Global Information Assurance Certification Paper

The Role of Mathematical Logic in Designing Secure Systems

By Kent Kavanagh

Mathematics truly is the "Universal Language". It has fascinated and frustrated men and women for thousands of years and continues to do so to this day. Unfortunately, many people's recollections of, and experiences with, mathematics are limited to the mundane and tedious rote learning that is popular in early education. As a result, many of these same people grow to despise maths as they know it, and are unable and/or unwilling to see the profound influence it has on our everyday life. It is fair to say that maths has made a fundamental contribution to the great majority of technological advances that have been made throughout the centuries, and it will continue to do so far into the future. As the "Mother of all Disciplines" scientists and technologists from all walks of life utilise the axioms and principles that maths has introduced over many long years of trial and error, theorising and proof.

The great beauty of mathematics is that its fundamental elements can be shown or proven – to use the mathematical lingo – to hold true in all given circumstances and, as a result, these fundamentals are relied on as a basis on which to build further ideas which can hence become fundamental also. Perhaps the greatest of these mathematical fundamentals is logic. The "*Star Trek*" television series is the best marketing engine mathematical logic has ever enjoyed, with the Vulcan, Dr Spok gleefully touting his superior intelligence through nothing more than the application of logical principles. While this may seem far-fetched, there is perhaps more truth to this scenario than anyone was aware and logic is as applicable to complex problems and systems as it ever was.

Computer systems have no intelligence to speak of and are reliant on the programs and people that control them to operate effectively and in a controlled manner. Computers accomplish this by continually making series of decisions based on the two conditions they are aware of – true or false. The bits and bytes within a computer can represent these conditions and the 1's and 0's that electronic pulses produce within a computer are the perfect medium for conveying one of these conditions. But what has this got to do with the design of secure systems? It is the aim of this paper to detail how mathematical logic can be utilised, both directly and indirectly, as part of a larger process, to assist with and enhance the design of secure systems.

Mathematical Modelling

Given the complexity of modern computer systems, it is no wonder that models of some form or another are used in an effort to gain an intimate understanding of their inner workings. More often than not, these models have their basis in mathematics. When such systems are concerned with "mission critical" or "trusted" tasks such as security – or safety and logistics as two more examples - mathematical logical is often used as a foundation for these models. It is the nature of analysis that discrete systems are far simpler to comprehend and assess, and the ability of logic to separate and clarify important components of these systems makes it an ideal choice in this area. Traditionally, logic, and more generally, mathematics is used for two purposes when modelling complex systems – specification and verification.

Specification

"Formal specification is the use of notations derived from formal logic to describe assumptions about the world in which a system will operate, requirements that the system is to achieve and to assist in a design to meet those requirements". **[7]** Perhaps a less formal definition is that specification allows the 'natural language' requirements of a system to be translated into a language that is more readily analysed through a formal process. The language most often used is mathematics and once translated, the requirements, as they are defined, can be subjected to the same analysis as other mathematical objects.

Verification

"The IEEE definition of verification is, confirmation by examination and provisions of objective evidence that specified requirements have been fulfilled". **[6]** In real terms what this means, is that verification is the application of known mathematical "truths" to the given specification of a system to determine if the system will behave in the desired manner. If it is not verified to behave as expected, specifications need to be changed so that when the mathematical techniques are applied to them, a verifiable system can be achieved. Simply put, verification is proving of specifications.

Bell-LaPadula Model

The best known and most widely used of all logical mathematical models when concerned with secure systems is the Bell-LaPadula Model **[2]**. This model is a formal state transition model of computer security policy that describes a set of access control rules. In this formal model, the entities in a computer system are represented as sets of subjects (S) and objects (O). The notion of a secure state is defined and it is proven that security, as defined, is maintained by moving from one secure state to another. As a result, the Principle of Mathematical Induction may be applied to prove that a system is secure if it conforms to particular rules. A system state is defined to be "secure" if the only permitted access (A) modes of subjects to objects are in accordance with a specific security policy. In order to determine whether or not a specific access mode is allowed, the clearance of a subject is compared to the classification of the object and a determination is made as to whether the subject is authorized for the specific access mode.
Bell and LaPadula define a system state v as an element of the (mathematical construct) set V, where the set V = (BxMxFxH) where:

B ➔ the set of current accesses, a sub-set SxOxA that defines the access each subject has to each object;
M ➔ is the access permission matrix, where M (in relation to a and b) exists as a sub-set of the accesses that the subject a may have to b;
F ➔ consists of three functions. F1 indicates each subjects "security clearance", F2

indicates the subject's current clearance and F3 indicates each objects classification; H ➔ defines the object hierarchy.
The set of requests that will grant or revoke given access of subjects to objects is R, and the set of responses to these requests is D. According to the model, the state of the system at any time can be considered as the sum of all the sets R,D,W and its initial state I. There are also a number of assumptions which help define the notion of a secure system, but, put as simply as possible, Bell and LaPadula developed a theorem which states that as long as the assumptions are satisfied and the system values R,D,W are matched with a secure state of I, a system can be proven to be secure. This theorem is known as The Basic Security Theorem (BST) and has proved remarkably effective as the basis for the specification and verification of many "trusted" systems.

