A Comment on the "Basic Security Theorem" of Bell and LaPadula*

John McLean

Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, D.C. 20375

Many claim that the security model developed by Bell and LaPadula and used as a basis for numerous prototype military computer systems is superior to others partly because its authors prove a "Basic Security Theorem" that applies to it. This paper shows that the theorem does not support such claims since it can be proven for security models that are obviously not secure. Further, the theorem provides little help to those who design and implement secure systems.

1. Introduction

The security model developed by Bell and LaPadula [1] has been widely used as a basis for designing systems with specified security properties [2]. It has been argued that one reason developers should have confidence in the security provided by systems based on this model is a theorem, called the "Basic Security Theorem" (BST) [1, p. 20], proven about a formalization of the model by its authors [1,p.90, corollary A1]. Several authors have proven similarly named theorems about related security models [3,4,5]. This note reviews the Bell-LaPadula model briefly and shows that the BST can be proven for systems that directly contradict the notion of security embodied in the Bell-LaPadula model. We conclude that the value of the BST is much overrated since there is a great deal more to security than it captures. Further, what is captured by the BST is so trivial that it is hard to imagine a realistic security model for which it doesn't hold.

2. Bell-LaPadula Model

The Bell-LaPadula model is based on a state machine in which subjects apply operations (rules) that may require access to objects. The state of the system includes a set of triples that define the current access mode each subject has to each object in the system. Permissible access is determined partly by a security level (classification or clearance) associated with each object and subject. These security levels are partially ordered. Each subject also has a current security level that is bounded above by its clearance. There is also an access matrix that further constrains the access mode an arbitrary subject is allowed to have to an arbitrary object.

The following formal description of the Bell-LaPadula model corresponds to the original notation [1] as closely as possible, but nonessential details are omitted. Consider the sets S, O, and A whose elements are known as *subjects*, *objects*, and *access modes*, respectively. Intuitively, S consists of all system users and programs; O consists of all system files; and A is $\{$*read*, *execute*, *write*, *append*$\}$, the set of all modes in which an element of S can have access to an element of O. Bell and LaPadula define a *system state* v as an element of $V=(B\times M\times F\times H)$, where

- B — is the set of current accesses, a subset of $S\times O\times A$ that gives the access modes each subject currently has to each object,
- M — is the access permission matrix, where $M_{ij}\subseteq A$ is the set of access modes subject i may have to object j,
- F — consists of the three functions f_s, which gives the security level (clearance) associated with each subject, f_o, which gives the security level (classification) associated with each object, and f_c, which gives the current security level for each subject, and
- H — defines the current object hierarchy and is of no concern here.

*From *Information Processing Letters 20* (1985), pp. 67-70.

The set of requests (*e.g.*, to acquire or rescind access to objects) is denoted by R, and the set of decisions (*e.g.*, *yes*, *no*, *error*) is denoted by D. Finally, $W \subset R \times D \times V \times V$ represents the actions of the system: a request r yields a decision d and moves the system from state v to its successor.

Let T be the set of positive integers, and X, Y, and Z the set of functions from T to R, D, and V, respectively. The Bell-LaPadula model defines a *system* $\Sigma(R,D,W,z_0)$ to be a subset of $X \times Y \times Z$ such that $(x,y,z) \in \Sigma(R,D,W,z_0)$ if and only if $(x_t, y_t, z_t, z_{t-1}) \in W$ for each $t \in T$, where z_0 is the initial state of the system. Each triple $(x,y,z) \in \Sigma(R,D,W,z_0)$ is called an *appearance* of the system, and each quadruple (x_t, y_t, z_t, z_{t-1}) is called an *action* of the system.

The concept of a *secure state* is defined by three properties: the *simple security (ss) property*, the **-property*, and the *discretionary security (ds) property*. A state satisfies the ss-property if, for each element of B that has an access mode of *read* or *write*, the clearance of the subject dominates (in the partial order) the classification of the object. A triple (s,o,x) satisfies the simple security condition relative to f (SSC rel f) if x is *execute* or *append*, or if x is *read* or *write* and $f_s(s)$ dominates $f_o(o)$.

A state satisfies the *-property if, for each (s,o,x) in B, the current security level of s is equal to the classification of o if the access mode is *write*, dominates the classification of o if the access mode is *read*, and is dominated by the classification of o if the access mode is *append*. A state is said to satisfy the *-property relative to S', where $S' \subset S$, if this condition holds for all triples of B in which $s \in S'$. Subjects not in S' (and therefore not bound by the *-property relative to S') are called *trusted subjects*.

