian computers (which recognize that you are authorized when you present yourself at immigration). Would it not be better if I were given something—a number—that intrinsically but unforgeably indicates that I have received authorization to enter Australia? Similar arrangements could indicate that I am old enough to buy liquor or see a movie, or am licensed to drive a car. Others could show that I have paid the postage on my mail, or have prepaid for a certain number of miles of travel. It is not hard to devise mechanisms to achieve all of these, and some have already been implemented, but the recent demise of E-Stamps Corporation shows that security must be combined with adequate convenience and benefit for customers. So the challenge is essentially one of requirements engineering—in particular, the integration of security with other requirements.

4 Conclusion

The reliability of many essential services—telephone, electric power, mail—used to be ensured by regulated monopolies. For whatever reasons, these monopolies have been dismantled (at least, in the USA, and the forces of globalization are likely to spread the trend) and we (particularly in California) have seen a significant reduction in reliability. Individuals can, to some extent, protect themselves against this unreliability by means within their own control (e.g., subscribing to two cellphone services, buying candles or a generator). The same trend towards decentralization complicates regulation and control of the security, privacy, and integrity of information—at the same time as the quantity and detail of information gathered is set to explode. Individuals have much less ability to compensate for inadequate security than for inadequate reliability in the services that they use. I believe that one of the most interesting challenges in requirements engineering for security will be to articulate requirements for privacy, the individual's ability to monitor and control information about them, and the combination of privacy with accountability in a decentralized, deregulated environment. The counterfactual character of many security requirements seems to be a source of difficulty in their formal specification, so that the technical challenges in this field seem to present interesting research opportunities for many years to come.

References

[1] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, October 1985.

[2] D. E. Bell and L. J. La Padula. Secure computer systems: A mathematical model. Technical Report MTR-2547 Vol. II, Mitre Corporation, Bedford, MA, May 1973. Reprinted in *Journal of Computer Security*, 4(2,3), pp. 239–263, 1996.

[3] D. E. Bell and L. J. La Padula. Secure computer system: Unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, Mitre Corporation, Bedford, MA, March 1976.

[4] Judith Crow and Ben L. Di Vito. Formalizing Space Shuttle software requirements: Four case studies. *ACM Transactions on Software Engineering and Methodology*, 7(3):296–332, July 1998.

[5] R. J. Feiertag, K. N. Levitt, and L. Robinson. Proving multilevel security of a system design. In *Sixth ACM Symposium on Operating System Principles*, pages 57–65, November 1977.

[6] Riccardo Focardi and Roberto Gorrieri. A classification of security properties for process algebras. *Journal of Computer Security*, 3(1):5–33, 1994.

[7] J. A. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of the Symposium on Security and Privacy*, pages 11–20, IEEE Computer Society, Oakland, CA, April 1982.

[8] J. Haigh and W. Young. Extending the noninterference model of MLS for SAT. In *Proceedings of the Symposium on Security and Privacy*, pages 232–239, IEEE Computer Society, Oakland, CA, April 1986.

[9] S. B. Lipner (Moderator). Panel session: Bell/La Padula and alternative models of security. In *Proceedings of the Symposium on Security and Privacy*, IEEE Computer Society, Oakland, CA, April 1983.

[10] Daryl McCullough. Specifications for multi-level security and a hook-up property. In *Proceedings of the Symposium on Security and Privacy*, pages 161–166, IEEE Computer Society, Oakland, CA, April 1987.

[11] John McLean. A comment on the "basic security theorem" of Bell and La Padula. *Information Processing Letters*, 20:67–70, 1985.

[12] Sylvan Pinsky. Absorbing covers and intransitive non-interference. In *Proceedings of the Symposium on Security and Privacy*, pages 102–113, IEEE Computer Society, Oakland, CA, May 1995.

[13] *DO-178B: Software Considerations in Airborne Systems and Equipment Certification*. Requirements and Technical Concepts for Aviation, Washington, DC, December 1992. This document is known as EUROCAE ED-12B in Europe.

[14] A. W. Roscoe and M. H. Goldsmith. What is intransitive noninterference? In *12th Computer Security Foundations Workshop*, pages 228–238, IEEE Computer Society, Mordano, Italy, June 1999.

[15] John Rushby. The design and verification of secure systems. In *Eighth ACM Symposium on Operating*

System Principles, pages 12–21, Asilomar, CA, December 1981. (ACM *Operating Systems Review*, Vol. 15, No. 5).

[16] John Rushby. Kernels for safety? In T. Anderson, editor, *Safe and Secure Computing Systems*, chapter 13, pages 210–220. Blackwell Scientific Publications, 1989. (Proceedings of a Symposium held in Glasgow, October 1986).

[17] John Rushby. Noninterference, transitivity, and channel-control security policies. Technical Report SRI-CSL-92-2, Computer Science Laboratory, SRI International, Menlo Park, CA, December 1992.

[18] John Rushby. Partitioning for safety and security: Requirements, mechanisms, and assurance. NASA Contractor Report CR-1999-209347, NASA Langley Research Center, June 1999. Available at `http://techreports.larc.nasa.gov/ltrs/PDF/1999/cr/NASA-99-cr209347.pdf`; also to be issued by the FAA.

[19] Fred Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, February 2000.

[20] T. Taylor. Comparison paper between the Bell and La Padula model and the SRI model. In *Proceedings of the Symposium on Security and Privacy*, pages 195–202, IEEE Computer Society, Oakland, CA, April 1984.