Applying the BST to Information Systems

Great, I hear you say, but what has all this to do with information systems? On the surface, this seems as applicable to secure systems as adding x and y was to "real life" when we were all first taught basic algebra in primary school. However, both directly and indirectly, the lessons and thought processes born out of the mathematical logic which the BST has as it basis, can be applied to the security of information systems. To begin with, many Government and Military agencies wish to have assurances that the systems, which they are using, have been tested and verified to behave as they are supposed to. Three prominent incarnations of mathematical logic models used in the specification and verification of information systems are the Trusted Computer Security Evaluation Criteria (TCSEC – The Orange Book), Common Criteria and Information Technology Security Evaluation Criteria (ITSEC). All of these criteria have different levels of assurance that can be met and certified, depending on the specifications of a given system and the rigour applied in verifying these specifications. The reality is that these types of evaluations mean little to organizations not willing to spend the time and money to obtain them, and who are more interested a business-oriented approach to security. However, it is possible that the same principals applied in specifying and verifying systems may be applied, indirectly, to enhance the security of information systems designs. To illustrate this point let us consider **[4]**.
In this paper, the authors describe "a methodology for enumerating the vulnerabilities of a system, and determining what countermeasures can best close those vulnerabilities"**[4]**. The first point to note here is that it is impossible for mathematical logic to directly assist in enumerating such vulnerabilities and countermeasures. The types of vulnerabilities and countermeasures referred to require a good deal of experience and technical knowledge to correctly and effectively document and the generality of the logical process provided by mathematical theory provides no assistance in these technically specific areas. However, the same logical process may be employed to provide a framework through which an ordered assessment of a system may be undertaken. The paper is divided into three parts: an adversary model; a vulnerability model, and lastly; a proposed methodology. It is interesting to note that each of these parts of the paper has similarities to a mathematically logical process. To begin with an adversary is characterised in terms of their resources, access, risk tolerance and objectives. Each of these factors can be considered to be a logical state,

and the profile of an adversary will change depending on the state of each of these characteristics. Accordingly, the entire model will be affected, all the way through to which countermeasures could be applied, depending on which logical state an adversary assumes. Secondly, the "additive" nature of the adversary profile also has similarities to rigorous mathematical modelling, in that at any one time, an adversary profile is considered to be a combination of a number of these different states eg a "hacker" is profiled as an adversary of low resources, low access, moderate risk tolerance, while an "info warrior" is profiled as possessing significant resources, high access and high risk tolerance.
Secondly, a vulnerability modelling process is employed. The objective of Salter et al paper is to determine a "vulnerability landscape" of a system, which can be analysed in order to develop effective countermeasures. Once again, this landscape can be considered a logical construct or a set, where the profile of the set is simply the sum of all its parts. Similar to the adversary model, the landscape can be changed by the altering of any one of its parts (states). It is interesting to note here that not only is this method used to defend a system, but attackers also employ the same techniques in order to launch more directed attacks. This process, whether it be for defenders or attackers is nothing more than the specification of a system. The specification considers the entire life cycle of the components that is, once again, analogous to considering the different logical states that each component may find itself in. o further illustrate this point, the concept of the "trust model" is highlighted. The trust model is nothing more than assigning the access that each object (in this case employees) has to other objects within the system.
The similarities between the "Secure System Engineering Methodology" described in **[4]**, and the logical models described previously should be clear by now. The shortfall in the mathematical modelling process is in identifying countermeasures. Such countermeasures are best determined through experience, technical expertise, and a consideration of the business objectives of the system. If a theoretical approach were taken to identifying countermeasures, unrealistic and unworkable systems could be developed which is not the objective of applying security techniques to the design of information systems. However, the similarities between the formal mathematical models, as demonstrated by the BST, and more generic security models, as described in the paper by Salter et al, should now be clear.