A state satisfies the ds-property if, for each member of B, the specified access mode is included in the access matrix entry for the corresponding subject-object pair. A state is *secure* if and only if it satisfies the ss-property, *-property relative to S' and the ds-property.

In addition to restricting subjects from having direct access to information for which they are not cleared, this concept of security is intended to prevent the unauthorized flow of information from a higher security level to a lower one. The *-property relative to S' specifically prevents nontrusted subjects from simultaneously having *read* access to information at one level and *write* access to information at a lower level.

Bell and LaPadula introduce analogous constraints on a system. A system appearance $(x,y,z) \in \Sigma(R,D,W,z_0)$ satisfies the ss-property if each state in the sequence $<z_0, z_1, \cdots>$ satisfies it.[1] A system satisfies the ss-property if each of its appearances does. Analogous definitions introduce the notions of a system satisfying the *- and ds-properties and the concept of a *secure system*. Theorems A1, A2, and A3 [see below], for the ss-, *-, and ds-properties respectively, show that a system $\Sigma(R,D,W,s_0)$ satisfies the property in question for any initial state that satisfies the property if and only if W (1) adds no new elements to B that would violate the property and (2) deletes any elements that, following the state change, would violate that property. The BST is presented without proof as a corollary of theorems A1, A2, and A3:

Basic Security Theorem: A system $\Sigma(R,D,W,z_0)$ is secure if and only if z_0 is a secure state and W satisfies the conditions of theorems A1, A2, and A3 for each action.[2]

3. Basic Security Theorem for an Alternative Security Model

Suppose that a different set of properties were chosen to define the concept of a secure state. If the BST is indeed a basis for having confidence that the Bell-LaPadula model captures the desired notion of security, then it should not be possible to prove a comparable theorem for a security model that has a substantially different definition for "secure state", and it certainly should not be possible to prove the theorem for a security model that is obviously not secure. The example below shows that this is not only possible but simple.

Define the *†–property* to hold for a state if, for each triple $(s,o,write)$ in B, the current security level of s dominates the classification of o. This is in essence the reverse of the *-property of Bell-LaPadula

---

1. In [1] an appearance satisfies the ss-property if each state in $<z_1, z_2, \cdots>$ satisfies the property; no restriction is placed on z_0. Nevertheless, the intent is clear since without this restriction, the BST as stated in [1] is false. See n. 2 below.
2. As noted in n. 1 above, this theorem as presented in [1] is actually false since it is possible for a system to be secure even though its initial state is not secure.

and allows subjects to transfer information from higher security levels to lower security levels. Hence, the model is not secure since it allows secret information to be copied into unclassified files, *e. g.*, the *Washington Post.* Define a secure state to be one that satisfies the ss-property, †-property and the ds-property. Following Bell and LaPadula, a "Basic Security Theorem" will be proven as a corollary from three other theorems.

Theorem A1: $\Sigma(R,D,W,z_0)$ satisfies the ss-property for any initial state z_0 that satisfies the ss-property iff W satisfies the following conditions for each action $(R_i,D_i,(b^*,M^*,f^*,H^*),(b,M,f,H))$:

(i) each $(s,o,x) \in b^* \sim b$ satisfies SSC rel f^*;

(ii) if $(s,o,x) \in b$ does not satisfy SSC rel f^*, then $(s,o,x) \notin b^*$.

Proof: Given in [1].

Theorem A2†: $\Sigma(R,D,W,z_0)$ satisfies the †-property relative to S', a subset of S, for any initial state z_0 that satisfies the †-property relative to S' iff W satisfies the following conditions for each action $(R_i,D_i,(b^*,M^*,f^*,H^*),(b,M,f,H))$:

(i) for each $s \in S'$, any $(s,o,x) \in b^* \sim b$ satisfies the †-property with respect to f^*;

(ii) for each $s \in S'$, if $(s,o,x) \in b$ does not satisfy the †-property with respect to f^*, then $(s,o,x) \notin b^*$.

[N.B.: This is the †-property analogue of a simplified statement of the original theorem A2.]

Proof:

(←)

Proof by strong induction: Assume that for all $i<n$ z_i satisfies the theorem and that z_n satisfies (i) and (ii). It follows that z_n satisfies the †-property as can be seen by the following argument:

> If $n=0$ then z_n satisfies the †-property by hypothesis. If $n>0$, then z_{n-1}. satisfies the †-property by hypothesis. The only way z_n could fail to satisfy the property is if a new write access has been granted that violates the property with respect to f^* or if an old write access is kept that violates the property relative to f^*. But the former possibility is ruled out by (i) and the latter by (ii).

(→)

Proof by contradiction: Assume that some state $\hat{z}$ satisfies the †-property but not (i). Then there is an $s \in S'$ such that (s,o,x) is in $b^* \sim b$ (and hence b^*), but fails to satisfy the †-property, yielding a contradiction. Similarly, if $z$satisfies the †-property but fails to satisfy (ii), then there is an $s \in S'$ such that (s,o,x) is in b, fails to satisfy the †-property, and is in b^* as well, also yielding a contradiction.

Theorem A3: $\Sigma(R,D,W,z_0)$ satisfies the ds-property iff the initial state z_0 satisfies the ds-property and W satisfies the following condition for each action $(R_i,D_i,(b^*,M^*,f^*,H^*),(b,M,f,H))$:

(i) if $(s_k,o_l,x) \in b^* \sim b$, then $x \in M^*_{k,l}$;

(ii) if $(s_k,o_l,x) \in b$ and $x \notin M^*_{k,l}$, then $(s_k,o_l,x) \notin b^*$.

Proof: Provided in [1].

Basic Security Theorem: $\Sigma(R,D,W,z_0)$ is a *secure system* (i. e. satisfies the ss-, †-, and ds-properties) iff z_0 is a secure state and W satisfies the conditions of theorems A1, A2†, and A3 for each action.

Clearly this exercise could be repeated, substituting alternative versions of either of the other security properties (e.g., only allowing users to read information classified *above* their clearances) as well.

4. Discussion

We have shown that the Basic Security Theorem does nothing to establish that a system is really secure. An analogue holds for any definition of "secure state" in a system whose states can be indexed to support induction. As such, it is a property of state indexing, not of security.

The real problems to be dealt with in considering a security model are (1) is the definition of "security" offered in the model a good one, i. e., does it capture what we really mean by "security", and (2) can we prove that a real system meets the definition. The first problem is not addressed by the BST since it is hard to imagine an explication of security for which there is not an analogous theorem, and the second problem is not made any simpler by the BST owing to the uninformativeness of the theorem's hypothesis.

This latter point deserves emphasis. It may seem as though the Basic Security Theorem is a significant tool in that it provides a means for proving security of every reachable system state by only considering the initial state and the rules that transform a system from one state to the next. However, the triviality of the tool renders it all but useless. Stripped of all formalism, the theorem states that if a system starts in a secure state and if all its transitions are such that at each state any old access that violates security under the new state's clearance functions is withdrawn and no new access is introduced that violates security, then the system will remain secure. But this is so obvious that it is of virtually no help.

In short, the theorem does not address the real problems. Nevertheless, the theorem has been advanced [1,3,6] as a substantial argument in favor of adopting the Bell-LaPadula model as a basis for developing secure systems, probably because people have confused the theorem with the nontrivial task of proving that an implementation meets the conditions of a given security definition. What is perhaps more damaging is that every new explication of "security" is expected to be accompanied by an analogue of the BST even though the time spent proving the theorem is, as should be clear by now, wasted. By focusing on the theorem, the security community has lost track of what needs to be done.

In fairness to Bell and La Padula, they do not seem to have suggested that the theorem addresses the above problems. They merely wanted to show that security is an "inductive" property, unlike (according to them) deadlock. Given the previous discussion, it should be obvious that an analogue of the theorem holds for deadlock as well. In any event, once security has been shown to be inductive, why insist on proving it over and over again?

Acknowledgments

The exposition of the Bell-LaPadula model owes much to Carl Landwehr. Marv Schaefer, when confronted with my doubts about the BST, first formulated the †-property and challenged me to prove the BST for the resulting security model.

References

[1] D. E. Bell and L. J. LaPadula, Secure computer system: unified exposition and Multics interpretation, MITRE MTR-2997, March 1976. Available as NTIS AD-A023 588.

[2] C. E. Landwehr, Best available technologies for computer security, IEEE Computer, July, 1983.

[3] K. G. Walter, W. F. Ogden, W. C. Rounds, F. T. Bradshaw, S. R. Ames, and D. G. Shumway, Primitive models for computer security, ESD-TR-74-117, January, 1974. Available as NTIS AD-778 467.

[4] J. McLean, C. E. Landwehr, and C. L. Heitmeyer, A formal statement of the MMS security model, Proc. 1984 Symposium on Security and Privacy, (IEEE Computer Society Press, 1984).

[5] C. E. Landwehr, C. L. Heitmeyer, and J. McLean, A security model for military message systems, Transactions on Computer Systems, to appear August, 1984.

[6] Panel session on Bell-LaPadula and alternative models of security, S. B. Lipner moderator, IEEE Symposium on Security and Privacy, April 1983.