Summary

The similarities between formal mathematical models, as demonstrated by the BST, and more generic security models, as described in the paper by Salter et al, should now be clear. It has been demonstrated how the logical methods and thought processes apparent in models such as the BST, can be applied in a more general sense to assist in modelling information systems, in order to assist in enhancing the design of security into these systems.
The specification process utilised in the BST can prove extremely valuable when attempting to map the objects of an information system and the vulnerabilities that are associated with these objects. The transformation of a natural language description of these objects into a more formal language allows for more effective analysis of the objects, and can assist in describing the interaction of the system's components. The analysis that is then applied to the specification need not be of a complicated

mathematical nature – indeed, to rely on this purely theoretical analysis can prove very dangerous as will be described below – but some analysis is a requirement none the less, and this analysis is greatly enhanced if a clear specification is provided. Such specification of a system is an essential requirement if any verification is to be attempted. This is because verification is nothing more than an analysis of a given system's specification to determine if the objects within a system behave as expected. An example of how such specification and verification can be applied to all components of an information system is given in **[5]**. This paper gives the example of specifying a system's security policy and how this specification can then be used to analyse policies and determine any areas that do not perform the duty that is required of them. It is simply another example of transforming natural language descriptions of a system into a more formal notation, which, in turn, lends itself to more effective analysis.
In its most general sense, the type of modelling described in this paper is nothing more than threat and risk analysis. In order to determine the risks to a system, each component of that system needs to be understood and the threats to that component, and hence the entire system, need to be considered. This type of threat and risk analysis is fundamental to the design of good security in information systems, and it has been demonstrated how the formal mathematical techniques of theoretical models can assist in this type of threat and risk analysis. This type of modelling is just as important to the security process as technical concerns and good security professionals should be aware of its usefulness as part of the overall process.

Conclusion

In concluding this paper, I would like to emphasise that modelling of information systems, and the formal mathematical notions of specification and verification that are a basis for these models, are only a part of the entire security process. Further, it needs to be noted that the formal foundations of these models are, in some ways, flawed and any honest assessment of their effectiveness needs to consider this fact.
To highlight this point, we will consider **[3]** "A comment on the BST". As stated when describing Bell and LaPadula's BST, a number of assumptions are made when defining the notion of a secure system. As is the case when proving many mathematical theorems, these assumptions need to be in effect for the proof to be correct and it is no different with the BST. What is interesting in the case of the BST is that if a different set of assumptions is used, the BST can still be proven to be mathematically correct. The startling point is that this is true, even if assumptions are made that obviously describe an insecure system!! The result of this is that the BST "… is so trivial that it is hard to imagine a realistic security model for which it doesn't hold" and, therefore is not the "silver bullet" that many would like it to be. This needs to be considered when assessing the claims made about systems that have been specified and verified utilising formal mathematical models (eg ITSEC, TCSEC…). Even though systems evaluated under these types of criteria can be proven mathematically to behave as expected (and therefore be trusted), the assumptions made in specifying the system to begin with need to be taken into account. An example of this is the Windows NT operating system, which was able be evaluated under ITSEC to E3. This may seem to indicate that NT could be considered a secure operating system, however the evaluation to this level was made without any

networking capability at all, meaning that the assurance level provided is of no use in any meaningful implementation of the operating system. This is the danger of formal methods for specifying and verifying information systems, and it needs to be remembered that the formal proving of security measures in these instances can only be performed against the given claims of security. As McClean points out, the real value provided by the BST is that security can be demonstrated to be an inductive property – meaning that a change in state of one object can affect the state of other objects and hence the entire system. It is this type of fundamental logical thinking that may be taken and applied to more general concerns of security and used to enhance the design of secure systems.

So remember, even though mathematics has for centuries been used to advance technology, even it cannot provide the security "silver bullet" that is hoped for. Indeed, logic dictates that a system can't be proved to be secure, but can only be proved to be insecure. However, these types of mathematical processes, when utilised as part of the overall security process, can assist in enhancing the design of secure systems.

References

[1] Dhillon, Gurpreet and Backhouse, James. Managing for secure organizations: a critique of information systems security research approaches. http://csrc.lse.ac.uk/People/BackhouseJ/isapproaches.htm

[2] LaPadula, Len. Secure Computer Systems: Mathematical Foundations. November 1996 – an electronic reconstruction of the original MITRE Technical Report 2547, Volume 1.

[3] McLean, John. A Comment on the "Basic Security Theorem" of Bell and LaPadula. Center for High Assurance Computer Systems, Naval Research Labaoratory. http://chacs.nrl.navy.mil/publications/CHACS/Before1990/1985mclean-ipl.pdf

[4] Salter, Chris et al. Toward a Secure System Engineering Methodology. www.counterpane.com/secure-methodology.html

[5] Sloman, M. et al. SecPol: Specification and Analysis of Security Policy for Distributed Systems. Imperial College, Department of Computing. http://www-dse.doc.ic.ac.uk/projects/secpol/SecPol-overview.html.

[6] www.asc-tech.com/vv.htm

[7] http://shemesh.larc.nasa.gov/fm/fm-what-def-fspec.html